Firepower Threat Defense(FTD)クラスタのトラブルシューティング

内容

はじめに

このドキュメントでは、Firepower Next-Generation Firewall(NGFW)でのクラスタセットアップのトラブルシューティングについて説明します。

前提条件

要件

次の項目に関する知識があることが推奨されます（リンクについては、「関連情報」セクションを参照）。

- Firepower プラットフォーム アーキテクチャ
- Firepowerクラスタの設定と動作
- FTDおよびFirepower eXtensible Operating System(FXOS)CLIに精通していること
- NGFW/データプレーンの口グ
- NGFW/データプレーンパケットトレーサ
- FXOS/データプレーンのキャプチャ

使用するコンポーネント

- HW: Firepower 4125
- SW:6.7.0（ビルド65）– データプレーン9.15(1)

このドキュメントの情報は、特定のラボ環境にあるデバイスに基づいて作成されました。このドキュメントで使用するすべてのデバイスは、クリアな（デフォルト）設定で作業を開始しています。本稼働中のネットワークでは、各コマンドによって起こる可能性がある影響を十分確認してください。

背景説明

このドキュメントで説明する項目のほとんどは、適応型セキュリティアプライアンス(ASA)クラスタのトラブルシューティングにも完全に適用できます。

設定

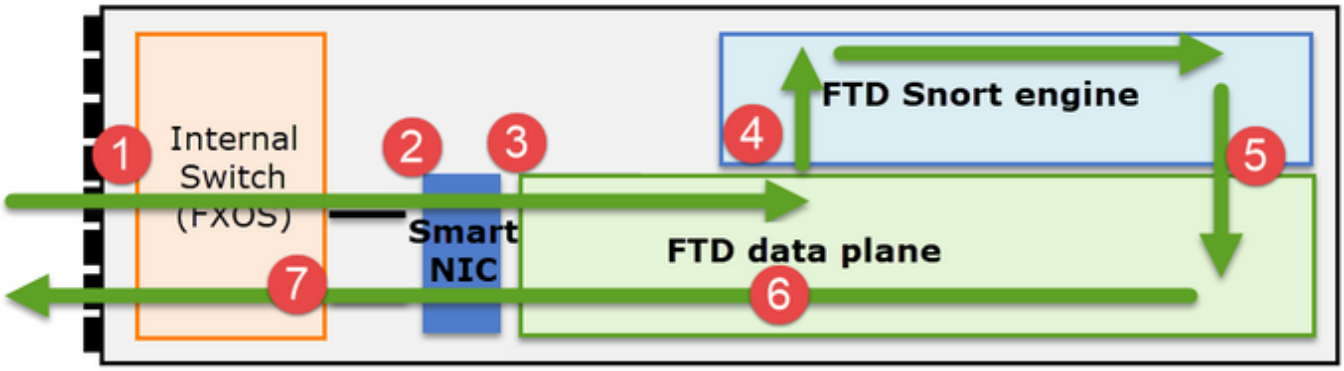
クラスタ導入の設定部分については、FMCおよびFXOSの設定ガイドで説明します。

- Firepower Threat Defenseのクラスタリング
- スケーラビリティとハイアベイラビリティを実現するFirepower Threat Defense用のクラスタの導入

クラスタの基本

NGFWアーキテクチャ

Firepower 41xxまたは93xxシリーズが中継パケットを処理する方法を理解することが重要です。



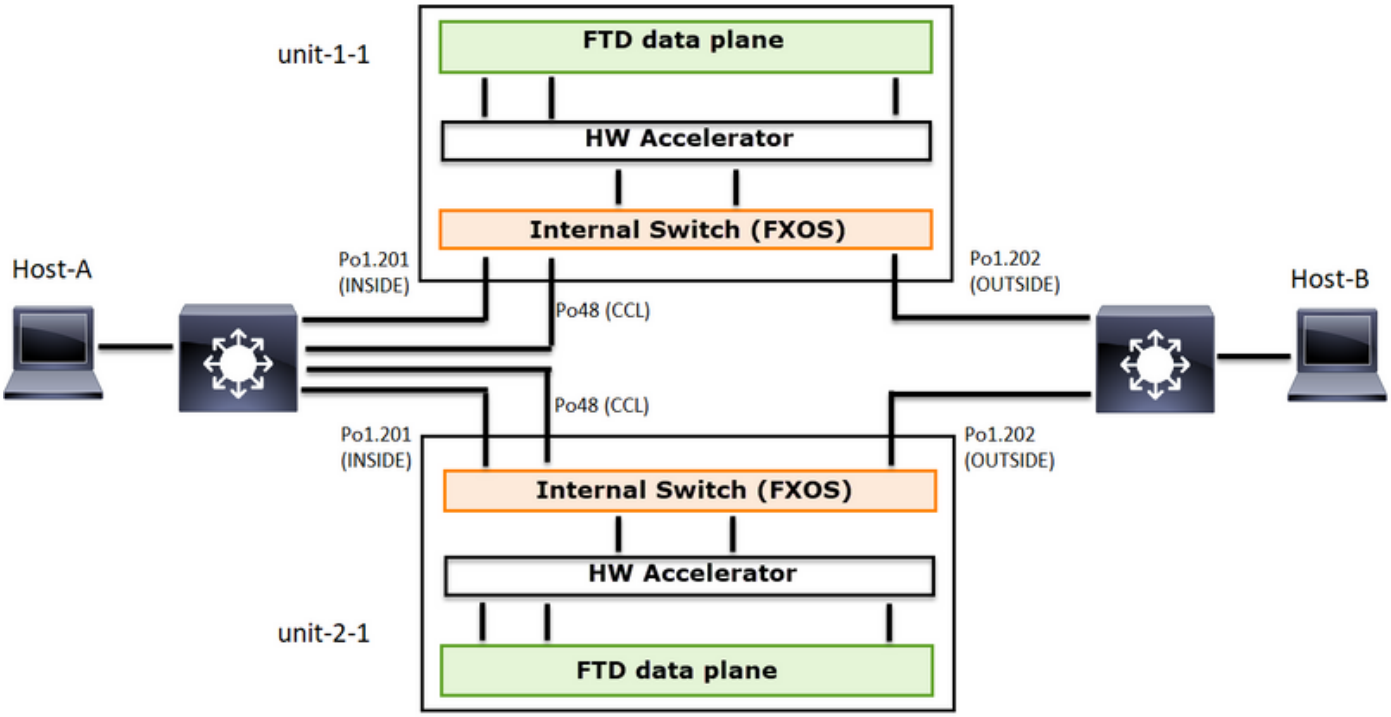
1. パケットが入力インターフェイスに入り、シャーシの内部スイッチによって処理されます。
2. パケットはSmart NICを通過します。フローがオフロード（HWアクセラレーション）されると、パケットはSmart NICによってのみ処理され、ネットワークに戻されます。
3. パケットがオフロードされない場合、パケットは主にL3/L4チェックを行うFTDデータプレーンに入ります。
4. ポリシーで必要とされる場合、パケットはSnortエンジンによって検査されます（主にL7検査）。
5. Snortエンジンは、パケットに対する判定（許可やブロックなど）を返します。
6. データプレーンは、Snortの判定に基づいてパケットをドロップまたは転送します。
7. パケットがシャーシの内部スイッチを通過してシャーシから出ます。

クラスタのキャプチャ

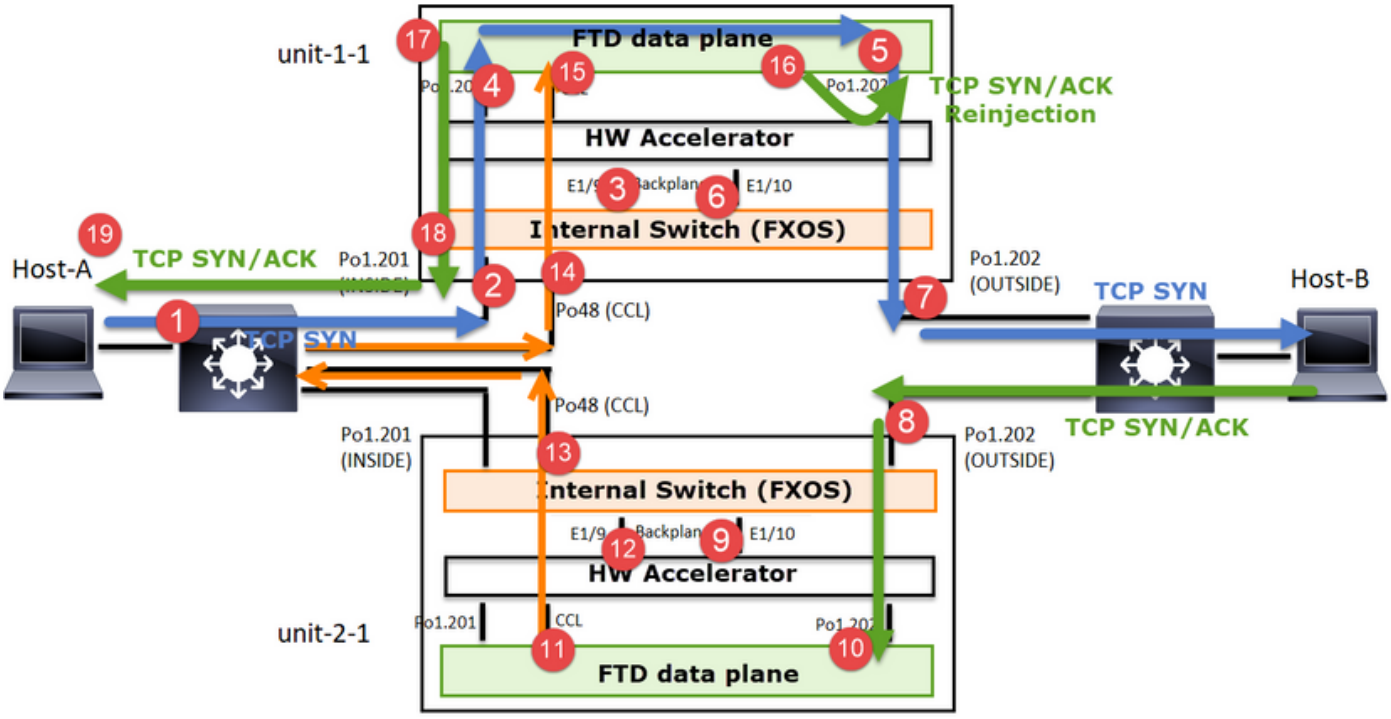
Firepowerアプライアンスは、中継フローを可視化する複数のキャプチャポイントを提供します。クラスタキャプチャのトラブルシューティングと有効化を行う際の主な課題は次のとおりです。

- クラスタ内のユニット数が増えるほど、キャプチャの数も増えます。
- クラスタを通過するパケットを追跡するには、クラスタが特定のフローをどのように処理するかを認識する必要があります。

次の図は、2ユニットのクラスタ（FP941xx/FP9300など）を示しています。



非対称TCP接続が確立される場合、TCP SYN、SYN/ACK交換は次のようになります。



転送トラフィック

1. TCP SYNがホストAからホストBに送信されます。
2. TCP SYNがシャーシ（Po1のメンバの1つ）に着信します。
3. TCP SYNは、シャーシバックプレーンインターフェイスの1つ（E1/9、E1/10など）を介してデータプレーンに送信されます。
4. TCP SYNがデータプレーン入力インターフェイス（Po1.201/内部）に着信します。 この例では、unit1-1はフローの所有権を取得し、初期シーケンス番号(ISN)のランダム化を行い、シーケンス番号の所有権(cookie)情報をエンコードします。

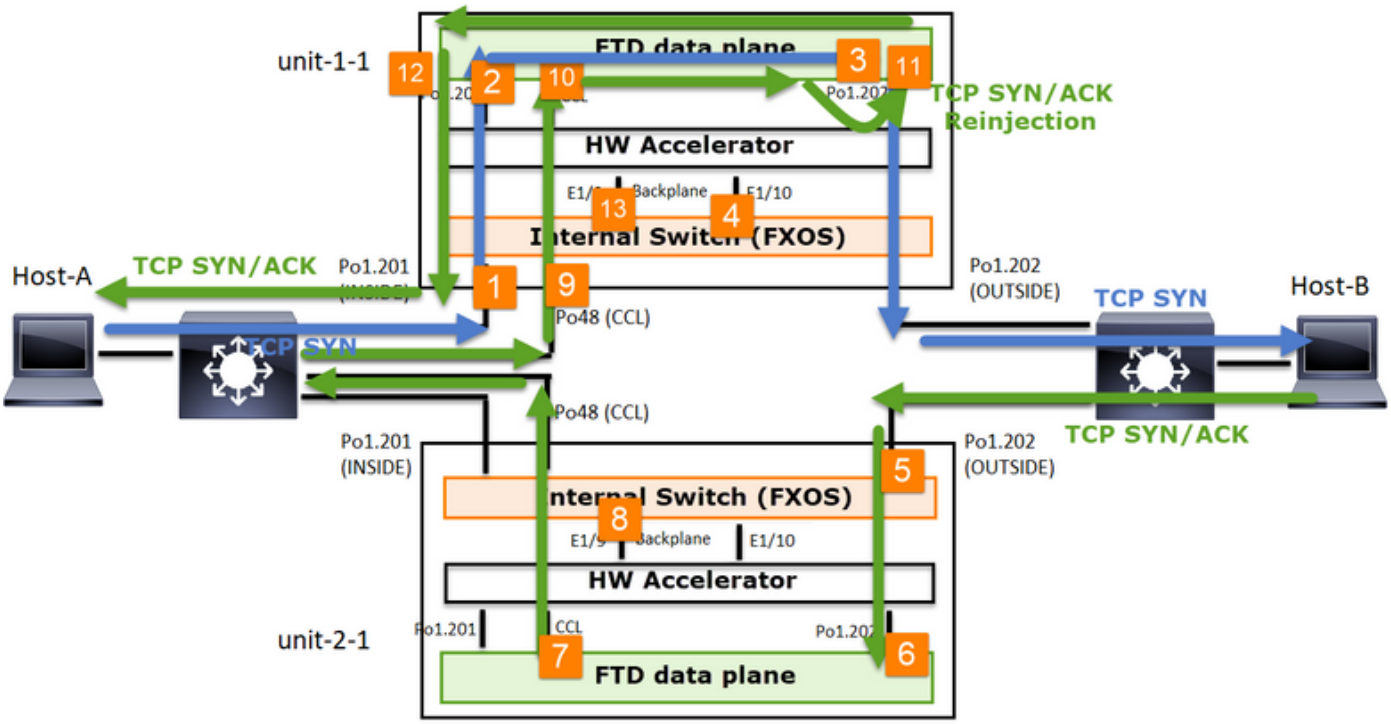
5. TCP SYNはPo1.202/OUTSIDE（データプレーン出力インターフェイス）から送信されます。
6. TCP SYNがシャーシバックプレーンインターフェイスの1つに到着します（E1/9、E1/10など）。
7. TCP SYNは、シャーシの物理インターフェイス（Po1のメンバの1つ）からHost-Bに向けて送信されます。

リターントラフィック

8. TCP SYN/ACKがホストBから送信され、ユニット2-1（Po1のメンバの1つ）に到達します。
9. TCP SYN/ACKは、シャーシバックプレーンインターフェイスの1つ（E1/9、E1/10など）を介してデータプレーンに送信されます。
10. TCP SYN/ACKがデータプレーン入力インターフェイス(Po1.202/OUTSIDE)に到達します。
11. TCP SYN/ACKは、Cluster Control Link(CCL)からユニット1-1に向けて送信されます。デフォルトでは、ISNは有効になっています。したがって、フォワーダは、ディレクタの関与なしにTCP SYN+ACKの所有者情報を検出します。その他のパケットの場合、またはISNが無効の場合は、ディレクタへの問い合わせが行われます。
12. TCP SYN/ACKがシャーシバックプレーンインターフェイスの1つに到着します（E1/9、E1/10など）。
13. TCP SYN/ACKは、シャーシの物理インターフェイス（Po48のメンバの1つ）からユニット1-1に向けて送信されます。
14. TCP SYN/ACKがユニット1-1（Po48のメンバの1つ）に到達します。
15. TCP SYN/ACKは、シャーシバックプレーンインターフェイスのいずれかを介して、データプレーンCCLポートチャネルインターフェイス（nameifクラスタ）に転送されます。
16. データプレーンは、TCP SYN/ACKパケットをデータプレーンインターフェイスPo1.202/OUTSIDEに再注入します。
17. TCP SYN/ACKは、Po1.201/INSIDE（データプレーン出力インターフェイス）からHOST-Aに向けて送信されます。
18. TCP SYN/ACKは、シャーシバックプレーンインターフェイスの1つ（E1/9、E1/10など）を通過し、Po1のいずれかのメンバから出力されます。
19. TCP SYN/ACKがホストAに到達します。

このシナリオの詳細については、「クラスタ接続の確立ケーススタディ」の関連セクションを参照してください。

このパケット交換に基づいて、考えられるすべてのクラスタキャプチャポイントは次のとおりです。



転送トラフィック（TCP SYNなど）のキャプチャの対象：

1. シャーシの物理インターフェイス（Po1メンバなど）。 このキャプチャは、Chassis Manager(CM)UIまたはCM CLIから設定します。
2. データプレーン入力インターフェイス（Po1.201 INSIDEなど）。
3. データプレーン出力インターフェイス（Po1.202 OUTSIDEなど）。
4. シャーシバックプレーンインターフェイスFP4100には2つのバックプレーンインターフェイスがあります。FP9300では、合計6個（モジュールあたり2個）あります。 パケットがどのインターフェイスに到達するかわからないため、すべてのインターフェイスでキャプチャを有効にする必要があります。

リターントラフィック（TCP SYN/ACKなど）のキャプチャは次のとおりです。

5. シャーシの物理インターフェイス（Po1メンバなど）。 このキャプチャは、Chassis Manager(CM)UIまたはCM CLIから設定します。
6. データプレーン入力インターフェイス（Po1.202 OUTSIDEなど）。
7. パケットはリダイレクトされるため、次のキャプチャポイントはデータプレーンCCLです。
8. シャーシバックプレーンインターフェイスここでも、両方のインターフェイスでキャプチャを有効にする必要があります。
9. ユニット1-1シャーシCCLメンバーインターフェイス
10. データプレーンCCLインターフェイス（nameifクラスタ）。
11. 入力インターフェイス（Po1.202外部）。 これは、CCLからデータプレーンに再注入されたパケットです。
12. データプレーン出力インターフェイス（Po1.201 INSIDEなど）。
13. シャーシバックプレーンインターフェイス

クラスタキャプチャを有効にする方法

FXOSキャプチャ

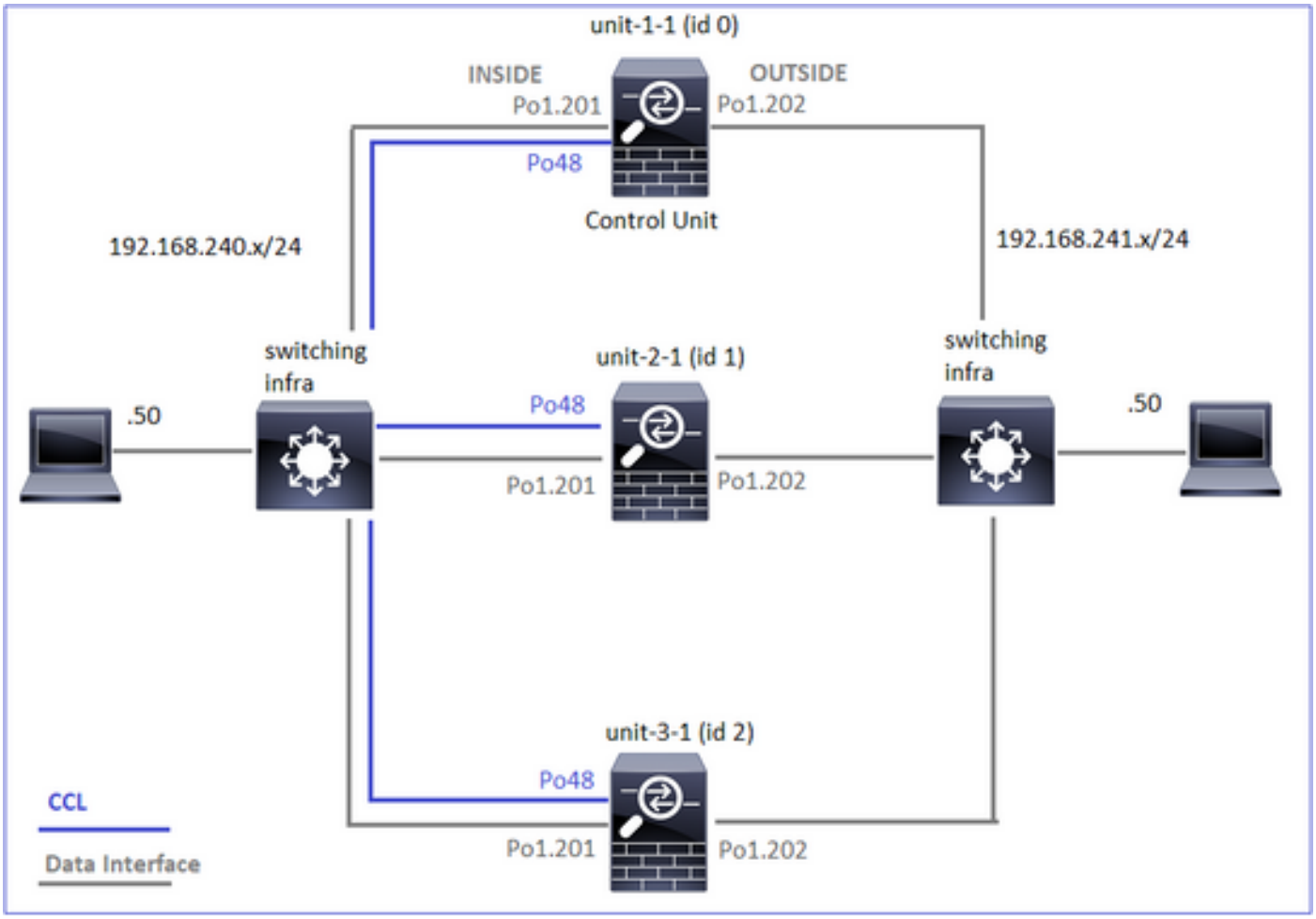
このプロセスは、FXOS構成ガイド：パケットキャプチャで説明されています。

注:FXOSキャプチャは、内部スイッチの観点からは入力方向でのみ取得できます。

データプレーンのキャプチャ

すべてのクラスタメンバーでキャプチャを有効にする推奨方法は、cluster exec コマンドを使用することです。

3ユニットのクラスタを考えてみます。



すべてのクラスタユニットにアクティブなキャプチャがあるかどうかを確認するには、次のコマンドを使用します。

```
<#root>

firepower#

cluster exec show capture


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

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

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

Po1.201（内部）のすべてのユニットでデータプレーンキャプチャを有効にするには、次の手順を実行します。

<#root>

```
firepower#
```

cluster exec capture CAPI interface INSIDE

キャプチャフィルタを指定し、大量のトラフィックが予想される場合は、キャプチャバッファを増やすことを強くお勧めします。

<#root>

```
firepower#
```

cluster exec capture CAPI buffer 33554432 interface INSIDE match tcp host 192.168.240.50 host 192.168.24

検証

<#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
```

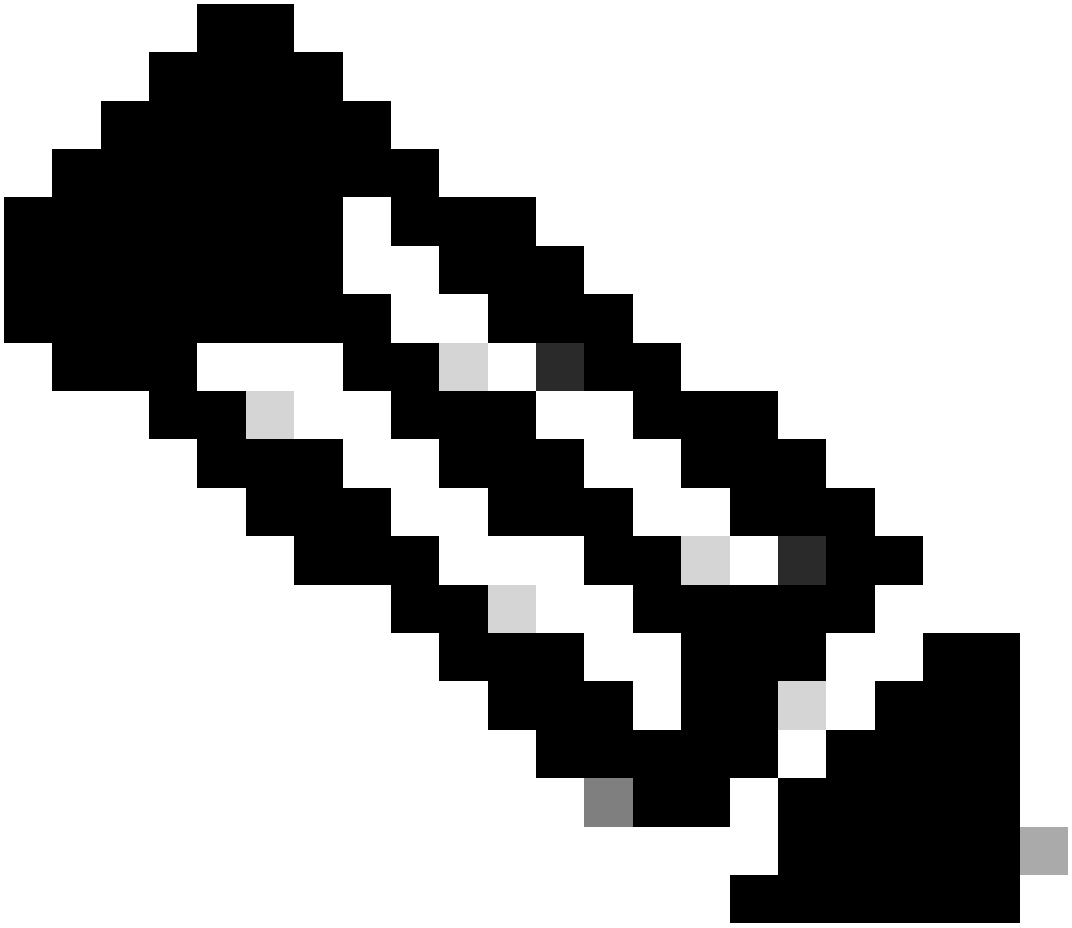
すべてのキャプチャの内容を表示するには、次のコマンドを実行します（この出力は非常に長くなることがあります）。

<#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 P0 192.168.240.50.45456 > 192.168.241.50.80: S 2225395909:2225395909
2: 11:33:09.880401 802.1Q vlan#201 P0 192.168.241.50.80 > 192.168.240.50.45456: S 719653963:719653963(0
3: 11:33:09.880691 802.1Q vlan#201 P0 192.168.240.50.45456 > 192.168.241.50.80: . ack 719653964 win 229
4: 11:33:09.880783 802.1Q vlan#201 P0 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
```

トレースのキャプチャ

各ユニットで入力パケットがデータプレーンでどのように処理されるかを調べるには、traceキーワードを使用します。これにより、最初の50個の入力パケットがトレースされます。最大1000の入力パケットをトレースできます。



注：インターフェイスに複数のキャプチャを適用する場合、1つのパケットをトレースできるのは1回だけです。

すべてのクラスタユニットのインターフェイスOUTSIDEで最初の1000個の入力パケットをトレースするには、次のコマンドを実行します。

<#root>

firepower#

cluster exec cap CAPO int OUTSIDE buff 33554432 trace trace-count 1000 match tcp host 192.168.240.50 hos

対象のフローをキャプチャしたら、各ユニットで対象のパケットをトレースしていることを確認する必要があります。特定のパケットをUnit-1-1で#1信し、別のユニットで#2信するなどの方法をとることが重要です。

この例では、SYN/ACKはユニット2-1ではパケット#2ですが、ユニット3-1ではパケット#1であることがわかります。

```
<#root>

firepower#

cluster exec show capture CAPO | include S.*ack


unit-1-1(LOCAL):*******************************************************
1: 12:58:31.117700 802.1Q vlan#202 P0 192.168.240.50.45468 > 192.168.241.50.80: S 441626016:441626016(0
2: 12:58:31.118341 802.1Q vlan#202 P0 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 P0 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>
```

ローカルユニットでパケット#2(SYN/ACK)をトレースするには、次のコマンドを実行します。

```
<#root>

firepower#

cluster exec show cap CAPO packet-number 2 trace


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

2: 12:58:31.118341 802.1Q vlan#202 P0 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
...
```

リモートユニットで同じパケット(SYN/ACK)をトレースするには、次のコマンドを実行します。

```
<#root>

firepower#

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



1: 12:58:31.111429 802.1Q vlan#202 P0 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キャプチャ

CCLリンク（すべてのユニット）でキャプチャを有効にするには、次の手順を実行します。

```
<#root>

firepower#

cluster exec capture CCL interface cluster


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

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

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

再インジェクト非表示

デフォルトでは、データプレーンデータインターフェイスで有効になっているキャプチャは、すべてのパケットを示します。

- 物理ネットワークから到達するもの
- CCLから再注入されるもの

再注入されたパケットを表示したくない場合は、reinject-hide オプションを使用します。これは、フローが非対称であるかどうかを確認する場合に役立ちます。

```
<#root>

firepower#

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

このキャプチャでは、ローカルユニットが特定のインターフェイスで物理ネットワークから直接受信した内容だけが示されており、他のクラスタユニットからは受信していません。

ASPドロップ

特定のフローのソフトウェアドロップを確認する場合は、asp-dropキャプチャを有効にできます。フォーカスするドロップの理由がわからない場合は、キーワードallを使用します。また、パケットペイロードを調べない場合は、headers-only キーワードを指定できます。これにより、20 ~ 30倍のパケットをキャプチャできます。

```
<#root>

firepower#

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


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

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

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

さらに、ASPキャプチャで対象のIPを指定できます。

```
<#root>

firepower#

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


match ip host 192.0.2.100 any
```

キャプチャのクリア

すべてのクラスタユニットで実行されているキャプチャのバッファをクリアします。これにより、キャプチャは停止されず、バッファがクリアされるだけです。

```
<#root>

firepower#

cluster exec clear capture /all


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

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

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

キャプチャの停止

すべてのクラスタユニットでアクティブなキャプチャを停止するには、2つの方法があります。後で再開できます。

方法1

```
<#root>

firepower#

cluster exec cap CAPI stop


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

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

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

再開するには

```
<#root>

firepower#

cluster exec no capture CAPI stop


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

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

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

方法2

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

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

再開するには

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

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

キャプチャの収集

キャプチャをエクスポートする方法は複数あります。

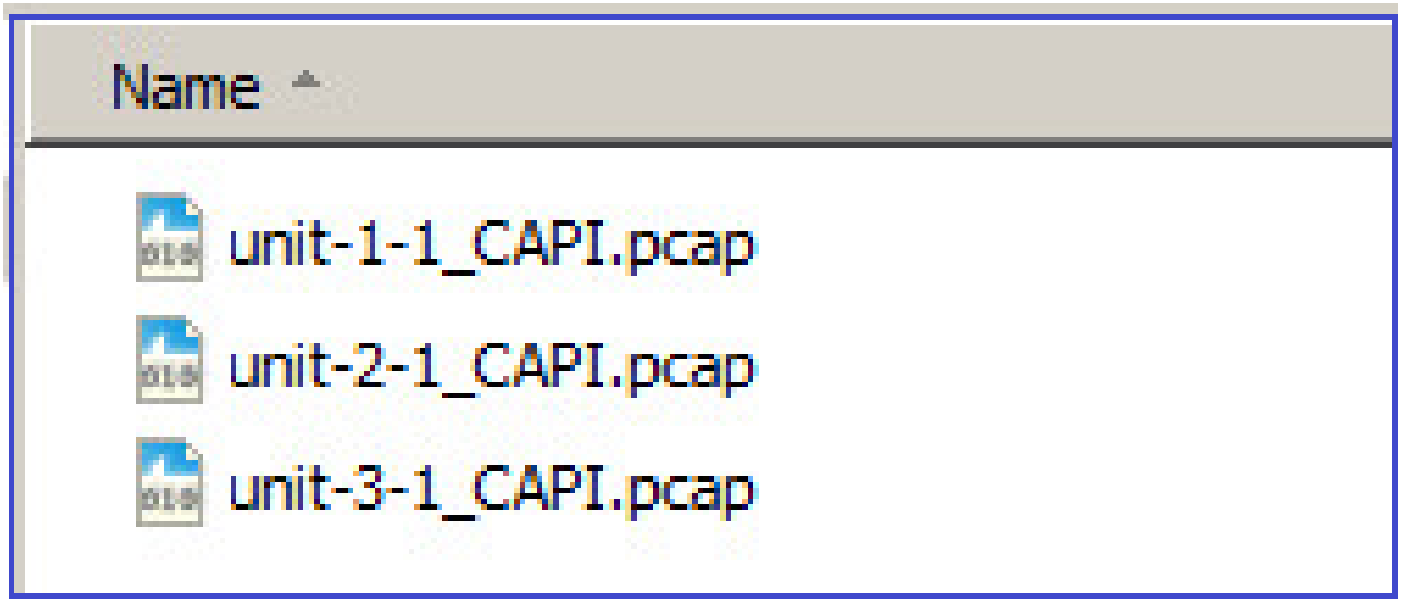
方法1：リモートサーバへ

これにより、データプレーンからリモートサーバ（TFTPなど）にキャプチャをアップロードできます。 キャプチャ名は、ソースユニットを反映して自動的に変更されます。

```
<#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 !
```

アップロードされたpcapファイル：



方法2:FMCからキャプチャを取得する

この方法は、FTDにのみ適用されます。最初に、キャプチャをFTDディスクにコピーします。

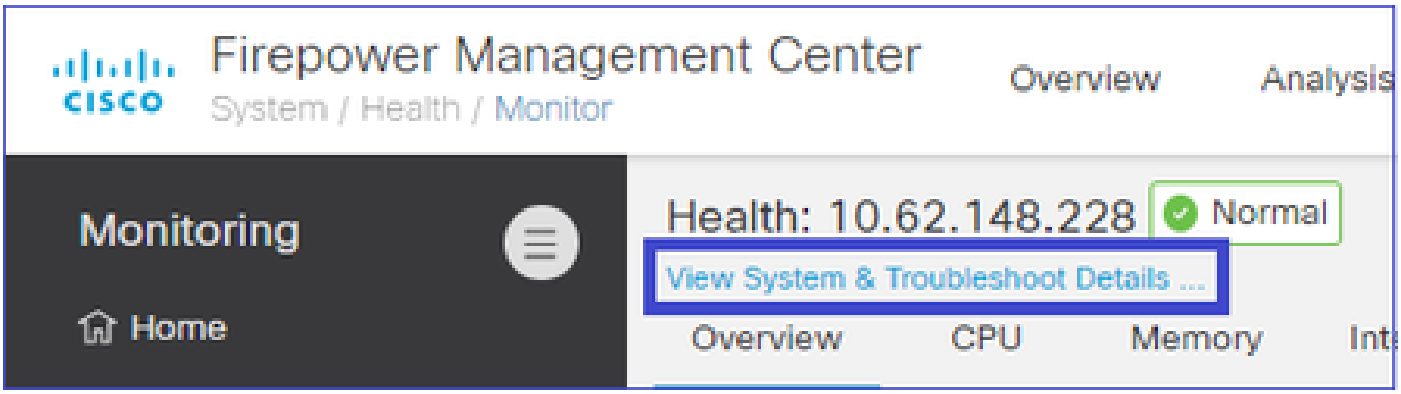
<#root>

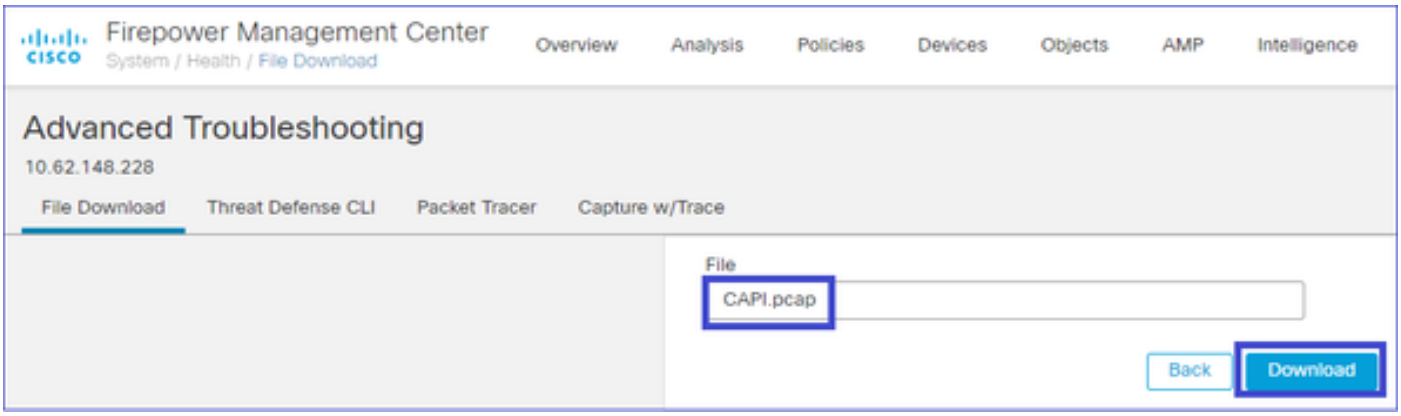
```
firepower#

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
```

expertモードから、/mnt/disk0/から/ngfw/var/common/ディレクトリにファイルをコピーします。

```
<#root>

>

expert

admin@firepower:~$

cd /mnt/disk0

admin@firepower:/mnt/disk0$

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

最後に、FMCでSystem > Health > Monitorセクションに移動します。View System & Troubleshoot Details > Advanced Troubleshooting の順に選択し、キャプチャファイルを取得します。





キャプチャの削除

すべてのクラスタユニットからキャプチャを削除するには、次のコマンドを使用します。

```
<#root>
```

```
firepower#
cluster exec no capture CAPI

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

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

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

オフロードされたフロー

FP41xx/FP9300では、フローをハードウェアアクセラレータに静的（Fastpathルールなど）または動的にオフロードできます。フローオフロードの詳細については、次のドキュメントを確認してください。

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

フローがオフロードされる場合、少数のパケットだけがFTDデータプレーンを通過します。それ以外は、ハードウェアアクセラレータ(Smart NIC)によって処理されます。

キャプチャの観点から見ると、これはFTDのデータプレーンレベルのキャプチャだけを有効にしても、デバイスを通過するすべてのパケットが表示されないことを意味します。この場合、FXOSシャーシレベルのキャプチャも有効にする必要があります。

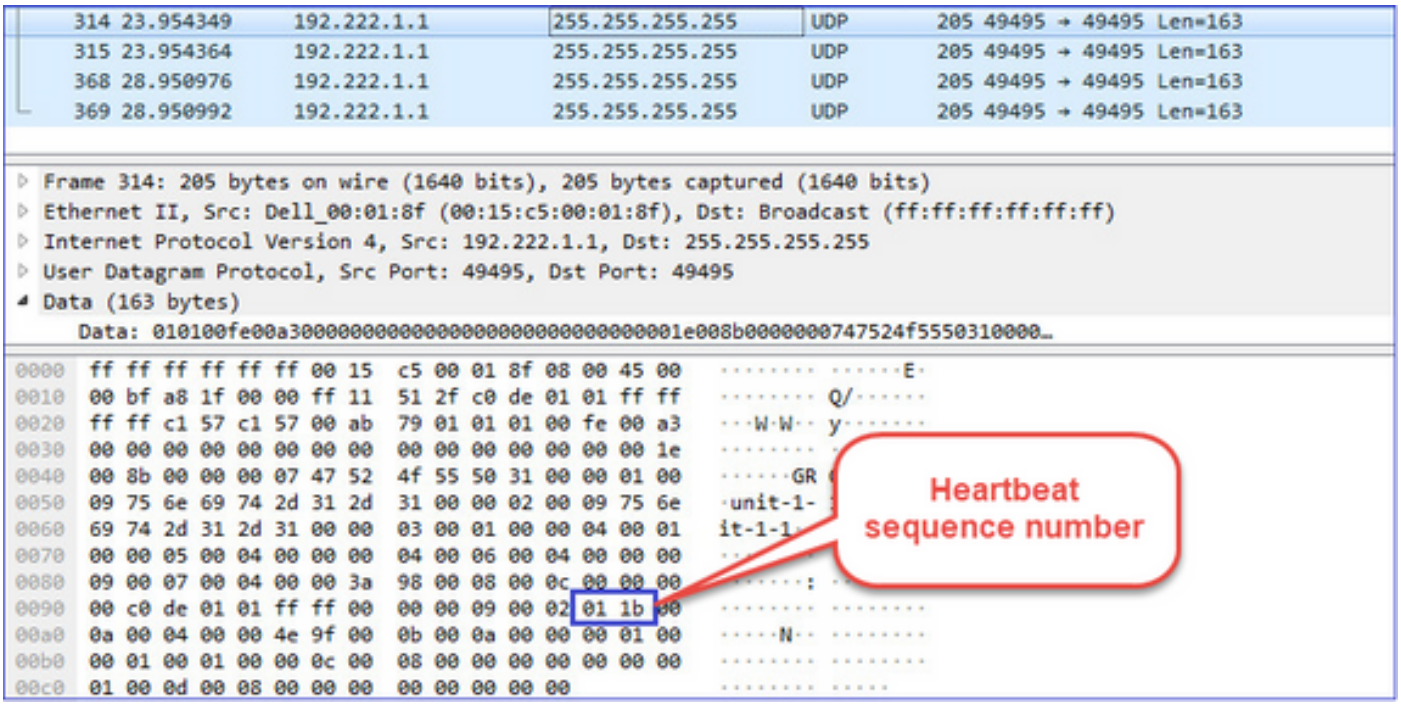
Cluster Control Link(CCL)メッセージ

CCLでキャプチャを実行すると、クラスタユニットが異なるタイプのメッセージを交換していることがわかります。対象となるのは次の項目です。

プロトコル	説明
UDP 49495	クラスタハートビート（キープアライブ） ・ L3ブロードキャスト(255.255.255.255) ・ これらのパケットは、ヘルスチェックの保留時間値の3分の1の時間に、すべてのクラスタユニットから送信されます。 ・ キャプチャに表示されるすべてのUDP 49495パケットがハートビートとは限らないことに注意してください ・ ハートビートにはシーケンス番号が含まれます。
UDP 4193	クラスタ制御プロトコルのデータパスメッセージ

	・ ユニキャスト ・ これらのパケットには、フロー所有者、ダイレクタ、バックアップ所有者などに関する情報（メタデータ）が含まれています。次に例を示します。 ・ 新しいフローが作成されると、所有者からダイレクタに「cluster add」メッセージが送信される ・ フローが終了すると、「cluster delete」メッセージが所有者からダイレクタに送信される
データパケット	クラスタを通過するさまざまなトラフィックフローに属するデータパケット

クラスタハートビート



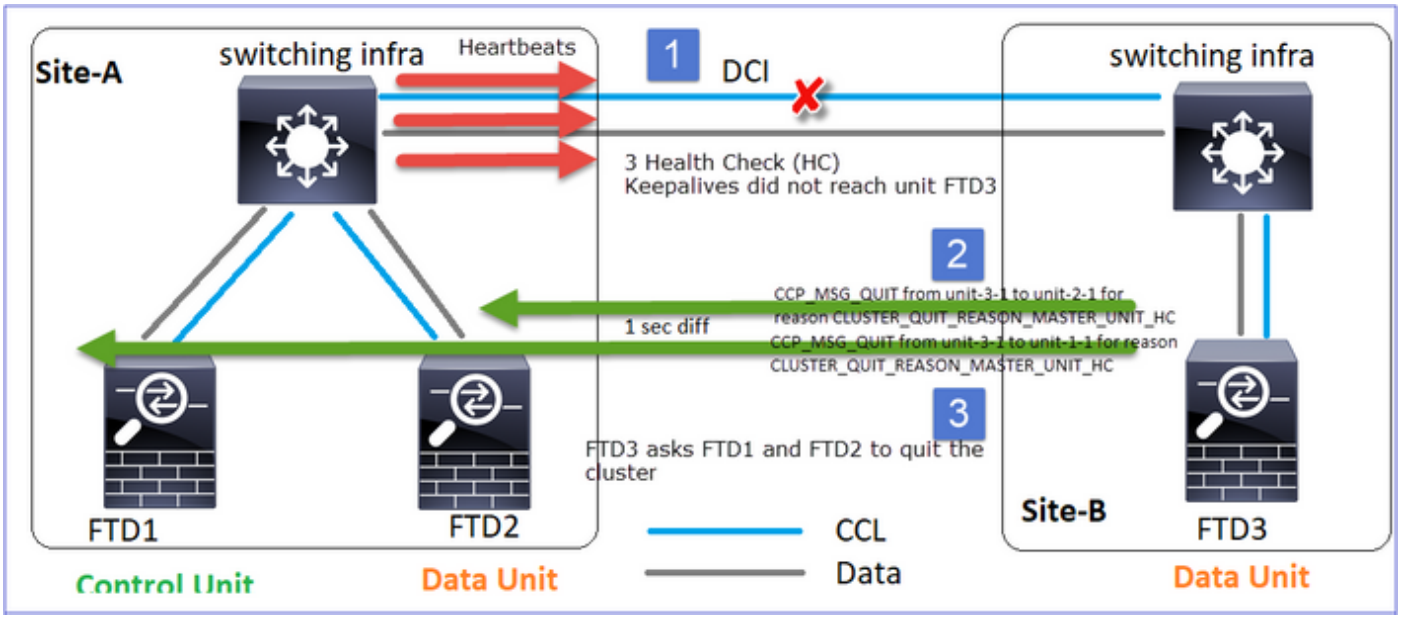
クラスタ制御ポイント(CCP)メッセージ

ハートビートメッセージに加えて、特定のシナリオでCCLを介して交換される多数のクラスタ制御メッセージがあります。一部はユニキャストメッセージで、他はブロードキャストです。

CLUSTER_QUIT_REASON_PRIMARY_UNIT_HC（クラスタ終了の理由_プライマリユニット_HC）

ユニットで制御ノードからの3つの連続するハートビートメッセージが失われるたびに、CCLを介してCLUSTER_QUIT_REASON_PRIMARY_UNIT_HCメッセージが生成されます。このメッセージの内容：

- ユニキャストです
- これは1秒の間隔で各ユニットに送信されます。
- ユニットがこのメッセージを受信すると、はクラスタを終了し(DISABLED)、再結合します。

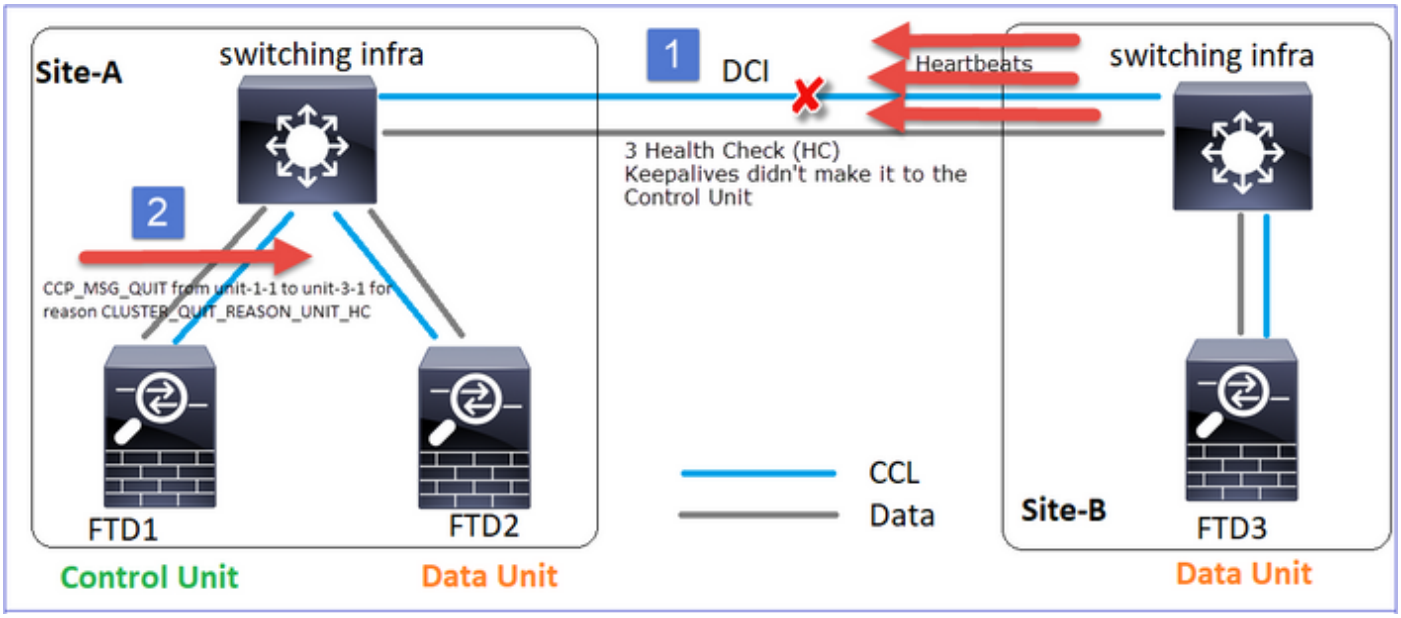


Q. CLUSTER_QUIT_REASON_PRIMARY_UNIT_HCの目的は何ですか。

A. unit-3-1(Site-B)から見ると、Unit-1-1とunit-2-1の両方への接続をSite Aから失うため、できるだけ早くメンバリストからこれらの接続を削除する必要があります。そうしないと、unit-2-1がメンバリストに残っていて、unit-2-1が接続のディレクタになっている場合、パケットが失われる可能性があり、unit-2-1へのフロークエリは失敗します。

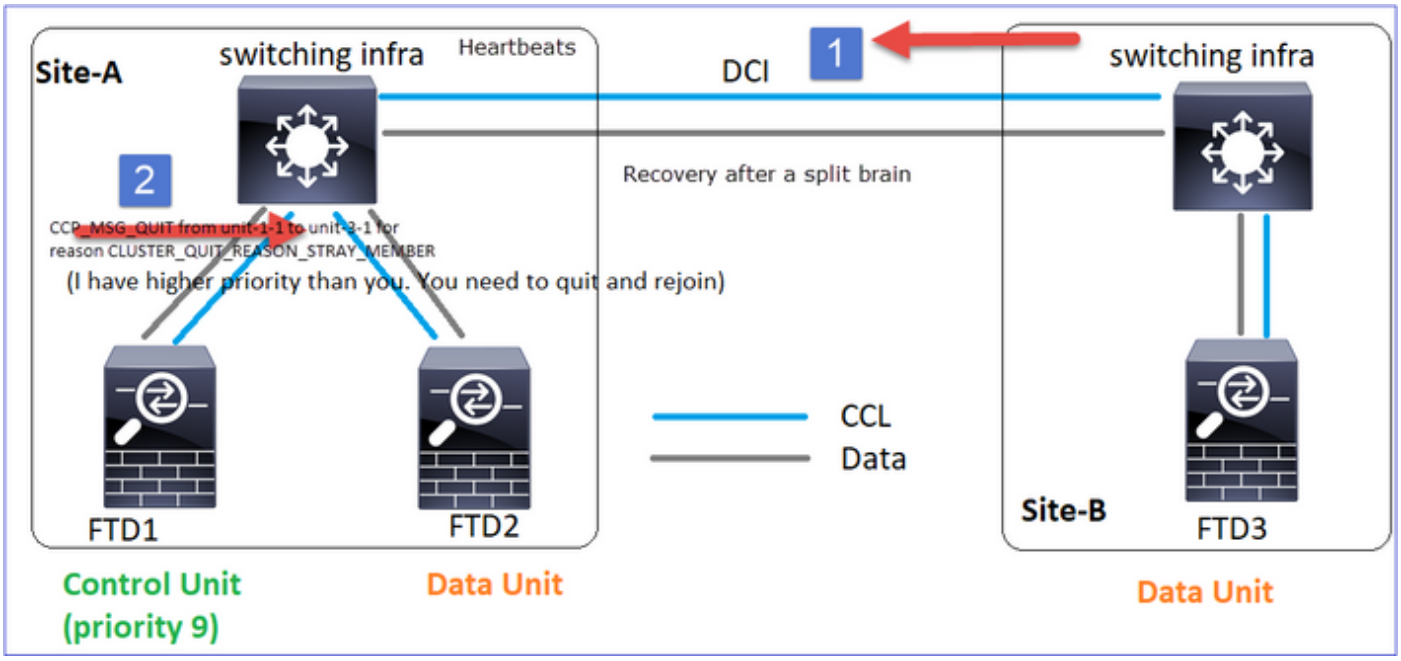
CLUSTER_QUIT_REASON_UNIT_HC (クラスタ終了の理由ユニット_HC)

制御ノードは、データノードから連続して3つのハートビートメッセージを失うと、CCL経由でCLUSTER_QUIT_REASON_UNIT_HCメッセージを送信します。このメッセージはユニキャストです。



CLUSTER_QUIT_REASON_STRAY_MEMBERです。

分割パーティションがピアパーティションと再接続すると、新しいデータノードは支配的な制御ユニット(CU)によって流用メンバとして扱われ、
CLUSTER_QUIT_REASON_STRAY_MEMBERの理由を含むCCP終了メッセージを受信します。



CLUSTER_QUIT_MEMBER_DROPOUTです。

データノードによって生成され、ブロードキャストとして送信されるブロードキャストメッセージ。ユニットでこのメッセージが受信されると、はDISABLEDステータスに移行します。さらに、自動再参加はキックオフしません。

```
<#root>

firepower#

show cluster info trace | include DROPOUT

Nov 04 00:22:54.699 [DBUG]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 [DBUG]Receive CCP message: CCP_MSG_QUIT from unit-3-1 to unit-2-1 for reason

CLUSTER_QUIT_MEMBER_DROPOUT
```

クラスタ履歴には次のように表示されます。

```
<#root>

PRIMARY       DISABLED      Received control message DISABLE (
```

```
member dropout announcement
)
```

Cluster Health-Check(HC)メカニズム

主な注意点

- 各クラスタユニットは、ヘルスチェック保留時間値の1/3ごとにハートビートを他のすべてのユニット（ブロードキャスト255.255.255.255）に送信し、CCL経由の転送としてUDPポート49495を使用します。
- 各クラスタユニットは、Pollタイマーと Pollカウント値を使用して、他のすべてのユニットを個別にトラッキングします。
- クラスタユニットが、ハートビート間隔内にクラスタピアユニットからパケット（ハートビートまたはデータパケット）を受信しなかった場合は、ポーリングカウントの値が増加します。
- クラスタピアユニットのポーリングカウント値が3になると、そのピアはダウンしていると見なされます。
- ハートビートを受信するたびにシーケンス番号がチェックされ、以前に受信したハートビートとの差分が1と異なる場合、それに応じてハートビートドロップカウンタが増加します。
- クラスタピアのPoll countカウンタが0以外の値で、パケットがピアによって受信されると、カウンタは0にリセットされます。

次のコマンドを使用して、クラスタのヘルスカウンタを確認します。

```
<#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)|       650|         0|      4999|         1|         0|
|            unit-3-1 ( 2)|       650|         0|      4999|         1|         0|
-------------------------------------------------------------------------------
```

メイン列の説明

カラム	説明
単位(ID)	リモートクラスタピアのID。

ハートビート数	CCLを介してリモートピアから受信したハートビートの数。
ハートビートのドロップ	失われたハートビートの数。このカウンタは、受信したハートビートシーケンス番号に基づいて計算されます。
平均ギャップ	受信したハートビートの平均時間間隔。
ポーリングカウント	このカウンタが3になると、ユニットはクラスタから削除されます。ポーリングクエリ間隔はハートビート間隔と同じですが、独立して実行されます。

カウンタをリセットするには、次のコマンドを使用します。

<#root>

firepower#

clear cluster info health details

Q.ハートビートの頻度を確認する方法は？

A.ギャップの平均値を確認します。

<#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|
----------------------------------------------------------------------------------
```

Q. FTDのクラスタ保留時間を変更するには、どうすればよいのですか。

A. FlexConfigの使用

Q.スプリットブレインの後にコントロールノードになるのは誰ですか？

A.プライオリティが最も高い（最も小さい番号）ユニット：

<#root>

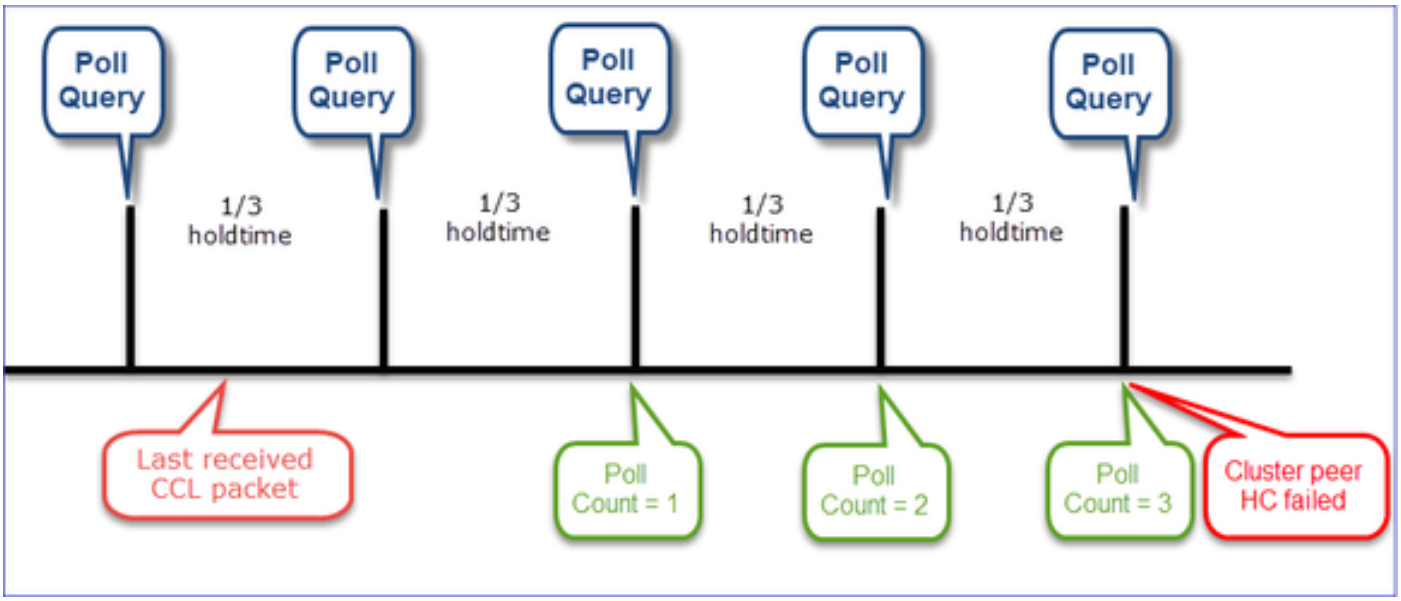
firepower#

show run cluster | include priority

priority 9

詳細については、HC障害シナリオ1を確認してください。

クラスタHC機構の可視化



指標タイマー：最小値と最大値は、最後に受信したCCLパケットの到達によって異なります。

Hold time	ポーリングクエリチェック (周波数)	最小検出時間	最大検出時間
3秒（デフォルト）	~ 1秒	~ 3.01秒	~ 3.99秒

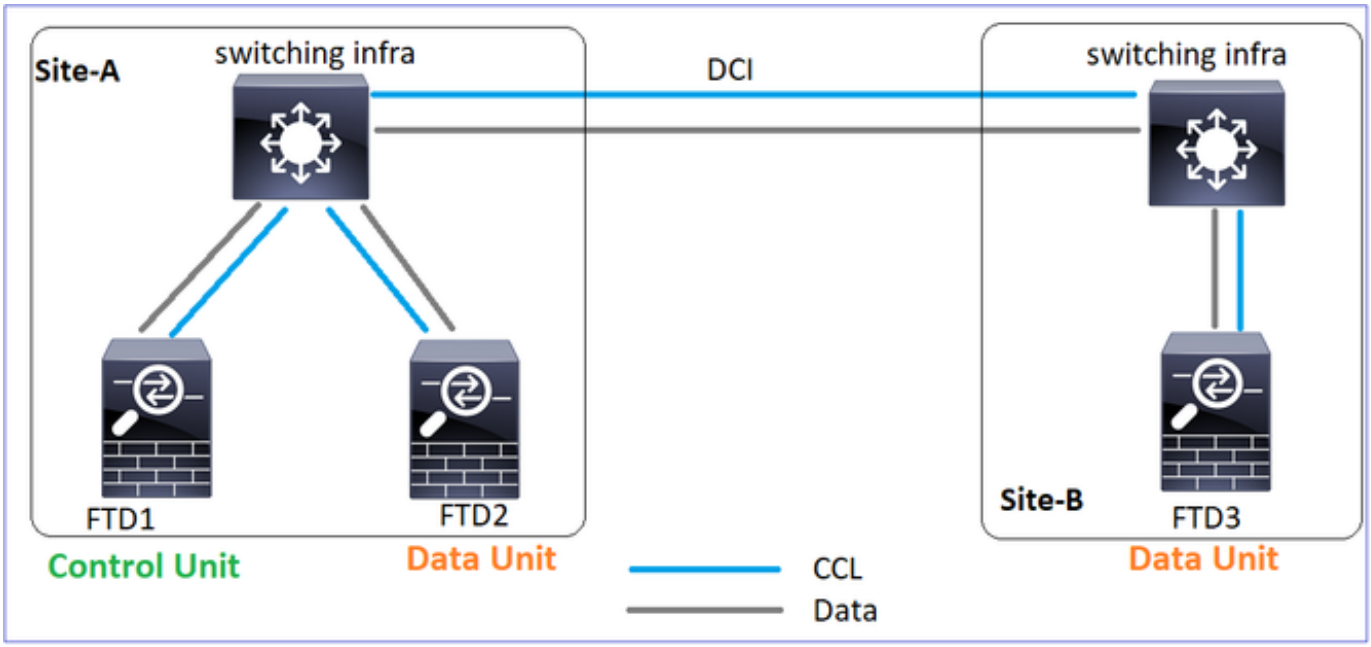
4 秒	~ 1.33秒	~ 4.01秒	~ 5.32秒
5 秒	~ 1.66秒	~ 5.01秒	~ 6.65秒
6 秒	~ 2秒	~ 6.01秒	~ 7.99秒
7 秒	~ 2.33秒	~ 7.01秒	~ 9.32秒
8 秒	~ 2.66秒	~ 8.01秒	~ 10.65秒

クラスタHCの障害シナリオ

このセクションの目的は次のデモンストレーションを行うことです。

- さまざまなクラスタHC障害シナリオ
- さまざまなログとコマンド出力を関連付ける方法

トポロジ



クラスタの設定

ユニット1-1	ユニット2-1
`cluster group GROUP1`	`cluster group GROU`

```
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
```

```
key *****
local-unit unit-2
cluster-interface
priority 17
health-check hold
health-check data
health-check clus
health-check syst
health-check moni
site-id 1
enable
```

クラスタステータス

ユニット1-1

```
<#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
```

ユニット2-1

```
<#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.c500.028
        Last join : 20:44:46 UTC
        Last leave: 20:44:38 UTC
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.c500.018
        Last join : 20:25:36 UTC
        Last leave: 20:25:28 UTC
```

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	**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.c500.038 Last join : 20:58:45 UTC Last leave: 20:58:37 UTC

シナリオ 1

両方向で最大4秒以上のCCL通信損失。

障害が発生する前

FTD1	FTD2	FTD3
サイトA	サイトA	サイトB
制御ノード	データノード	データノード

リカバリ後（ユニットの役割に変更なし）

FTD1	FTD2	FTD3
サイトA	サイトA	サイトB
制御ノード	データノード	データノード

分析

障害（CCL通信が失われました）。

ユニット3-1のデータプレーンコンソールメッセージ：

<#root>

```
firepower#
WARNING: dynamic routing is not supported on management interface when cluster interface-mode is 'spann
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.
```

ユニット1-1クラスタトレースログ：

<#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
[DBUG]Send CCP message to all: CCP_MSG_QUIT from unit-1-1 to unit-3-1 for reason CLUSTER_QUIT_MEMBER_DRO

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
[DBUG]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
```

スプリットブレイン

ユニット1-1	ユニット2-1

```
<#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-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
```

```
<#root>

firepower#

show cluster info


Cluster GROUP1: On
    Interface mode: spanned
    This is "unit-2-1" in state SE
        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 N
        Last leave: 20:44:38 UTC N
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 N
        Last leave: 20:25:28 UTC N
```

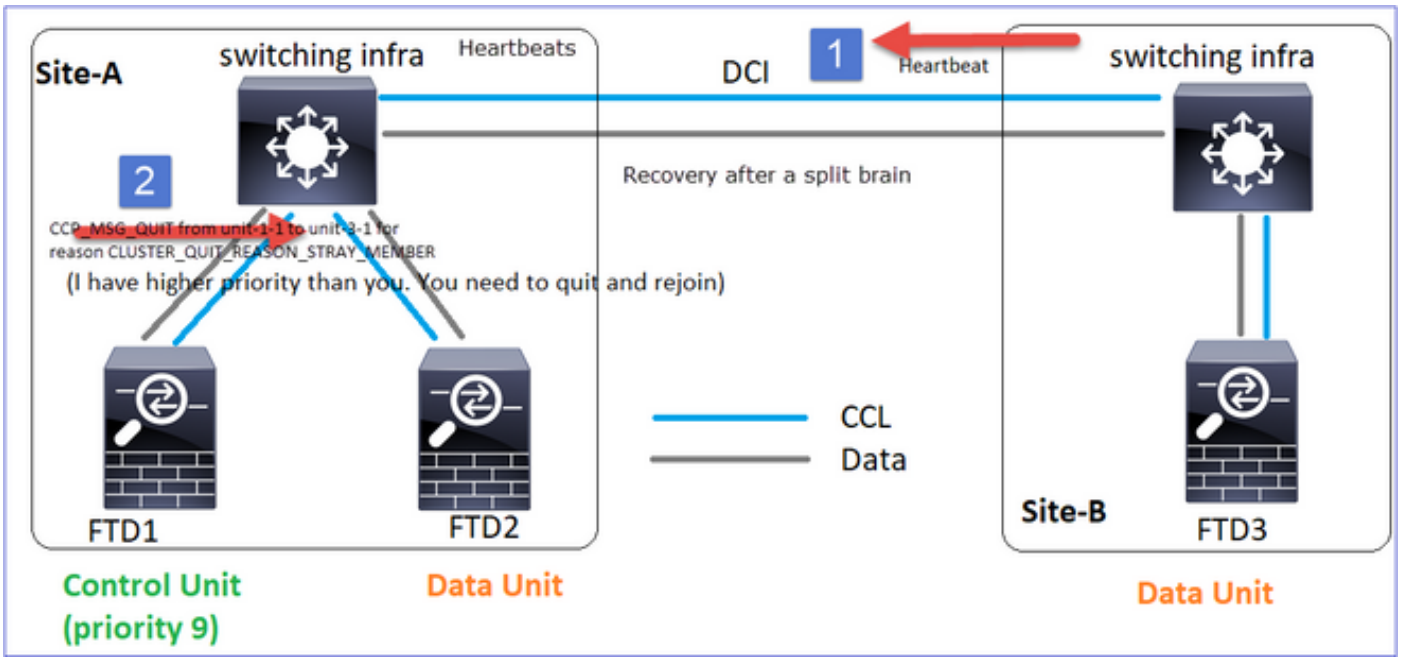
クラスタ履歴

ユニット1-1	ユニット2-1	ユニット3-1
イベントなし	イベントなし	<#root> 09:38:16 UTC Nov 2 2020 **SECONDARY PRIMARY_POST_CONFIG Primary relinquished rol** 09:38:17 UTC Nov 2 2020 PRIMARY_POST_CONFIG Primary Primary post config done and

CCL通信復旧

Unit-1-1は現在の制御ノードを検出し、Unit-1-1の方が優先順位が高いため、Unit-3-1にCLUSTER_QUIT_REASON_STRAY_MEMBERメッセージを送信して新しい選択プロセスをトリガーします。最後に、unit-3-1はデータノードとして再結合します。

分割パーティションがピアパーティションと再接続すると、データノードは支配的な制御ノードによって流用メンバとして扱われ、CLUSTER_QUIT_REASON_STRAY_MEMBERの理由を含むCCP終了メッセージを受信します。



<#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
```

両方のユニット（ユニット–1-1とユニット–3-1）がクラスタログに表示されます。

```
<#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
```

split-brainに対して生成されるsyslogメッセージもあります。

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

クラスタ履歴

ユニット1-1	ユニット2-1	ユニット3-1
イベントなし	イベントなし	<#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 messag 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 conf

```
09:48:29 UTC Nov 2 2020
SECONDARY_CONFIG          SECONDARY_FILESYS   Configuration replication
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
```

シナリオ 2

両方向で約3 ~ 4秒間のCCL通信損失。

障害が発生する前

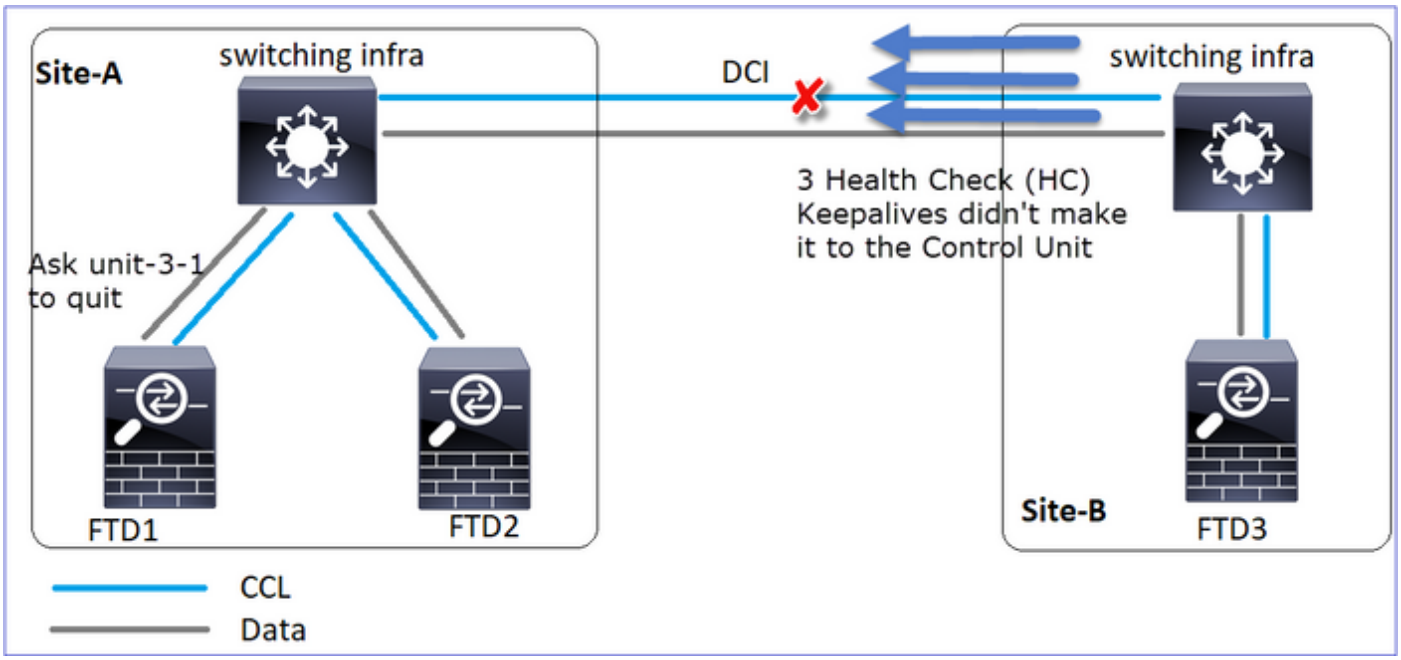
FTD1	FTD2	FTD3
サイトA	サイトA	サイトB
制御ノード	データノード	データノード

リカバリ後（ユニットの役割に変更なし）

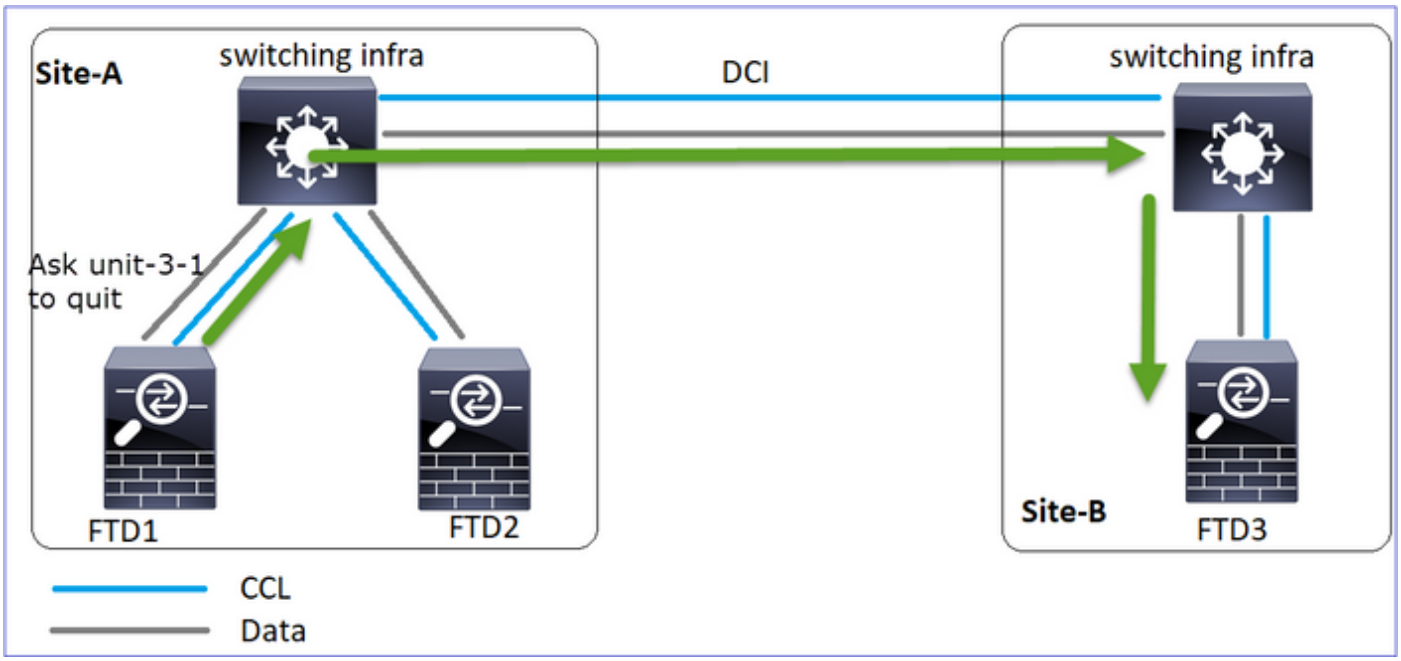
FTD1	FTD2	FTD3
サイトA	サイトA	サイトB
制御ノード	データノード	データノード

分析

イベント1：コントロールノードはユニット–3-1から3つのHCを失い、クラスタを離れるようユニット–3-1にメッセージを送信します。



イベント2: CCLが非常に速く回復し、制御ノードからの CLUSTER_QUIT_REASON_STRAY_MEMBERメッセージがリモート側に送信されました。ユニット3-1がDISABLEDモードに直接移行し、スプリットブレインが発生しない



ユニット1-1（コントロール）では次のように表示されます。

<#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

ユニット3-1（データノード）では、次のように表示されます。

```
<#root>

firepower#

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

クラスタユニット3-1がDISABLED状態に移行し、CCL通信が復元されると、データノードとして再結合します。

```
<#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
```

シナリオ 3

両方向で約3 ~ 4秒間のCCL通信損失。

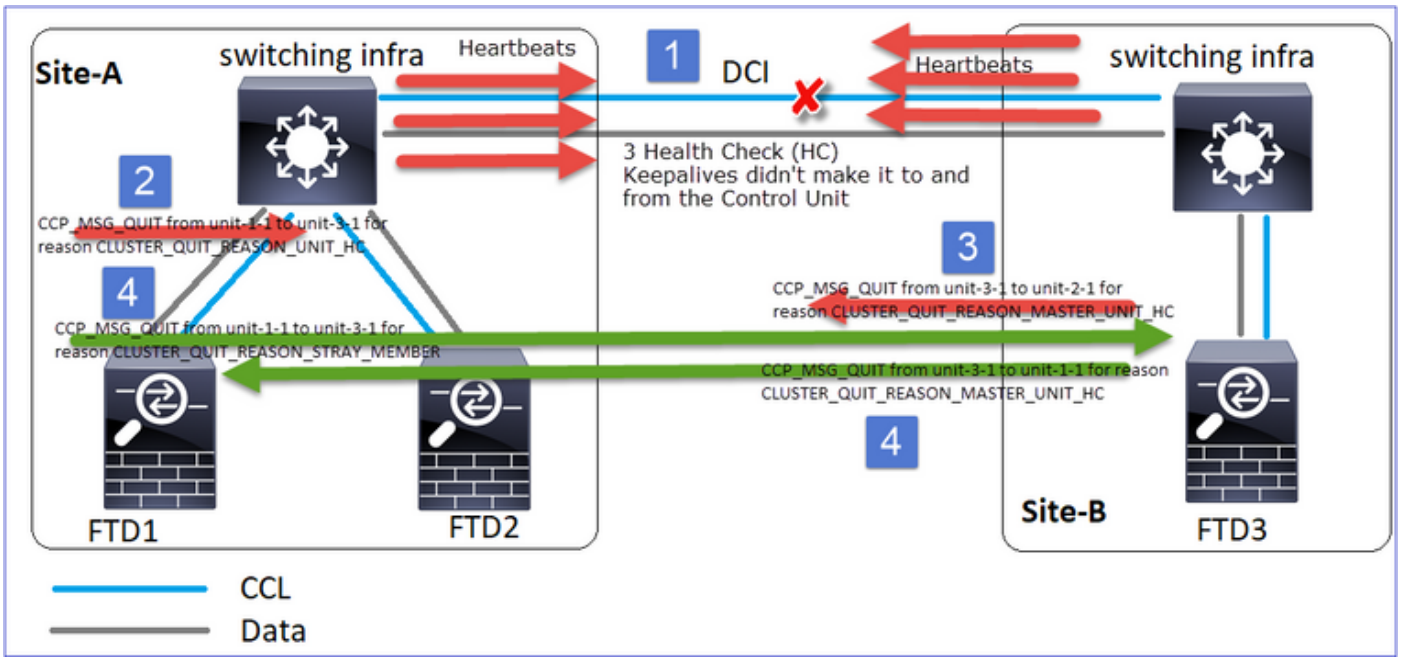
障害発生前。

FTD1	FTD2	FTD3
サイトA	サイトA	サイトB
制御ノード	データノード	データノード

リカバリ後（制御ノードが変更された場合）

FTD1	FTD2	FTD3
サイトA	サイトA	サイトB
データノード	制御ノード	データノード

分析



1. CCLがダウンします。
2. Unit-1-1はUnit-3-1から3つのHCメッセージを受信せず、QUITメッセージをUnit-3-1に送信します。このメッセージがユニット3-1に到達することはありません。
3. ユニット3-1がユニット2-1にQUITメッセージを送信します。このメッセージがユニット2-1に到達することはありません。

CCLが回復します。

4. Unit-1-1は、Unit-3-1が自身を制御ノードとしてアドバタイズしたことを確認し、

QUIT_REASON_STRAY_MEMBERメッセージをUnit-3-1に送信します。unit-3-1でメッセージが受信されると、DISABLED状態になります。同時に、unit-3-1はQUIT_REASON_PRIMARY_UNIT_HCメッセージをunit-1-1に送信し、終了を依頼します。ユニット1-1が受信すると、このメッセージはDISABLED状態になります。

クラスタ履歴

ユニット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
```

シナリオ 4

約3 ~ 4秒間のCCL通信損失

障害が発生する前

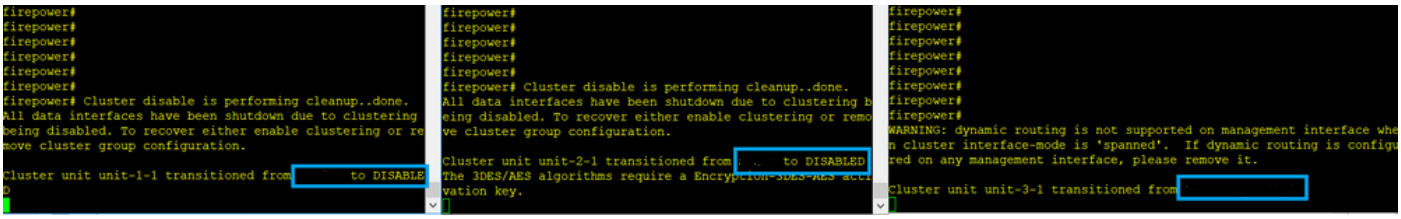
FTD1	FTD2	FTD3
サイトA	サイトA	サイトB
制御ノード	データノード	データノード

リカバリ後（制御ノードがサイトを変更）

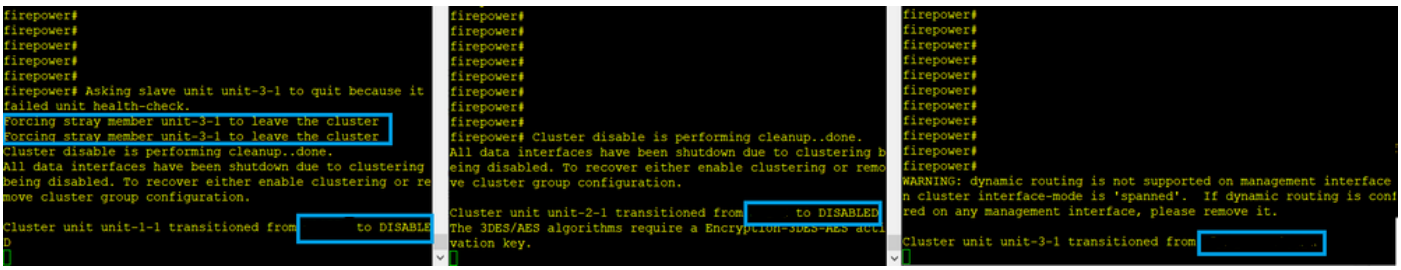
FTD1	FTD2	FTD3
サイトA	サイトA	サイトB
データノード	データノード	制御ノード

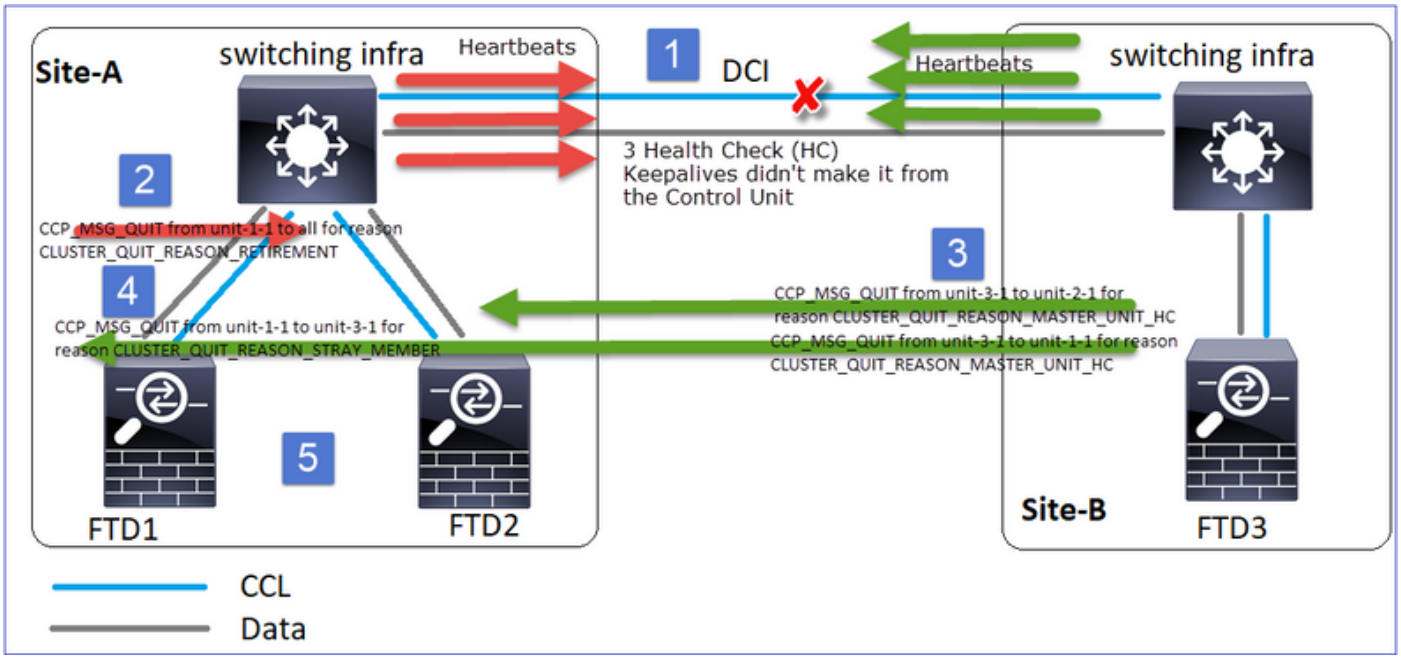
分析

障害

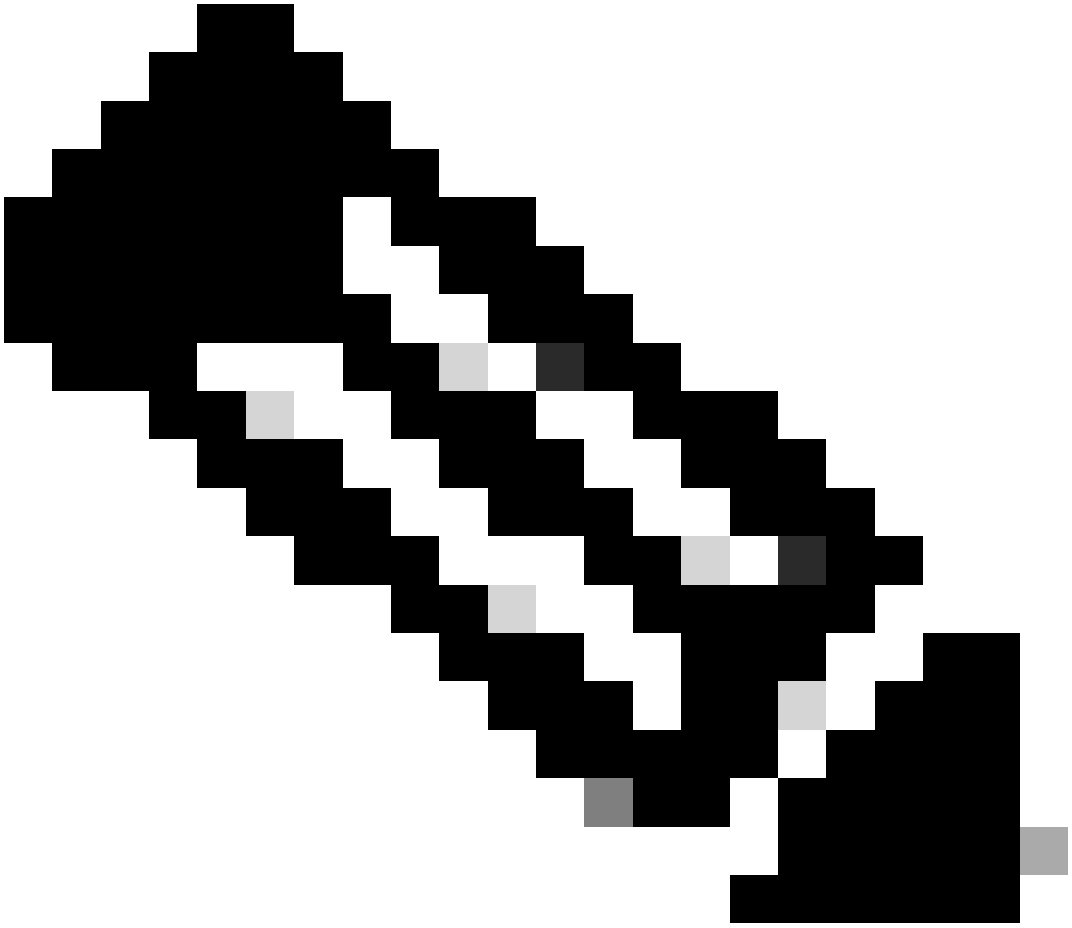


同じ障害の異なる種類。この場合、ユニット–1-1もユニット–3-1から3つのHCメッセージを受信せず、新しいキープアライブを受信した後、STRAYメッセージを使用してユニット–3-1をキックアウトしようとしましたが、メッセージはユニット–3-1に到達しませんでした。





1. CCLが数秒間単方向になります。ユニット–3-1は、ユニット–1-1から3つのHCメッセージを受信せず、制御ノードとなる。
2. ユニット2-1は、CLUSTER_QUIT_REASON_RETIREMENTメッセージ（ブロードキャスト）を送信する。
3. ユニット–3-1がユニット–2-1にQUIT_REASON_PRIMARY_UNIT_HCメッセージを送信します。ユニット2-1はこれを受信し、クラスタを終了します。
4. ユニット–3-1がユニット–1-1にQUIT_REASON_PRIMARY_UNIT_HCメッセージを送信します。ユニット1-1はそれを受信し、クラスタを終了します。CCLが回復します。
5. ユニット1-1と2-1は、データノードとしてクラスタに再参加します。



注：ステップ5でCCLが回復しない場合、サイトAではFTD1が新しい制御ノードになり、CCLの回復後に新しい選択が行われます。

ユニット1-1のsyslogメッセージ：

<#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
```

ユニット1-1のクラスタトレースログ：

<#root>

```
firepower#

show cluster info trace | include QUIT


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

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 [DBUG]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 [DBUG]Receive CCP message: CCP_MSG_QUIT from unit-2-1 for reason CLUSTER_QUIT_REASO
Nov 03 23:13:09.769 [DBUG]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 [DBUG]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 [DBUG]Send CCP message to id 1: CCP_MSG_QUIT from unit-1-1 to unit-3-1 for reason C
```

ユニット3-1のsyslogメッセージ：

<#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
```

クラスタ履歴

ユニット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

シナリオ 5

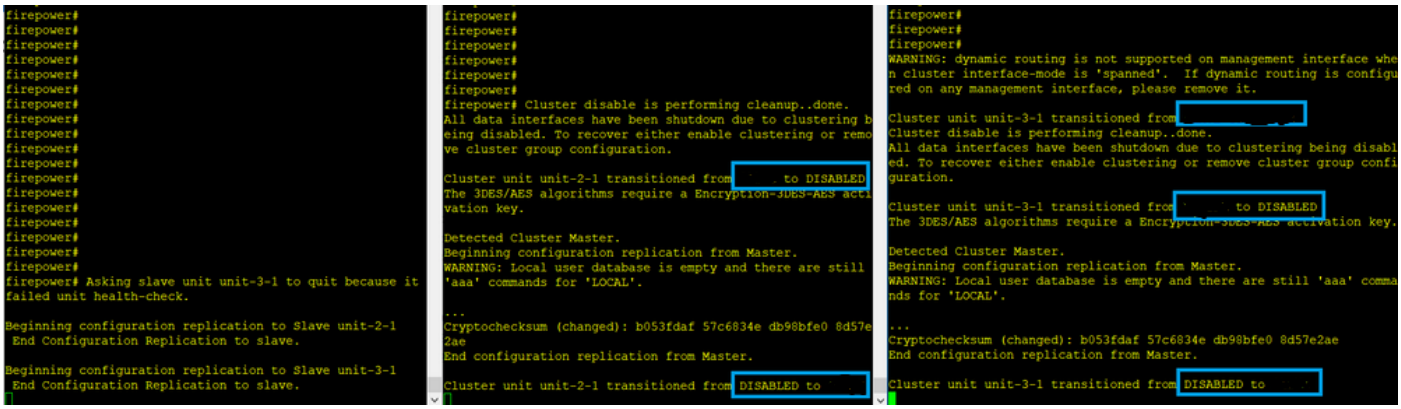
障害が発生する前

FTD1	FTD2	FTD3
サイトA	サイトA	サイトB
制御ノード	データノード	データノード

リカバリ後（変更なし）

FTD1	FTD2	FTD3
サイトA	サイトA	サイトB
制御ノード	データノード	データノード

障害



ユニット3-1はユニット1-1とユニット2-1の両方にQUITメッセージを送信しましたが、接続の問題によりユニット2-1のみがQUITメッセージを受信しました。

ユニット1-1クラスタトレースログ：

<#root>

firepower#

show cluster info trace | include QUIT

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

ユニット2-1クラスタトレースログ：

<#root>

firepower#

show cluster info trace | include QUIT

Nov 04 00:52:10.389 [DBUG]Receive CCP message: CCP_MSG_QUIT from unit-3-1 for reason CLUSTER_QUIT_REASO
Nov 04 00:51:47.019 [DBUG]Send CCP message to all: CCP_MSG_QUIT from unit-2-1 for reason CLUSTER_QUIT_R
Nov 04 00:51:46.999 [DBUG]

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 [DBUG]Receive CCP message: CCP_MSG_QUIT from unit-1-1 to unit-3-1 for reason CLUSTE

クラスタ履歴

ユニット1-1	ユニット2-1
イベントなし	<#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 configurati sync done 00:52:54 UTC Nov 4 2020 SECONDARY_CONFIG SECONDARY_FILESYS Configuration replication finishe 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

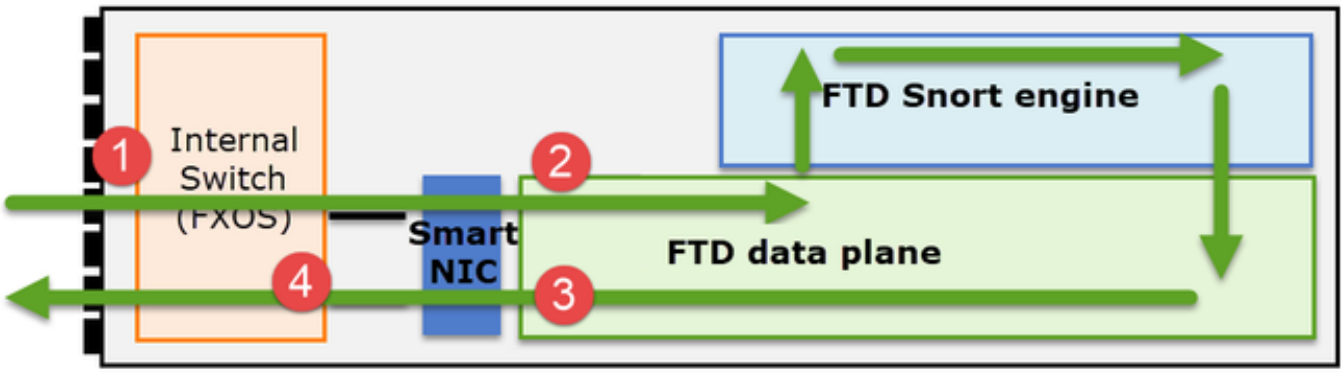
クラスタデータプレーン接続の確立

NGFWキャプチャポイント

NGFWは、次のポイントでキャプチャ機能を提供します。

- シャーシ内部スイッチ(FXOS)
- FTDデータプレーンエンジン
- FTD Snort エンジン

クラスタ上のデータパスの問題をトラブルシューティングする場合、ほとんどの場合に使用されるキャプチャポイントは、FXOSおよびFTDデータプレーンエンジンのキャプチャです。



1. 物理インターフェイスでのFXOS入力キャプチャ
2. データプレーンエンジンでのFTD入力キャプチャ
3. データプレーンエンジンでのFTD出力キャプチャ
4. バックプレーンインターフェイスでのFXOS入力キャプチャ

NGFWキャプチャの詳細については、次のドキュメントを参照してください。

クラスタユニットフローロールの基本

接続は、次のような要因に応じて、複数の方法でクラスタを介して確立できます。

- トラフィックのタイプ（TCP、UDPなど）
- 隣接スイッチで設定されたロードバランシングアルゴリズム
- ファイアウォールに設定された機能
- ネットワーク状態（IPフラグメンテーション、ネットワーク遅延など）

フローロール	説明	フラグ
主催者（Owner）	通常は、最初に接続を受信したユニット	UIO

Director	フォワーダからの所有者参照要求を処理するユニットです。	Y
バックアップ所有者	ダイレクタが所有者と同じユニットでない限り、ダイレクタはバックアップ所有者でもあります。オーナーが自分自身をダイレクタとして選択した場合は、別のバックアップ・オーナーが選択されます。	Y（ダイレクタがバックアップ・オーナーでもある場合） y（ダイレクタがバックアップ・オーナーでない場合）
フォワーダ	パケットを所有者に転送するユニット	z
フラグメントオーナー	フラグメント化されたトラフィックを処理するユニット	-
シャーシのバックアップ	シャーシ間クラスタでは、ディレクタ/バックアップフローと所有者フローの両方が同じシャーシのユニットによって所有されている場合、他のいずれかのシャーシ内のユニットがセカンダリバックアップ/ディレクタになります。 このロールは、ブレードが1つ以上あるFirepower 9300シリーズのシャーシ間クラスタに固有のものです。	w

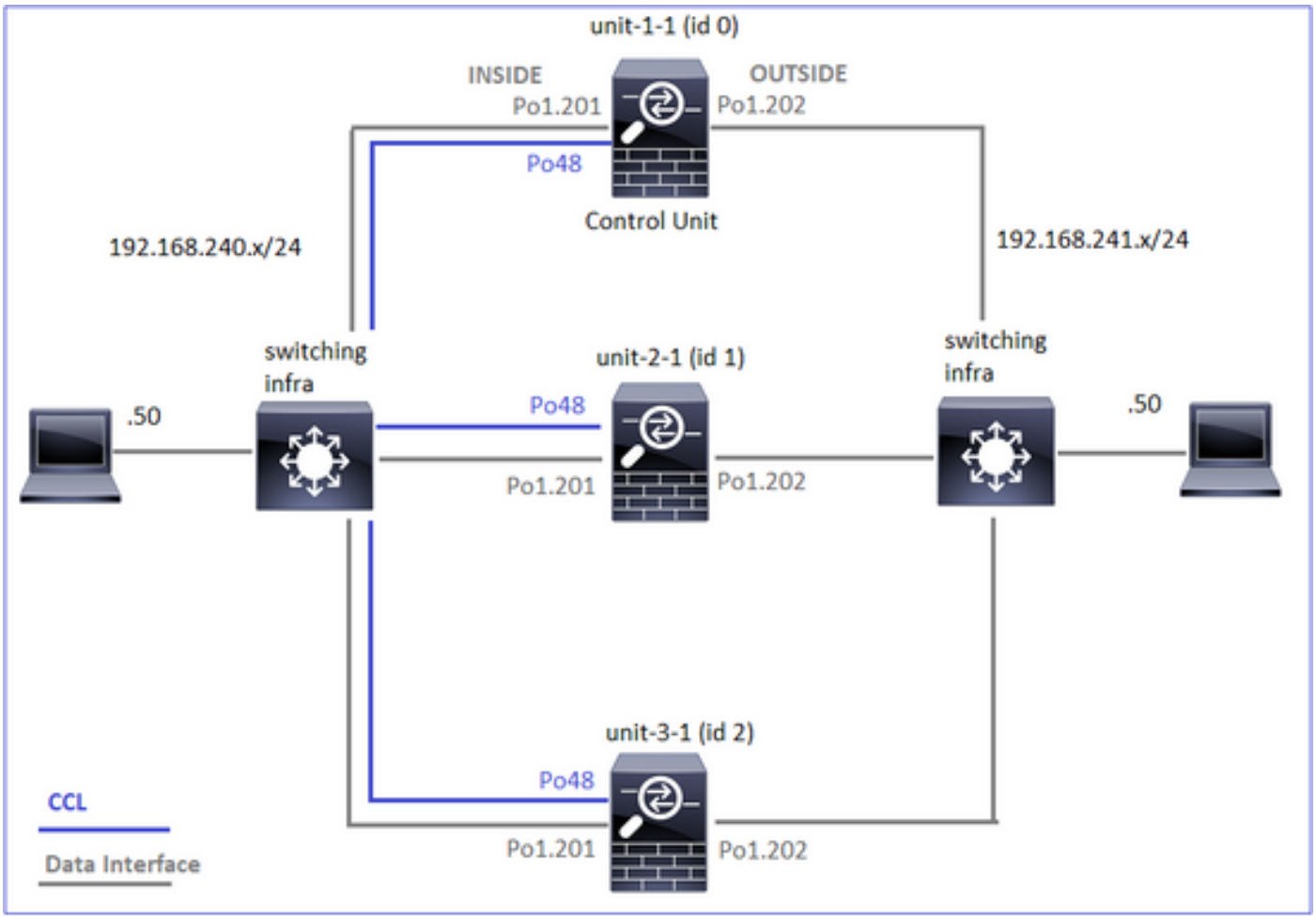
- 詳細については、『コンフィギュレーションガイド』の関連セクションを参照してください（「関連情報」のリンクを参照）
- 特定のシナリオ（ケーススタディのセクションを参照）では、一部のフラグが常に表示されるわけではありません。

クラスタ接続の確立のケーススタディ

次のセクションでは、クラスタを介して接続を確立する方法の一部を示すさまざまなケーススタディについて説明します。目標は次のとおりです。

- ユニットのさまざまな役割を理解します。
- さまざまなコマンド出力を関連付ける方法をデモンストレーションします。

トポロジ



クラスタユニットとID:

ユニット1-1	ユニット2-1
<#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	<#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

有効なクラスタキャプチャ：

```
cluster exec cap CAPI int INSIDE buffer 33554432 match tcp host 192.168.240.50 host 192.168.241.50 eq 8
cluster exec cap CAPO int OUTSIDE buffer 33554432 match tcp host 192.168.240.50 host 192.168.241.50 eq
cluster exec cap CAPI_RH reinject-hide int INSIDE buffer 33554432 match tcp host 192.168.240.50 host 19
cluster exec cap CAPO_RH reinject-hide int OUTSIDE buffer 33554432 match tcp host 192.168.240.50 host 1
cluster exec cap CCL int cluster buffer 33554432
```

注：これらのテストは、クラスタを通過するトラフィックが最小限のラボ環境で実行されました。実稼働環境では、キャプチャの「ノイズ」を最小限に抑えるために、可能な限り特定のキャプチャフィルタ（たとえば、宛先ポートと可能な限り送信元ポート）を使用するようにしてください。

ケース スタディ 1対称トラフィック（オーナーはディレクタでもある）

観察1.reinject-hideキャプチャは、ユニット1-1のパケットのみを示しています。これは、両方向のフローがユニット1-1（対称トラフィック）を通過したことを意味します。

<#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
```

観察2.送信元ポート45954を使用したフローの接続フラグ分析

<#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

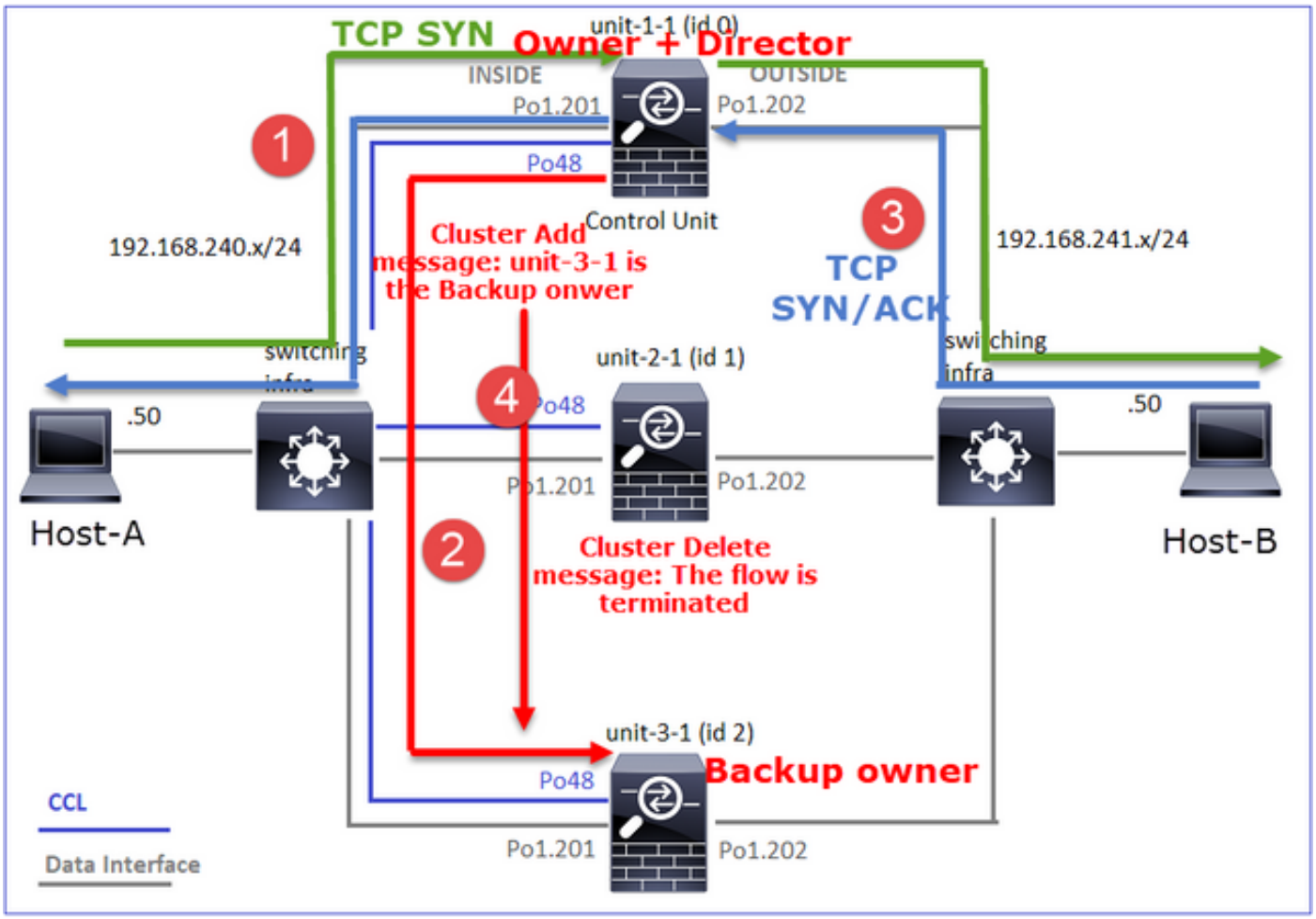
```
, idle 0:00:06, bytes 0,
```

flags y

ユニット	フラグ	注
ユニット1-1	UIO	· フロー所有者 – ユニットがフローを処理します · ダイレクタ：ユニット3-1には「Y」ではなく「Y」があるため、このフローのダイレクタとしてユニット1-1が選択されたことを示しています。したがって、このユニットは所有者でもあるため、別のユニット（この場合はユニット3-1）がバックアップ所有者として選出されています

ユニット2-1	-	-
ユニット3-1	y	ユニットはバックアップ所有者です

それを図で示します。



1. TCP SYNパケットがホストAからユニット1-1に到着します。ユニット1-1がフローオーナーになります。
2. ユニット1-1もフローディレクタとして選出されます（クラスタ追加メッセージ）。
3. TCP SYN/ACKパケットがホストBからユニット3-1に到着します。流れは対称です。
4. 接続が終了すると、所有者はクラスタ削除メッセージを送信して、バックアップ所有者からフロー情報を削除します。

観察3.トレース付きのキャプチャは、両方向がユニット1-1のみを通過することを示しています。

ステップ 1：送信元ポートに基づいて、すべてのクラスタユニットで対象となるフローとパケットを特定します。

<#root>

firepower#

```
cluster exec show capture CAPI | i 45954


unit-1-1(LOCAL):*******************************************************
1: 08:42:09.362697 802.1Q vlan#201 P0 192.168.240.50.45954 > 192.168.241.50.80: S 992089269:992089269(0)
2: 08:42:09.363521 802.1Q vlan#201 P0 192.168.241.50.80 > 192.168.240.50.45954: S 4042762409:4042762409
3: 08:42:09.363827 802.1Q vlan#201 P0 192.168.240.50.45954 > 192.168.241.50.80: . ack 4042762410 win 229
…
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 P0 192.168.240.50.45954 > 192.168.241.50.80: S 2732339016:2732339016
2: 08:42:09.363415 802.1Q vlan#202 P0 192.168.241.50.80 > 192.168.240.50.45954: S 3603655982:3603655982
3: 08:42:09.363903 802.1Q vlan#202 P0 192.168.240.50.45954 > 192.168.241.50.80: . ack 3603655983 win 229
…
unit-2-1:*******************************************************

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

ステップ 2：これはTCPフロートレースであるため、3ウェイハンドシェイクパケットをトレースします。この出力からわかるように、ユニット1-1が所有者です。わかりやすくするため、関連しないトレースフェーズは省略します。

<#root>

```
firepower#

show cap CAPI packet-number 1 trace


25985 packets captured
1: 08:42:09.362697 802.1Q vlan#201 P0 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


...
```

リターントラフィック(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
```

観察4.FTDデータプレーンのsyslogには、すべてのユニットでの接続の作成と終了が表示されます。

```
<#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
```

ケース スタディ 2対称トラフィック (ダイレクタと異なるオーナー)

- ケーススタディ#1と同じですが、このケーススタディでは、フローオーナーはディレクタとは別のユニットです。
- すべての出力はケーススタディ#1に類似しています。ケーススタディ#1との主な違いは、シナリオ1の「y」フラグを置き換える「Y」フラグです。

観察1.オーナーはディレクターとは異なります。

送信元ポート46278を使用したフローの接続フラグ分析

<#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:

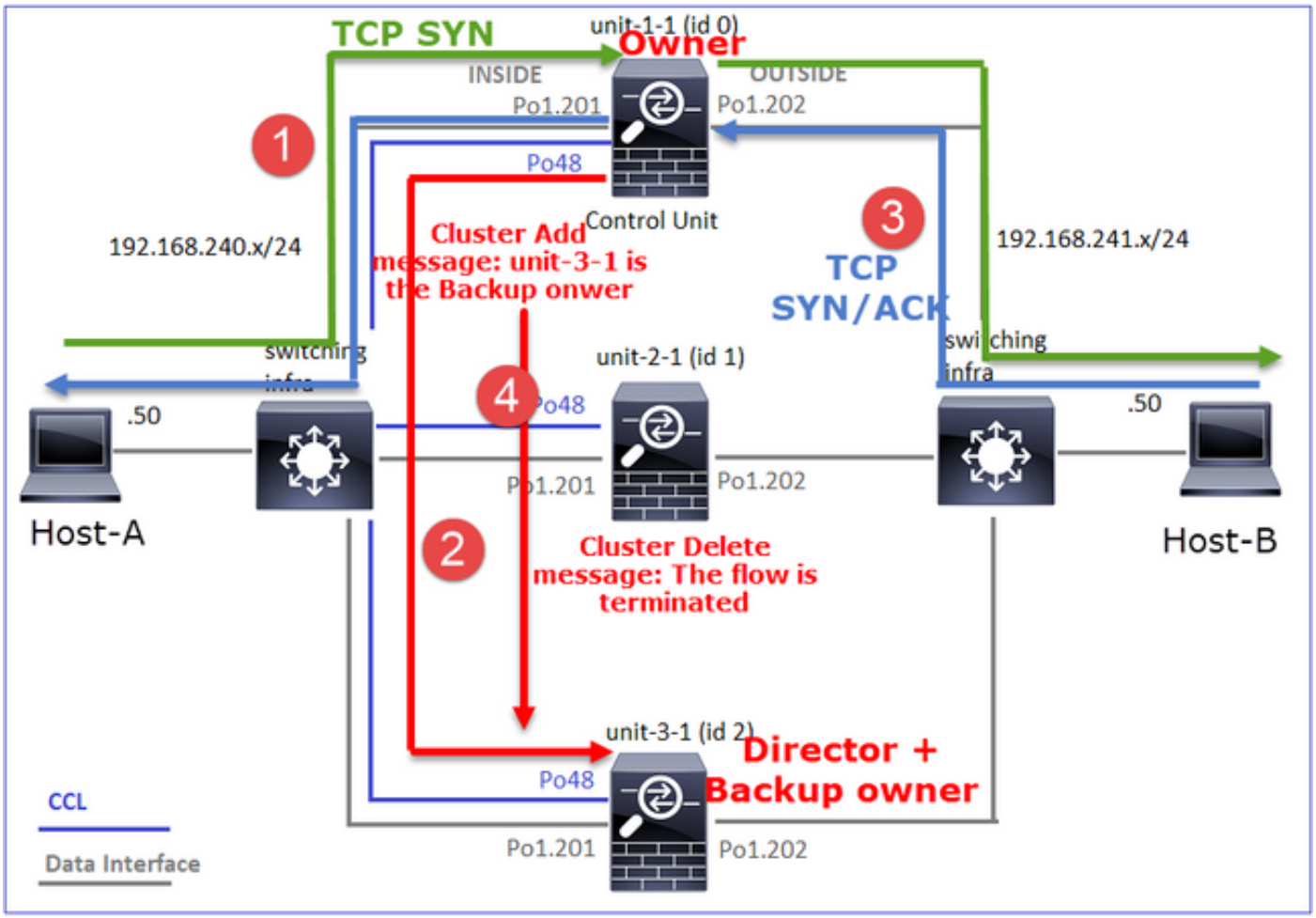
46278

, idle 0:00:06, bytes 0,

flags Y

ユニット	フラグ	注
ユニット1-1	UIO	・ フロー所有者 – ユニットがフローを処理します
ユニット2-1	-	-
ユニット3-1	Y	・ ダイレクタとバックアップ・オーナー：ユニット3-1のフラグはY（ダイレクタ）です。

それを図で示します。



1. TCP SYNパケットがホストAからユニット1-1に到着します。ユニット1-1がフローオーナーになります。
2. Unit-3-1がフローディレクタに選出されるユニット3-1はバックアップ所有者でもあります（CCL上のUDP 4193の「cluster add」メッセージ）。
3. TCP SYN/ACKパケットがホストBからユニット3-1に到着します。流れは対称です。
4. 接続が終了すると、所有者はCCLを介してUDP 4193で「cluster delete」メッセージを送信し、バックアップ所有者からフロー情報を削除します。

観察2.トレース付きのキャプチャは、両方向がユニット1-1のみを通過することを示しています

ステップ 1：ケーススタディ1と同じアプローチを使用して、送信元ポートに基づいてすべてのクラスタユニットの対象フローとパケットを特定します。

<#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 P0 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 P0 192.168.240.50.46278 > 192.168.241.50.80: . ack 3524167696 win 229
…
unit-2-1:*************************************************************

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

OUTSIDEインターフェイスでのキャプチャ：

<#root>

```
firepower#

cluster exec show cap CAPO | include 46278



unit-1-1

(LOCAL):*******************************************************
3: 11:01:44.841921 802.1Q vlan#202 P0 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 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>
5: 11:01:44.842638 802.1Q vlan#202 P0 192.168.240.50.46278 > 192.168.241.50.80: . ack 3382481338 win 22
```

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

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

ステップ 2：入力パケット（TCP SYNおよびTCP SYN/ACK）に注目します。

<#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

ユニット1-1のSYN/ACKをトレースします。

```
<#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
```

観察3.FTDデータプレーンのsyslogには、所有者とバックアップ所有者の接続の作成と終了が表示されます。

```
<#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.
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
```

ケース スタディ 3非対称トラフィック（ディレクタがトラフィックを転送）。

観察1.reinject-hideキャプチャは、ユニット1-1（非対称フロー）とユニット2-1（非対称フロー）のパケットを示しています。

<#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 CAPO_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
```

観察2.送信元ポート46502を使用したフローの接続フラグ分析

<#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,
```

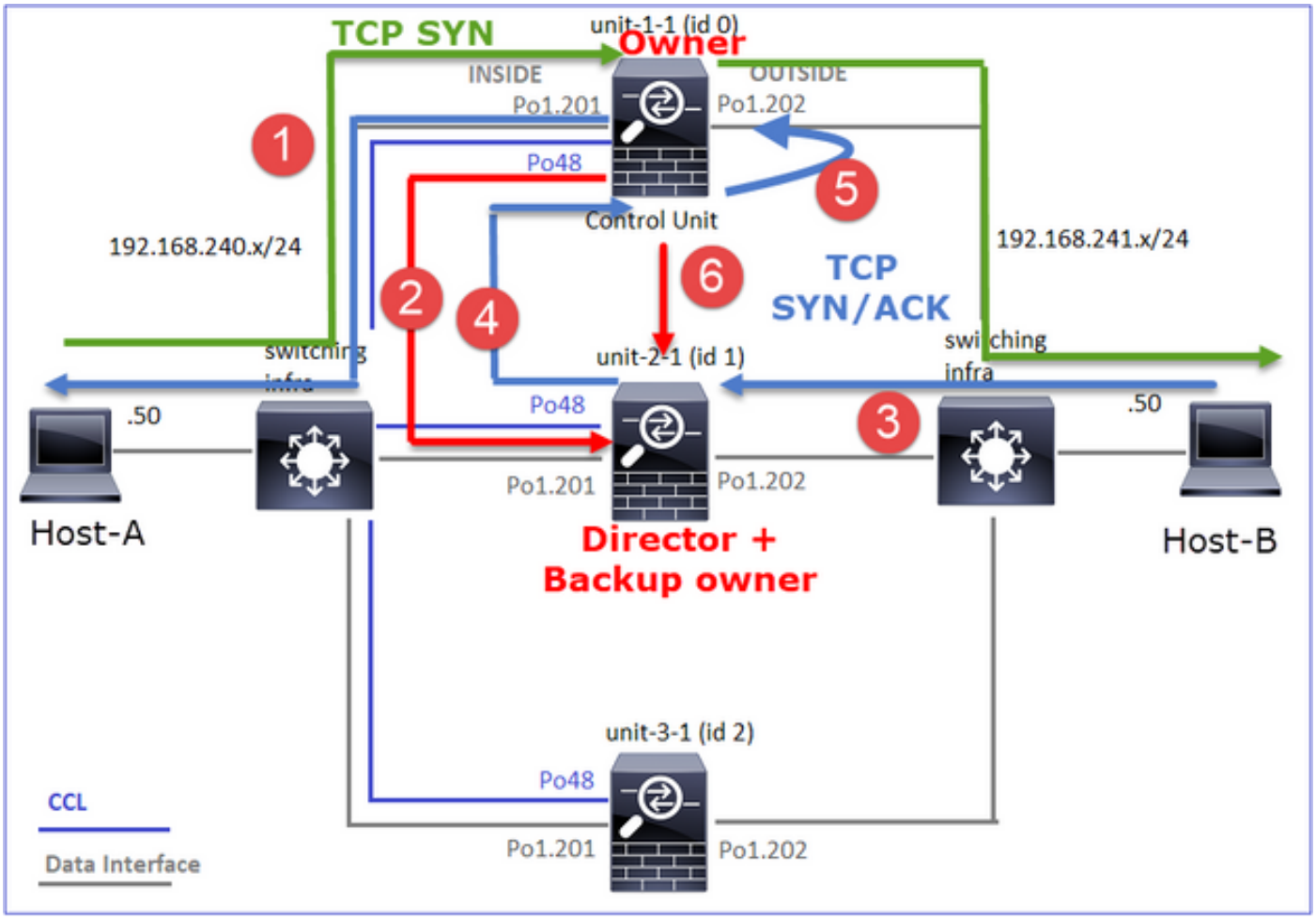
flags Y

```
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
```

ユニット	フラグ	注
ユニット1-1	UIO	· フローオーナー – ユニットがフローを処理します。
ユニット2-1	Y	· ダイレクタ：unit-2-1にはフラグ「Y」があるため、このフローのダイレクタとしてunit-2-1が選択されたことを示しています。 · バックアップ· オーナー · 最後に、この出力から明らかではありませんが、show captureおよびshow logの出力から、ユニット2-1がこのフローを所有者に転送していることがわかります（技術的には、このシナリオではフォワーダとは見なされません）。 注：1つのユニットをディレクタ（Yフロー）とフォワーダ（Zフロー）の両方にすることはできない。これらの2つのロールは相互に排他的である。ダイレクタ（Yフロー）は引き続きトラフィックを転送できます。show logの出力については、このケーススタディの後半を参

		照してください。
ユニット3-1	-	-

それを図で示します。



1. TCP SYNパケットがホストAからユニット1-1に到着します。ユニット1-1がフローオーナーになります。
2. ユニット2-1がフローディレクタとバックアップの所有者に選出されるフローオーナーは「cluster add」ユニキャストメッセージをUDP 4193で送信し、バックアップのオーナーにフローについて通知します。
3. TCP SYN/ACKパケットがホストBからユニット2-1に到着します。フローは非対称です。
4. ユニット2-1は、（TCP SYN Cookieにより）CCL経由でパケットを所有者に転送します。
5. 所有者はインターフェイスOUTSIDEでパケットを再注入してから、パケットをホストAに転送します。
6. 接続が終了すると、所有者はクラスタ削除メッセージを送信して、バックアップ所有者からフロー情報を削除します。

観察3.トレースを使用したキャプチャは、非対称トラフィックと、ユニット–2-1からユニット–1-1へのリダイレクションを示しています。

ステップ 1：対象のフロー(ポート46502)に属するパケットを特定します。

```
<#root>

firepower#

cluster exec show capture CAPI | include 46502


unit-1-1(LOCAL):*******************************************************
3: 12:58:33.356121 802.1Q vlan#201 P0 192.168.240.50.46502 > 192.168.241.50.80: S 4124514680:4124514680
4: 12:58:33.357037 802.1Q vlan#201 P0 192.168.241.50.80 > 192.168.240.50.46502: S 883000451:883000451(0)
5: 12:58:33.357357 802.1Q vlan#201 P0 192.168.240.50.46502 > 192.168.241.50.80: . ack 883000452 win 229
unit-2-1:*************************************************************

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

戻り値の方向：

```
<#root>

firepower#

cluster exec show capture CAPO | include 46502


unit-1-1(LOCAL):*******************************************************
3: 12:58:33.356426 802.1Q vlan#202 P0 192.168.240.50.46502 > 192.168.241.50.80: S 1434968587:1434968587
4: 12:58:33.356915 802.1Q vlan#202 P0 192.168.241.50.80 > 192.168.240.50.46502: S 4257314722:4257314722
5: 12:58:33.357403 802.1Q vlan#202 P0 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 P0 192.168.241.50.80 > 192.168.240.50.46502: S 4257314722:4257314722
2: 12:58:33.360302 802.1Q vlan#202 P0 192.168.241.50.80 > 192.168.240.50.46502: . ack 1434968736 win 23
3: 12:58:33.361004 802.1Q vlan#202 P0 192.168.241.50.80 > 192.168.240.50.46502: . 4257314723:4257316091
…

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

ステップ 2：パケットをトレースします。デフォルトでは、最初の50個の入力パケットのみがトレースされます。わかりやすくするために、関連しないトレースフェーズは省略します。

ユニット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 P0 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

ユニット2-1 (フォワーダ)

リターントラフィック(TCP SYN/ACK)。 対象のユニットは、ダイレクタ/バックアップの所有者であり、所有者にトラフィックを転送するユニット2-1です。

<#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).

観察4.FTDデータプレーンのsyslogには、すべてのユニットでの接続の作成と終了が表示されます。

<#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.
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 (192.168.240.50/4650
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 Forwa
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#
```

ケース スタディ 4非対称トラフィック（オーナーはダイレクタ）

観察1.reinject-hideキャプチャは、ユニット1-1（非対称フロー）とユニット2-1（非対称フロー）のパケットを示しています。

<#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 CAPO_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
```

観察2.送信元ポート46916を使用したフローの接続フラグ分析

<#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:
```

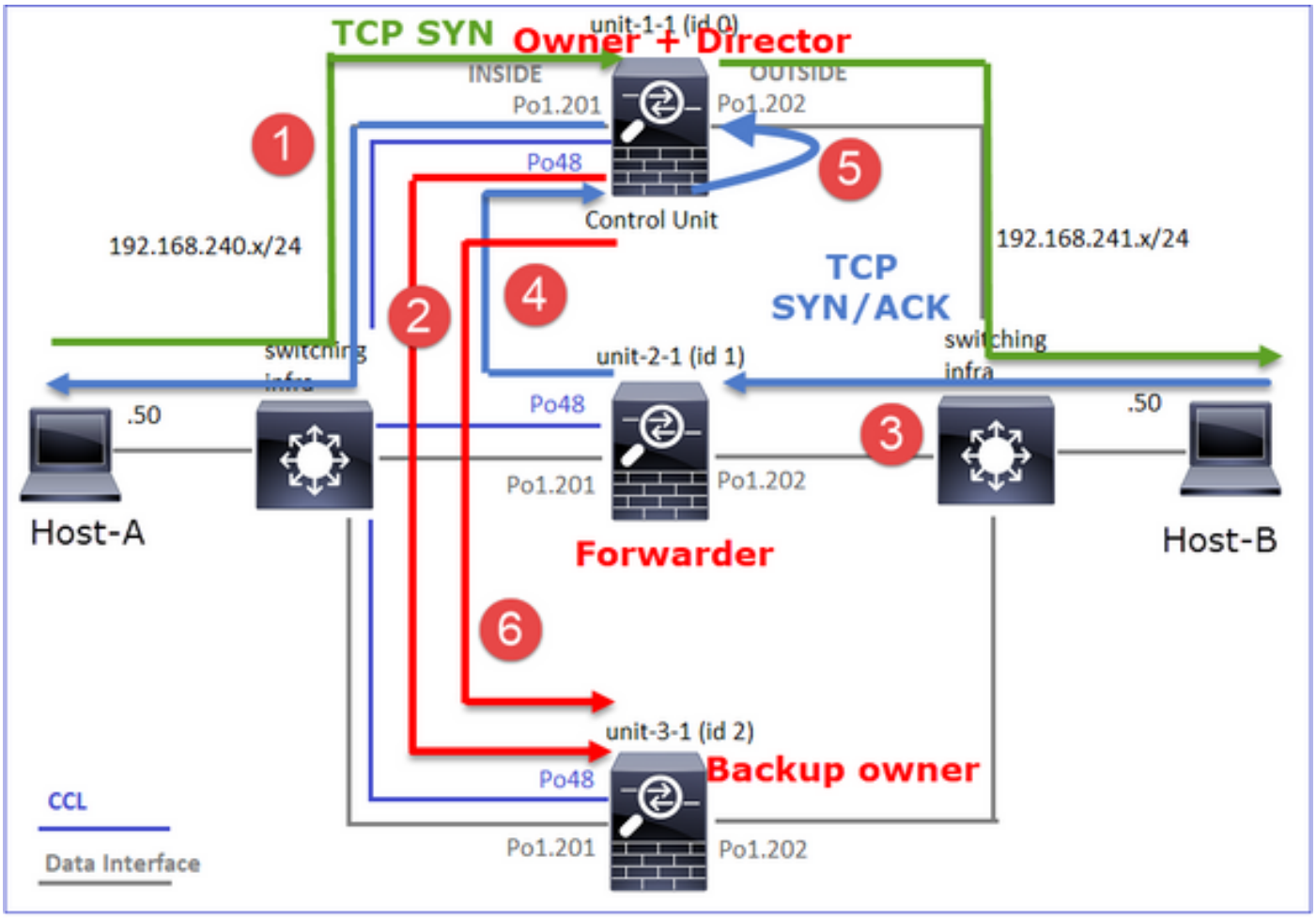
46916

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

flags y

ユニット	フラグ	注
ユニット1-1	UIO	· フロー所有者 – ユニットがフローを処理します · ダイレクタ：ユニット3-1には「Y」ではなく「Y」があるため、このフローのダイレクタとしてユニット1-1が選択されたことを示しています。したがって、このユニットは所有者でもあるため、別のユニット（この場合はユニット3-1）がバックアップ所有者として選出されています
ユニット2-1	z	· フォワーダ
ユニット3-1	y	– バックアップ所有者

それを図で示します。



1. TCP SYNパケットがホストAからユニット1-1に到着します。Unit-1-1がフローオーナーになり、ディレクタとして選出されます。
2. ユニット3-1がバックアップ所有者として選出されます。フローオーナーはユニキャストの「cluster add」メッセージをUDP 4193で送信し、バックアップのオーナーにフローを通知します。
3. TCP SYN/ACKパケットがホストBからユニット2-1に到着します。フローは非対称です。
4. ユニット2-1は、（TCP SYN Cookieにより）CCL経由でパケットを所有者に転送します。
5. 所有者はインターフェイスOUTSIDEでパケットを再注入してから、パケットをホストAに転送します。
6. 接続が終了すると、所有者はクラスタ削除メッセージを送信して、バックアップ所有者からフロー情報を削除します。

観察3.トレースを使用したキャプチャは、非対称トラフィックと、ユニット–2-1からユニット–1-1へのリダイレクションを示しています。

ユニット2-1（フォワーダ）

```
<#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).

観察4.FTDデータプレーンのsyslogには、すべてのユニットでの接続の作成と終了が表示されます。

- ユニット1-1（所有者）
- ユニット2-1（フォワーダ）
- ユニット3-1（バックアップオーナー）

<#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
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 (192.168.240.50/4691
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
```

ケース スタディ 5非対称トラフィック（オーナーがディレクタと異なる）

観察1.reinject-hideキャプチャは、ユニット1-1（非対称フロー）とユニット2-1（非対称フロー）のパケットを示しています。

<#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
```

観察2.送信元ポート46994を使用したフローの接続フラグ分析:

<#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:

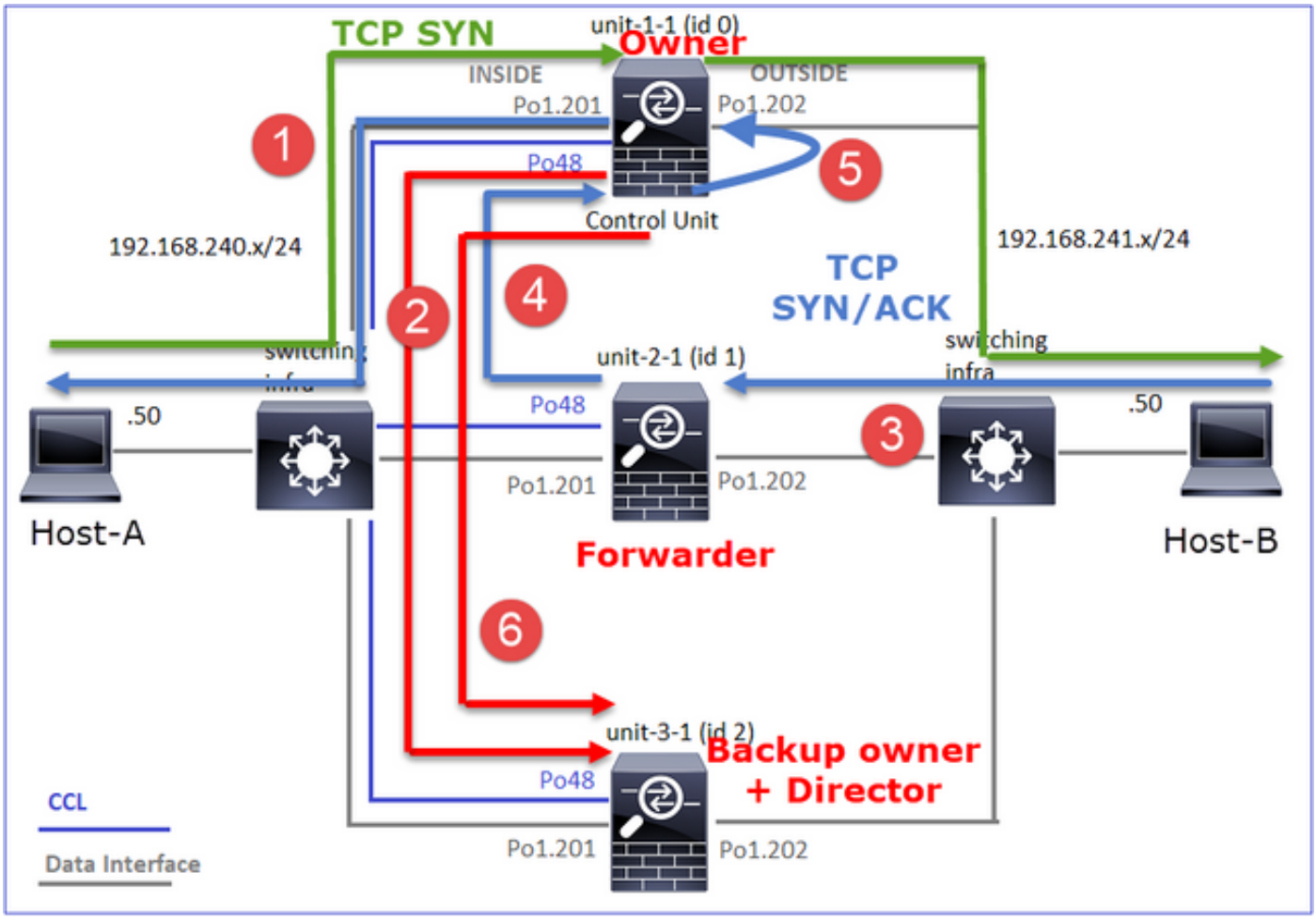
46994

, idle 0:00:05, bytes 0,

flags Y

ユニット	フラグ	注
ユニット1-1	UIO	· フロー所有者 – ユニットがフローを処理します
ユニット2-1	z	· フォワーダ
ユニット3-1	Y	· バックアップ· オーナー ·Director

それを図で示します。



1. TCP SYNパケットがホストAからユニット1-1に到着します。ユニット1-1がフローオーナーになります。
2. ユニット3-1がディレクタおよびバックアップ所有者として選出されます。フローオーナー

は「cluster add」ユニキャストメッセージをUDP 4193で送信し、バックアップのオーナーにフローについて通知します。
3. TCP SYN/ACKパケットがホストBからユニット2-1に到着します。フローは非対称です
4. ユニット2-1は、（TCP SYN Cookieにより）CCL経由でパケットを所有者に転送します。
5. 所有者はインターフェイスOUTSIDEでパケットを再注入してから、パケットをホストAに転送します。
6. 接続が終了すると、所有者はクラスタ削除メッセージを送信して、バックアップ所有者からフロー情報を削除します。

観察3.トレースを使用したキャプチャは、非対称トラフィックと、ユニット–2-1からユニット–1-1へのリダイレクションを示しています。

ユニット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
```

ユニット2-1（フォワーダ）

<#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).

観察4.FTDデータプレーンのsyslogには、すべてのユニットでの接続の作成と終了が表示されます。

- ユニット1-1（所有者）
- ユニット2-1（フォワーダ）
- ユニット3-1（バックアップ・オーナー/ダイレクタ）

<#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

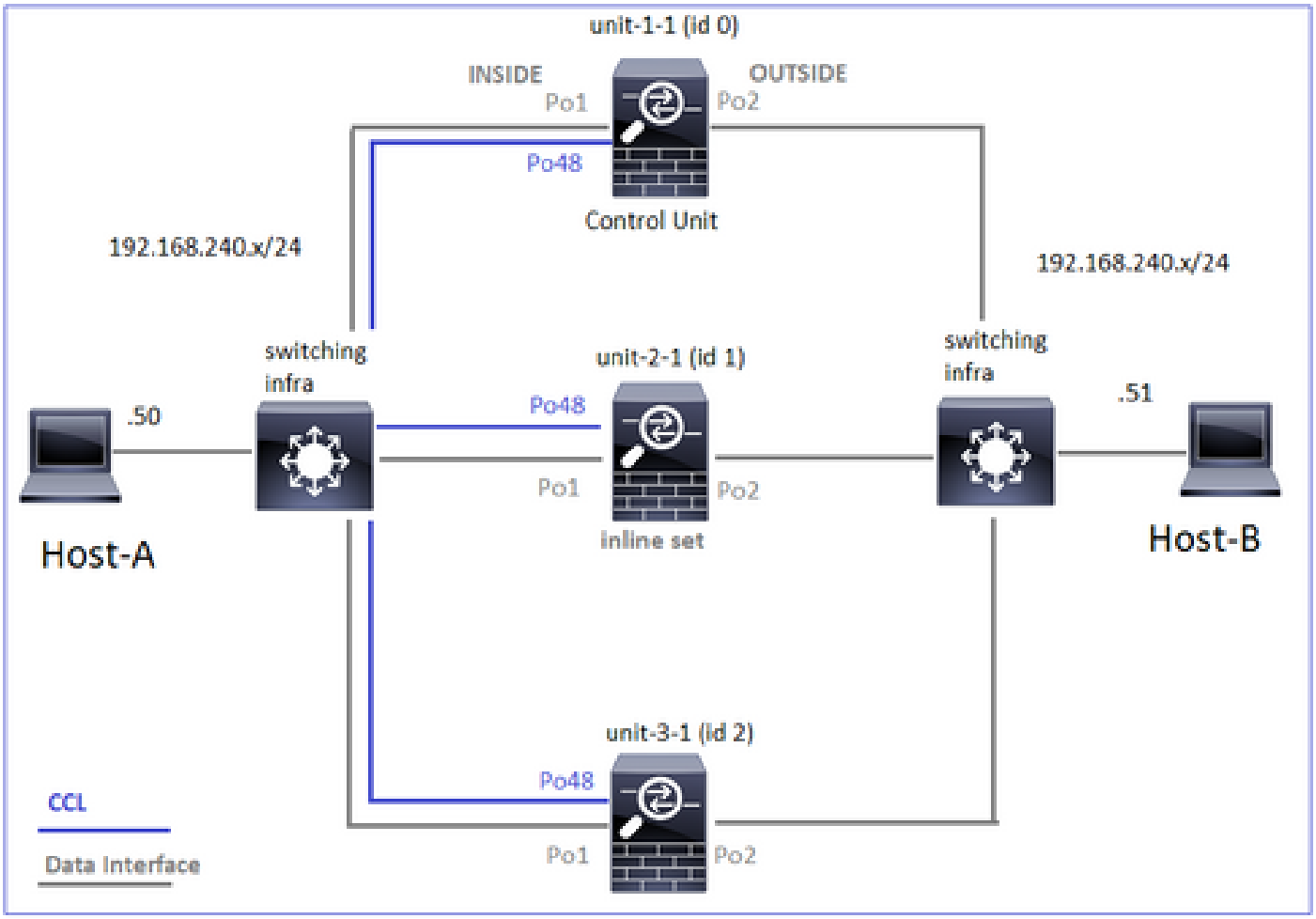
 10080 for INSIDE:192.168.240.50/46994 (192.168.240.50/46994) to OUTSIDE:192.168.241.50/80 (192.168.241
Dec 01 2020 16:46:53: %FTD-6-302014:

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 (192.168.240.50/4699
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
```

次のケーススタディでは、使用されるトポロジはインラインセットを持つクラスタに基づいています。



ケース スタディ 6非対称トラフィック (インライン・ セット、所有者はダイレクタ)

観察1.reinject-hideキャプチャは、ユニット1-1（非対称フロー）とユニット2-1（非対称フロー）のパケットを示しています。 さらに、所有者はユニット2-1です（reinject-hideキャプチャのINSIDEインターフェイスとOUTSIDEインターフェイスの両方にパケットがありますが、ユニット1-1はOUTSIDEのみにパケットがあります）。

<#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

観察2.送信元ポート51844を使用したフローの接続フラグ分析

<#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

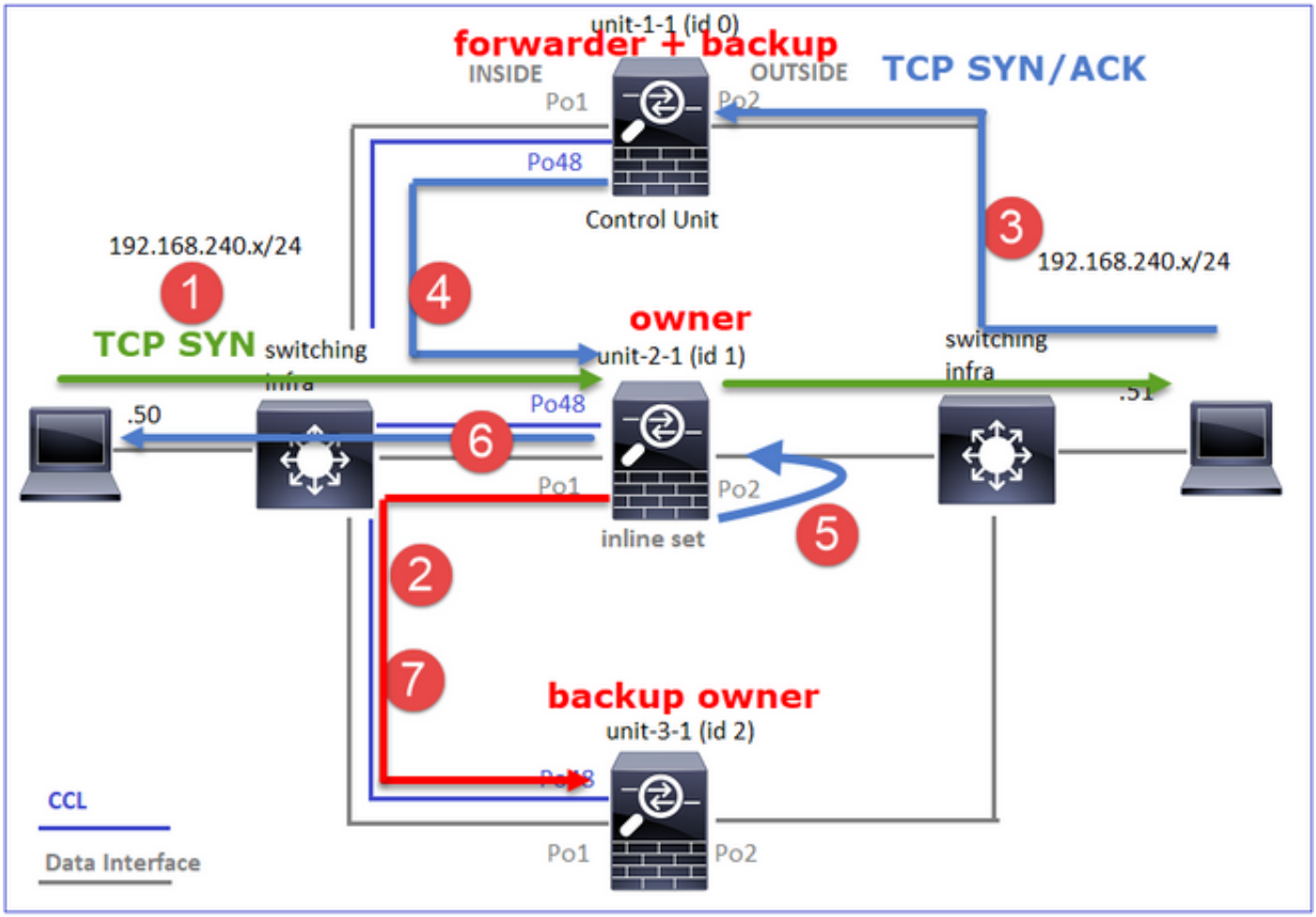
unit-3-1

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

ユニット	フラグ	注
ユニット1-1	z	· フォワーダ
ユニット2-1	b N	· フロー所有者 – ユニットがフローを処理します
ユニット3-1	y	· バックアップ· オーナー

それを図で示します。



1. TCP SYNパケットがホストAからユニット2-1に到着します。ユニット2-1がフローのオーナーになり、ディレクタに選出されます。
2. ユニット3-1がバックアップ所有者として選出されます。フローオーナーは「cluster add」ユニキャストメッセージをUDP 4193で送信し、バックアップのオーナーにフローについて通知します。
3. TCP SYN/ACKパケットがホストBからユニット1-1に到着します。フローは非対称です。
4. ユニット1-1は、パケットをCCL経由でディレクタ（ユニット2-1）に転送します。
5. Unit-2-1も所有者であり、インターフェイスOUTSIDEでパケットを再注入します。
6. ユニット2-1はパケットをホストAに転送します。
7. 接続が終了すると、所有者はクラスタ削除メッセージを送信して、バックアップ所有者からフロー情報を削除します。

観察3.トレース付きのキャプチャは、非対称トラフィックと、ユニット1-1からユニット2-1へのリダイレクトを示しています。

ユニット2-1（オーナー/ディレクタ）

<#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
```

ユニット1-1（フォワーダ）

```
<#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 w
Phase: 1
Type: CLUSTER-EVENT
Subtype:
Result: ALLOW
Config:
Additional Information:
Input interface: 'OUTSIDE'
Flow type: NO FLOW


I (0) am asking director (1).
```

リターントラフィック(TCP SYN/ACK)

ユニット2-1（オーナー/ディレクタ）

<#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 w
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

観察4.FTDデータプレーンのsyslogには、すべてのユニットでの接続の作成と終了が表示されます。

- ユニット1-1（所有者）
- ユニット2-1（フォワーダ）
- ユニット3-1（バックアップ・オーナー/ダイレクタ）

<#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 (192.168.240.50/5184
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
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

ケース スタディ 7非対称トラフィック（インライン・セット、所有者がダイレクタと異なる）

所有者はユニット2-1です（reinject-hideキャプチャのINSIDEインターフェイスとOUTSIDEインターフェイスの両方にパケットがありますが、ユニット3-1はOUTSIDEのみにパケットがあります）。

<#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
```

観察2.送信元ポート59210を使用したフローの接続フラグ分析

<#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:

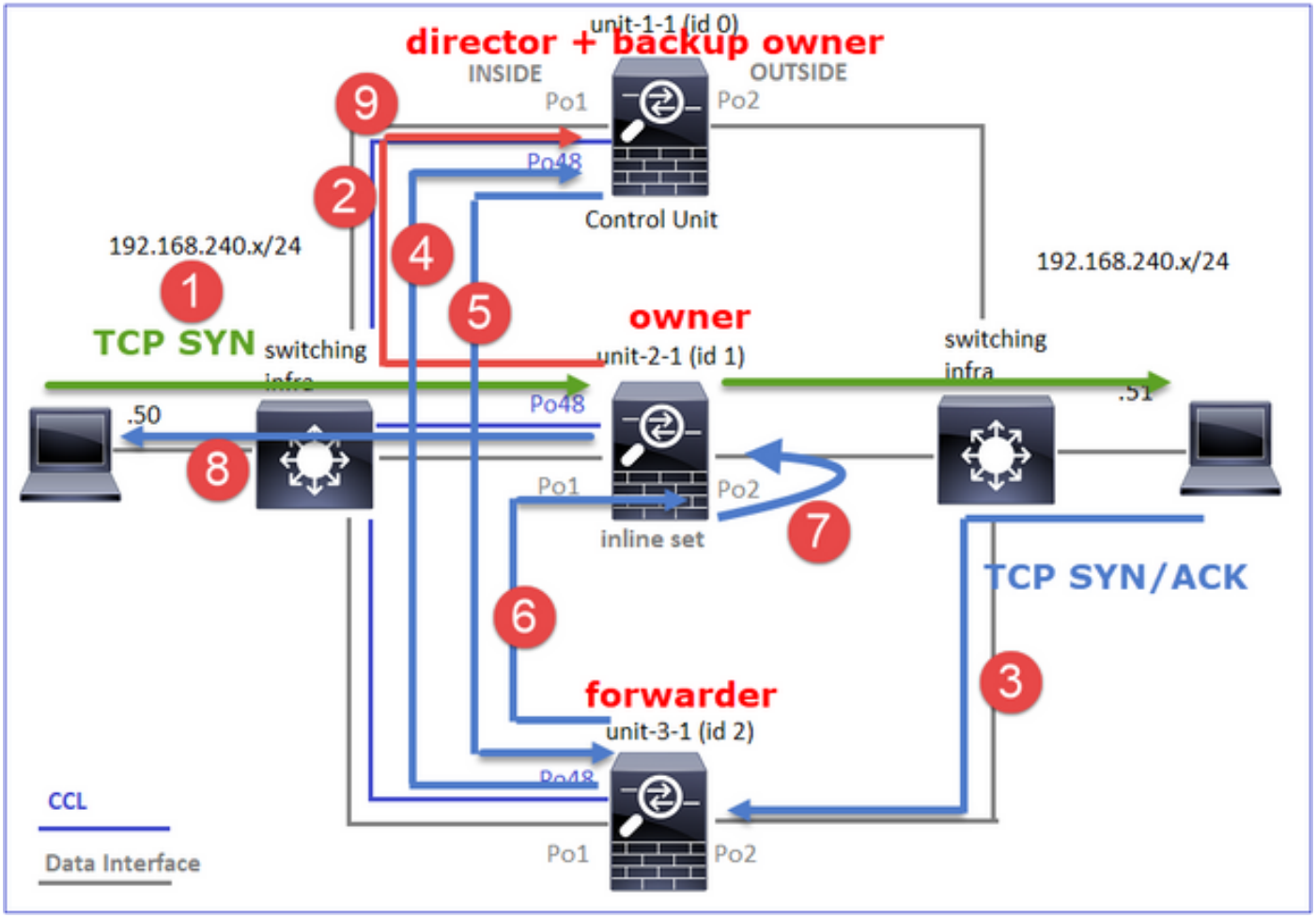
59210

, idle 0:00:00, bytes 0,

flags z

ユニット	フラグ	注
ユニット1-1	Y	· ダイレクタ/バックアップ· オーナー
ユニット2-1	b N	· フロー所有者 – ユニットがフローを処理します
ユニット3-1	z	· フォワーダ

それを図で示します。



1. TCP SYNパケットがホストAからユニット2-1に到着します。ユニット–2-1がフローのオーナーになり、ユニット–1-1がディレクタとして選出されます
2. ユニット1-1はディレクタであるため、バックアップ所有者として選出されます。 フローオーナーは「cluster add」ユニキャストメッセージをUDP 4193に送信します。バックアップのオーナーにフローを通知します。
3. TCP SYN/ACKパケットがホストBからユニット3-1に到着します。フローは非対称です。
4. ユニット3-1は、パケットをCCL経由でディレクタ（ユニット1-1）に転送します。
5. Unit-1-1（ダイレクタ）は、所有者がUnit-2-1であることを認識し、パケットをフォワーダ(Unit-3-1)に返送し、所有者がUnit-2-1であることを通知します。

6. ユニット3-1は、ユニット2-1（所有者）にパケットを送信します。
7. ユニット2-1は、インターフェイスOUTSIDEでパケットを再注入します。
8. ユニット2-1はパケットをホストAに転送します。
9. 接続が終了すると、所有者はクラスタ削除メッセージを送信して、バックアップ所有者からフロー情報を削除します。

注：ステップ2（CCL経由のパケット）がステップ4（データトラフィック）の前に行われることが重要です。 別のケース（たとえば、競合状態）では、ディレクタはフローを認識しません。したがって、インラインセットであるため、パケットを宛先に転送します。インターフェイスがインラインセットにない場合、データパケットはドロップされます。

観察3.トレースを使用したキャプチャは、CCL上の非対称トラフィックと交換を示します。

転送トラフィック(TCP SYN)

ユニット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 <mss
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
```

リターントラフィック(TCP SYN/ACK)

Unit-3-1（ID 2 – フォワーダ）は、パケットをCCL経由でUnit-1-1（ID 0 – ディレクタ）に送信します。

<#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（ダイレクタ）:Unit-1-1(ID 0)では、フロー所有者がUnit-2-1(ID 1)であることを認識し、CCLを介してパケットをUnit-3-1（ID 2 – フォワーダ）に返信します。

<#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 – フォワーダ）は、CCLを介してパケットを取得し、ユニット–2-1（ID 1 – 所有者）に送信します。

<#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).

所有者はパケットを再インジェクトし、宛先に転送します。

<#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).
```

観察4.FTDデータプレーンのsyslogには、すべてのユニットでの接続の作成と終了が表示されます。

- ユニット1-1（ダイレクタ/バックアップ・オーナー）
- ユニット2-1（所有者）
- ユニット3-1（フォワーダ）

```
<#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
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 T

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 (192.168.240.50/5921
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
```

トラブルシュート

クラスタトラブルシューティングの概要

クラスタの問題は、次のように分類できます。

- コントロールプレーンの問題（クラスタの安定性に関連する問題）
- データプレーンの問題（中継トラフィックに関連する問題）

クラスタデータプレーンの問題

NAT/PATの一般的な問題

設定に関する重要な考慮事項

- Port Address Translation（PAT；ポートアドレス変換）プールには、少なくともクラスタ内のユニット数と同じ数のIPが必要です。できれば、クラスタノード数よりも多くのIPが必要です。
- デフォルトのxlate per-sessionコマンドは、無効にする特別な理由がない限り、そのままにしておく必要があります。接続のために構築されたPAT xlateでセッションごとのxlateが無効になっているものは、常にクラスタ内のコントロールノードユニットで処理されるため、パフォーマンスの低下を引き起こす可能性があります。

低いポートから送信されたトラフィックによってPATプール範囲の使用率が高くなり、クラスタIPの不均衡が発生する

FTDはPAT IPを複数の範囲に分割し、xlateを同じ送信元範囲に維持しようとします。次の表は、送信元ポートが同じ送信元範囲内のグローバルポートにどのように変換されるかを示しています。

元の送信元ポート	変換済み送信元ポート
1-511	1-511
512-1023	512-1023
1024-65535	1024-65535

送信元ポート範囲がいっぱいになり、その範囲から新しいPAT xlateを割り当てる必要がある場合、FTDは次のIPに移動して、その送信元ポート範囲に新しい変換を割り当てます。

症状

クラスタを通過するNATトラフィックの接続の問題

検証

<#root>

#

show nat pool

FTDデータプレーンログにPATプールの枯渇が示されます。

<#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

緩和

NATフラットポート範囲を設定し、予約ポートを含める。

さらに、6.7/9.15.1以降では、PATの対象となる大量のバックグラウンドトラフィックによってノードがクラスタを出入りしたときにだけ、ポートブロック分散が不均衡になることがあります。単独で回復する唯一の方法は、ポートブロックがノード間で再分散されるように解放される場合です。

ポートブロックベースの分散では、たとえばpb-1、pb-2 ... pb-10などの10個のポートブロックを使用してノードが割り当てられます。ノードは常に最初の使用可能なポートブロックから開始し、空きポートがなくなるまでランダムポートを割り当てます。割り当ては、そのポイントまでのすべてのポートブロックが使い果たされた場合にのみ、次のポートブロックに移動します。

たとえば、ホストが512の接続を確立すると、ユニットはpb-1からの512の接続すべてに対してマップされたポートをランダムに割り当てます。ここで、これらの512接続がすべてアクティブな状態で、pb-1が使い果たされて以降ホストが513番目の接続を確立すると、ホストはpb-2に移動し、そこからランダムなポートを割り当てます。ここで再び、513の接続のうち、10番目の接続が終了し、pb-1で使用可能な1つのポートがクリアされたと仮定します。この時点で、ホストが514番目の接続を確立すると、pb-1には空きポート（10番目の接続の削除の一部として解放された）が存在するため、クラスタユニットはpb-2ではなくpb-1からマッピングされたポートを割り当てます。

注意すべき重要な点は、割り当ては空きポートを持つ最初の使用可能なポートブロックから行われ、それにより最後のポートブロックは通常ロードされるシステムで常に再配送に使用できるということです。また、PATは通常、短期間の接続に使用されます。ポートブロックがより短い時間で使用可能になる可能性は非常に高くなります。したがって、プール分散のバランスをとるために必要な時間は、ポートブロックベースのプール分散によって改善できます。

ただし、pb-1からpb-10までのすべてのポートブロックが使い果たされた場合、または各ポートブロックが長期間接続するためのポートを保持している場合は、ポートブロックが迅速に解放されて再配送されることはありません。このような場合、最も中断の少ないアプローチは次の点です。

1. 過剰なポートブロックがあるノードを特定します(show nat pool cluster summary)。
2. そのノードで最も使用率の低いポートブロックを特定します(show nat pool ip <addr> detail)。
3. このようなポートブロックに対してclear xlate (clear xlate global <addr> gport 'start-end')を実行し、再配送に使用できるようにします。

⚠ 警告：これにより、関連する接続が中断されます。

異なる宛先へのリダイレクトが発生すると、デュアルチャネルWebサイト（Webメール、銀行など）やSSO Webサイトを参照できません。

症状

デュアルチャネルWebサイト（Webメール、銀行のWebサイトなど）を参照できません。 ユーザが、クライアントに2つ目のソケット/接続を開くことを要求するWebサイトに接続し、2つ目の接続が、1つ目の接続のハッシュを受け取ったクラスタメンバとは異なるクラスタメンバにハッシュされ、トラフィックがIP PATプールを使用する場合、別のパブリックIPアドレスから接続を受け取ると、トラフィックはサーバによってリセットされます。

検証

データプレーンクラスタのキャプチャを取得し、影響を受ける中継フローの処理方法を確認します。この場合、宛先WebサイトからTCPリセットが送信されます。

緩和策（6.7より前、9.15.1より前）

- マルチセッションアプリケーションが複数のマッピングされたIPアドレスを使用するかどうかを確認します。
- プールが均等に分散されているかどうかを確認するには、show nat pool cluster summaryコマンドを使用します。
- cluster exec show connコマンドを使用して、トラフィックのロードバランスが適切に行われているかどうかを確認します。
- スティッキIPのプール使用量を確認するには、show nat pool cluster ip <address> detailコマンドを使用します。
- syslog 305021(6.7/9.15)を有効にして、スティッキIPの使用に失敗した接続を確認します。
- 解決するには、PATプールにさらにIPを追加するか、接続されたスイッチでロードバランス

アルゴリズムを微調整します。

EtherChannelロードバランシングアルゴリズムについて：

- 非FP9300の場合、および1つのサーバ経由で認証が発生する場合：隣接するスイッチで、送信元IP/ポートと宛先IP/ポートから送信元IPと宛先IPに向かって、EtherChannelロードバランシングアルゴリズムを調整します。
- 非FP9300の場合、および認証が複数のサーバを介して行われる場合：送信元IP/ポートと宛先IP/ポートから送信元IPへの隣接スイッチで、EtherChannelロードバランシングアルゴリズムを調整します。
- FP9300の場合：FP9300シャーシでは、ロードバランシングアルゴリズムはsource-dest-port source-dest-ip source-dest-macとして固定されており、変更できません。この場合の回避策は、FlexConfigを使用してxlate per-session denyコマンドをFTD設定に追加し、（問題のある、または互換性のないアプリケーションの）特定の宛先IPアドレスへのトラフィックが、シャーシクラスタ内の制御ノードによってのみ処理されるようにします。回避策には、次の副作用が伴います。
  - 異なる変換されたトラフィックのロードバランシングは行われない（すべてのトラフィックがコントロールノードに送られる）。
  - xlateスロットが枯渇する可能性（およびコントロールノード上の他のトラフィックのNAT変換に悪影響を及ぼす）。
  - シャーシクラスタ内のスケーラビリティの低下。

プール内のPAT IPが不十分なため、コントロールノードに送信されるすべてのトラフィックによるクラスタパフォーマンスの低下。

症状

クラスタには、データノードに空きIPを割り当てるのに十分なPAT IPがないため、PAT設定の対象となるすべてのトラフィックは、処理のためにコントロールノードに転送されます。

検証

show nat pool clusterコマンドを使用して各ユニットの割り当てを表示し、すべてのユニットがプール内で少なくとも1つのIPを所有していることを確認します。

緩和

6.7/9.15.1よりも前の場合は、クラスタ内のノードの数と少なくとも同じサイズのPATプールがあることを確認します。PATプールを使用する6.7/9.15.1以降では、すべてのPATプールIPからポートブロックを割り当てます。PATプールの使用率が非常に高く、プールが頻繁に使い果たされる場合は、PATプールサイズを増やす必要があります（「FAQ」のセクションを参照）。

xlateがセッション単位で有効になっていないため、コントロールノードに送信されるすべてのトラフィックによるパフォーマンスが低下します。

症状

多数の高速UDPバックアップフローがクラスタ制御ノードで処理されるため、パフォーマンスに影響を与える可能性があります。

背景

PATを使用するデータノードで処理できるのは、セッション単位で有効なxlateを使用する接続だけです。show run all xlateコマンドを使用して、xlateのセッション単位の設定を表示します。

Per-session enabledは、関連付けられている接続が解除されるとすぐにxlateが解除されることを意味します。これにより、接続にPATが適用される場合の1秒あたりの接続パフォーマンスが向上します。非セッションごとのxlateは、関連する接続が解除されてからさらに30秒間有効です。接続レートが十分に高い場合、各グローバルIP上の使用可能な65k TCP/UDPポートを短時間で使い切ることができます。

デフォルトでは、すべてのTCPトラフィックがxlate単位で有効になっており、UDP DNSトラフィックのみがセッション単位で有効になっています。これは、すべての非DNS UDPトラフィックが処理のためにコントロールノードに転送されることを意味します。

検証

クラスタユニット間の接続とパケット分散を確認するには、次のコマンドを使用します。

<#root>

```
firepower#
show cluster info conn-distribution

firepower#
show cluster info packet-distribution

firepower#
show cluster info load-monitor
```

cluster exec show connコマンドを使用して、どのクラスタノードがUDP接続を所有しているかを確認します。

<#root>

```
firepower#
cluster exec show conn
```

このコマンドを使用して、クラスタノード間のプール使用率を把握します。

<#root>

```
firepower#
```

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

```
      | in UDP
```

緩和

対象のトラフィック（UDPなど）のセッション単位のPAT(per-session permit udpコマンド)を設定します。 ICMPの場合、デフォルトのマルチセッションPATからは変更できないため、PATが設定されている場合、ICMPトラフィックは常にコントロールノードによって処理されます。

ノードがクラスタを離れたりクラスタに参加したりすると、PATプールの分散が不均衡になる

症状

- PATのIP割り当ては、時間の経過とともにクラスタを出入りするユニットによって不均衡になる可能性があるため、接続の問題が発生します。
- 6.7/9.15.1以降では、新しく参加したノードが十分なポートブロックを取得できない場合があります。ポートブロックを持たないノードは、トラフィックをコントロールノードにリダイレクトします。少なくとも1つのポートブロックを持つノードがトラフィックを処理し、プールが使い果たされるとドロップします。

検証

- データプレーンのsyslogには、次のようなメッセージが表示されます。

<#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
```

- プールの分散を確認するには、show nat pool cluster summaryコマンドを使用します。
- cluster exec show nat pool ip <addr> detailコマンドを使用して、クラスタノード間のプール使用率を確認します。

緩和

- 6.7/9.15.1よりも前のリリースでは、いくつかの回避策がCisco Bug ID CSCvd10530
.
- 6.7/9.15.1以降では、clear xlate global <ip> gport <start-end>コマンドを使用して、必要なノードに再配布するために他のノードのポートブロックの一部を手動でクリアします。

症状

クラスタによってPATされるトラフィックの主要な接続の問題。これは、FTDデータプレーンでは、設計上、グローバルNATアドレスに対してGARPが送信されないためです。

検証

直接接続されたデバイスのARPテーブルは、制御ノードの変更後に、クラスタデータインターフェイスのMACアドレスが異なることを示しています。

<#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
```

緩和

クラスタデータインターフェイスでスタティック（仮想）MACを設定します。

PAT障害の影響を受ける接続

症状

クラスタによってPATされるトラフィックの接続の問題。

検証/緩和

- 設定が正しく複製されていることを確認します。
- プールが均等に分散されていることを確認します。

- プールの所有権が有効であることを確認します。
- show asp clusterカウンタで障害カウンタが増加しません。
- ディレクタ/フォワーダフローが適切な情報で作成されていることを確認します。
- バックアップxlateが正常に作成、更新、およびクリーンアップされているかどうかを検証します。
- xlateが作成され、「セッション単位」の動作に従って終了するかどうかを検証します。
- 「debug nat 2」を有効にすると、エラーが表示されます。この出力は非常にノイズが多い可能性があることに注意してください。次に例を示します。

```
<#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
```

デバッグを停止するには、次の手順を実行します。

```
<#root>

firepower#

un all
```

- 接続およびNAT関連のsyslogを有効にして、情報を失敗した接続に関連付けます。

ASAおよびFTDクラスタリングのPATの改善（9.15および6.7以降）

何が変更されたのですか。

PATの動作が再設計されました。個々のIPアドレスが各クラスタメンバーに配布されることはありません。代わりに、PAT IPはポートブロックに分割され、IPスティッキ性動作と組み合わせて、これらのポートブロックをクラスタメンバー間で均等に（可能な限り）分散します。

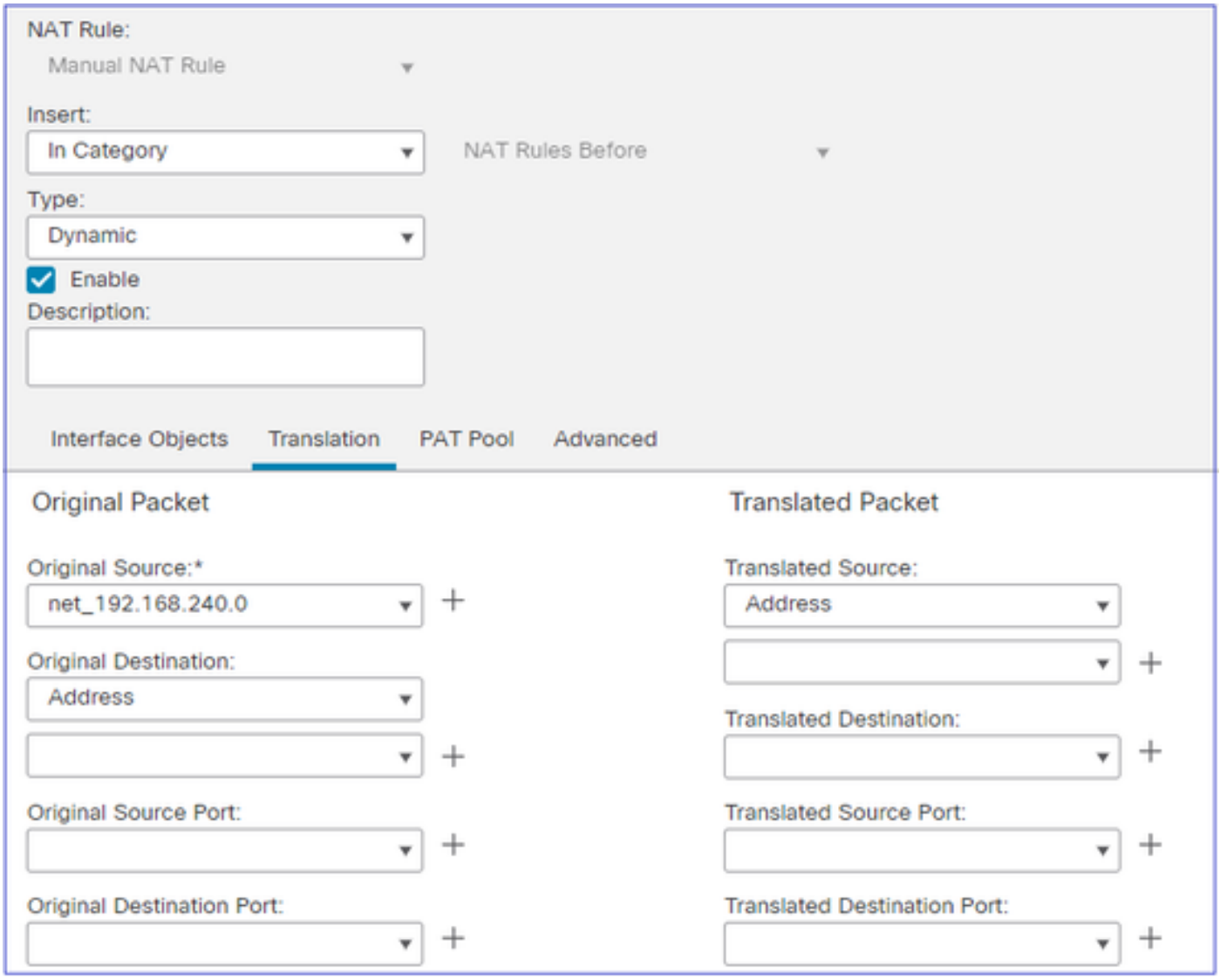
新しい設計では、次の制限事項に対処します（前のセクションを参照）。

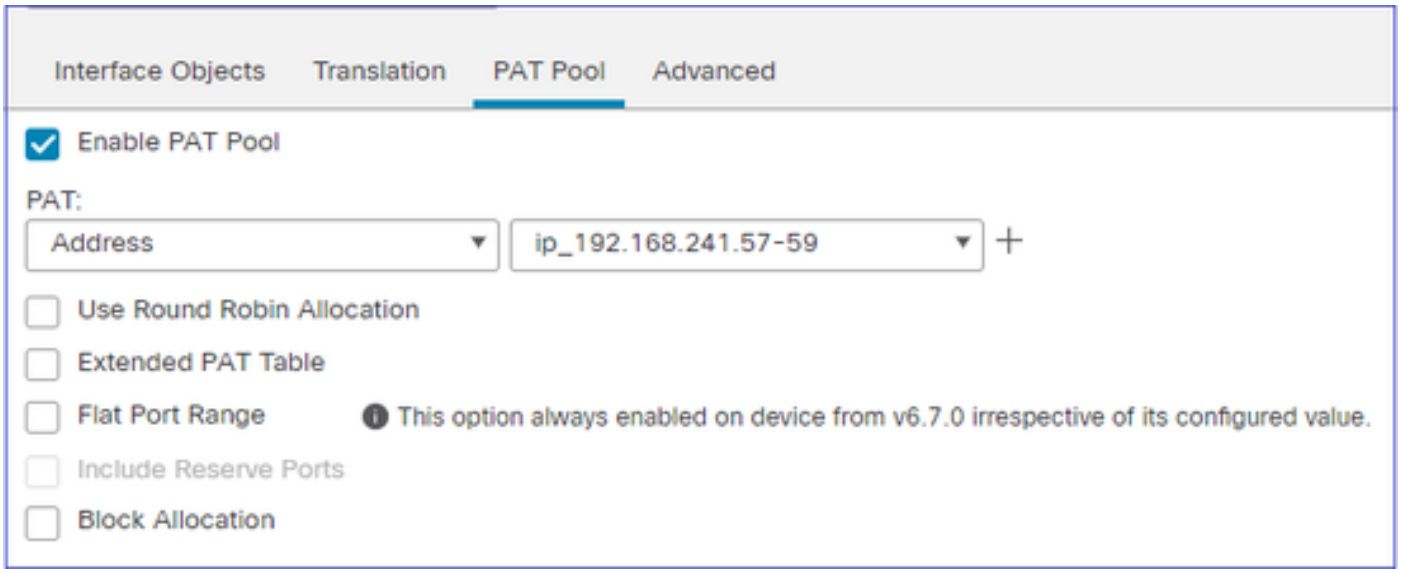
- マルチセッションアプリケーションは、クラスタ全体のIPスティッキ性の欠如による影響を受けます。
- この要件は、クラスタ内のノードの数と少なくとも同じサイズのPATプールを持つことです

。
- ノードがクラスタを離れたりクラスタに参加したりすると、PATプールの分散が不均衡になる
- PATプールの不均衡を示すsyslogはありません。

技術的には、デフォルトの1 ~ 511、512 ~ 1023、および1024 ~ 65535のポート範囲の代わりに、PATのデフォルトポート範囲として1024 ~ 65535が存在します。このデフォルト範囲は、通常のPAT用に1 ~ 1023の特権ポート範囲を含むように拡張できます（「include-reserve」オプション）。

次に、FTD 6.7でのPATプールの設定例を示します。詳細については、『設定ガイド』の次の関連セクションを参照してください。





PATに関するその他のトラブルシューティング情報

FTDデータプレーンのsyslog（6.7/9.15.1以降）

スティッキ性無効化syslogは、クラスタノード上のスティッキIPですべてのポートが使い果たされ、割り当てが空きポートを持つ次に使用可能なIPに移動すると生成されます。

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

Pool inbalance syslogは、ノードがクラスタに参加するときに生成され、ポートブロックの任意または不均等な共有を取得しません。次に例を示します。

```
%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 hav
```

Show コマンド

プールの配布ステータス

show nat pool cluster summaryの出力では、各PAT IPアドレスについて、平衡型分散シナリオのノード間で1ポートブロックを超える差を設けないようにする必要があります。平衡型と不平衡型のポートブロック分配の例。

<#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)
```

不均衡な分散：

<#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)
```

プール所有権の状態

show nat pool clusterの出力では、UNKNOWNとして所有者またはバックアップのいずれかが指定された単一のポートブロックは存在できません。存在する場合は、プール所有権通信に問題があることを示します。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>
```

ポートブロック内のポート割り当てのアカウンティング

show nat poolコマンドは、フィルタリングされた出力だけでなく詳細情報を表示するオプションが追加されて拡張されています。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 *
```

'*'は、バックアップされたポートブロックであることを示します。

この問題を解決するには、clear xlate global <ip> gport <start-end>コマンドを使用して、他のノードにあるポートブロックの一部を手動でクリアし、必要なノードに再配布します。

手動でトリガーされるポートブロックの再配布

- トラフィックが絶え間ない実稼働ネットワークでは、ノードがクラスタを離れて再参加する際（おそらくトレースバックが原因）、プールの均等な共有を取得できない場合や、最悪の場合はポートブロックを取得できない場合があります。
- show nat pool cluster summaryコマンドを使用して、必要以上のポートブロックを所有しているノードを特定します。
- より多くのポートブロックを所有するノードで、show nat pool ip <addr> detailコマンドを使用し、割り当て数が最も少ないポートブロックを特定します。
- clear xlate global <address> gport <start-end>コマンドを使用し、これらのポートブロックから作成された変換をクリアして、必要なノードに再配送できるようにします。次に例を示します。

<#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以降のPATに関するFAQ

Q.クラスタ内の使用可能なユニット数に対して使用可能なIPの数がある場合、オプションとしてユニットあたり1つのIPを使用できますか。

A.もはや使用されておらず、IPアドレスベースとポートブロックベースのプール配布方式を切り替える機能はありません。

以前のIPアドレスベースのプール分散方式では、ホストからの複数の接続（単一のアプリケーションのトランザクションの一部）がクラスタの異なるノードにロードバランスされ、マッピングされた異なるIPアドレスに変換されて宛先サーバが異なるエンティティから送信されたものと認識されてしまうマルチセッションアプリケーション障害が発生していました。

また、新しいポートブロックベースの分散方式を使用すると、1つのPAT IPアドレスで十分な数のPAT IPアドレスを処理できますが、PATが必要な接続数に基づいて十分な数のPAT IPアドレスを使用することが常に推奨されます。

Q.クラスタのPATプール用のIPアドレスのプールを引き続き使用できますか。

A.はい、できます。すべてのPATプールIPからのポートブロックがクラスタノード間で分散されます。

Q. PATプールで多数のIPアドレスを使用する場合、各メンバに割り当てられるポートのブロックは各IPアドレスごとに同じですか。

A.いいえ。各IPは個別に分散されています。

Q.すべてのクラスタノードにすべてのパブリックIPがありますが、ポートのサブセットだけですか。この場合、送信元IPが同じパブリックIPを使用するたびに、それが保証されますか。

A.正解です。各PAT IPの一部が各ノードによって所有されています。選択したパブリックIPアドレスがノードで使い果たされると、スティッキIPを保持できないことを示すsyslogが生成され、割り当ては次に使用可能なパブリックIPに移動します。スタンドアロン、HA、またはクラスタ展開であっても、IPスティッキ性は常にベストエフォートベースであり、プールの可用性によって異なります。

Q.すべてがPATプール内の1つのIPアドレスに基づいていますが、PATプール内で複数のIPアドレスが使用されている場合は適用されません。

A. PATプール内の複数のIPアドレスにも適用されます。PATプール内のすべてのIPからのポートブロックは、クラスタノード間で分散されます。PATプールのすべてのIPアドレスは、クラスタ内のすべてのメンバに分割されます。したがって、PATプール内にクラスCのアドレスがある場合、すべてのクラスタメンバは、すべてのPATプールアドレスからのポートプールを持ちます。

Q. CGNATで動作しますか。

A.はい。CGNATも同様にサポートされています。ブロック割り当てPATとも呼ばれるCGNATのデフォルトのブロックサイズは「512」で、xlate block-allocation size CLIで変更できます。通常のダイナミックPAT（非CGNAT）の場合、ブロックサイズは常に「512」で、固定で設定できません。

Q.装置がクラスタから離脱する場合、制御ノードはポートブロック範囲を他の装置に割り当てますか、それとも制御ノード自体に残しますか。

A.各ポートブロックには所有者とバックアップがあります。xlateは、ポートブロックから作成されるたびに、ポートブロックバックアップノードにも複製されます。ノードがクラスタを離れると、バックアップノードはすべてのポートブロックと現在の接続を所有します。バックアップノードは、これらの追加のポートブロックの所有者になったため、新しいバックアップを選択し、障害のシナリオを処理するために現在のすべてのxlateをそのノードに複製します。

Q.そのアラートに基づいて、スティッキ性を強化するためにどのようなアクションを実行できますか。

A.粘着性を保持できない理由は2つあります。

理由–1：トラフィックのロードバランシングが正しく行われない。これは、いずれかのノードで他のノードよりも多くの接続数が検出されることが原因で、特定のスティッキIP枯渇が発生するためです。トラフィックがクラスタノード間で均等に分散されるようにすれば、この問題に対処できます。たとえば、FPR41xxクラスタでは、接続されたスイッチでロードバランシングアルゴリズムを調整します。FPR9300クラスタでは、シャーシ全体でブレードの数が同じになるようにします。

理由2:PATプールの使用率が非常に高いため、プールが頻繁に使い果たされます。これに対処するには、PATプールサイズを増やします。

Q. extendedキーワードのサポートはどのように処理されるのですか。エラーが表示され、アップグレード中にNATコマンド全体が追加されるのが阻止されますか。それとも、拡張キーワードが削除されて、警告が表示されますか。

A. PAT拡張オプションは、ASA 9.15.1/FP 6.7以降のクラスタではサポートされていません。設定オプションは、CLI/ASDM/CSM/FMCからは削除されません。（アップグレードによって直接または間接的に）設定すると、警告メッセージが表示されて設定が受け入れられますが、実際にはPATの拡張機能は表示されません。

Q.同時接続と同じ数の変換ですか。

A. 6.7/9.15.1よりも前のリリースでは、1 ~ 65535でしたが、送信元ポートは1 ~ 1024の範囲ではあまり使用されないため、事実上1024 ~ 65535になります(64512 conns)。 デフォルトの動作

として「flat」を使用した6.7/9.15.1以降の実装では、これは1024-65535です。ただし、1-1024を使用する場合は、「include-reserve」オプションを使用できます。

Q.ノードがクラスタに戻ると、古いバックアップノードがバックアップとして存在し、そのバックアップノードが古いポートブロックをノードに提供します。

A.その時点でのポートブロックのアベイラビリティによって異なります。ノードがクラスタを離れると、そのノードのすべてのポートブロックがバックアップノードに移動します。次に、空きポートブロックを蓄積して必要なノードに配布するのは、制御ノードです。

Q.制御ノードの状態に変更がある場合、新しい制御ノードが選出され、PATブロックの割り当てが維持されるのですか。あるいは、ポートブロックが新しい制御ノードに基づいて再割り当てされるのですか。

A.新しい制御ノードは、どのブロックが割り当てられており、どれが空いているかがわかっています。

Q. xlateの最大数は、この新しい動作での同時接続の最大数と同じですか。

A. あります。xlateの最大数は、PATポートのアベイラビリティによって異なります。これは、同時接続の最大数とは無関係です。1つのアドレスだけを許可する場合は、接続65535可能です。より多くのIPアドレスが必要な場合は、割り当てるIPアドレスを増やす必要があります。十分なアドレス/ポートがあれば、最大同時接続数に到達できます。

Q.新しいクラスタメンバが追加されるときのポートブロック割り当てのプロセスはどのようなものですか。 リブートによってクラスタメンバーが追加された場合はどうなりますか。

A.ポートブロックは常にコントロールノードによって分散されます。ポートブロックが新しいノードに割り当てられるのは、空きポートブロックがある場合だけです。空きポートブロックは、ポートブロック内のマッピングされたポートを介して接続が提供されないことを意味します。

さらに、再結合時に、各ノードは所有できるブロックの数を再計算します。ノードが想定よりも多くのブロックを保持している場合、追加のポートブロックが使用可能になると、その追加のポートブロックがコントロールノードにリリースされます。次に、制御ノードはそれらを新しく結合されたデータノードに割り当てます。

Q. TCPおよびUDPプロトコルまたはSCTPのみがサポートされていますか。

A. SCTPはダイナミックPATではサポートされていません。SCTPトラフィックでは、スタティックネットワークオブジェクトNATのみを使用することを推奨します。

Q.ノードでブロックポートが使い果たされている場合、パケットはドロップされ、次に使用可能なIPブロックは使用されないのですか。

A.いいえ。すぐには廃棄されません。次のPAT IPから使用可能なポートブロックを使用するすべてのPAT IPのすべてのポートブロックが使い果たされると、トラフィックは廃棄されます。

Q.クラスタアップグレードの時間帯に制御ノードの過負荷を避けるために、制御ノードですべての接続が処理されるのを待つよりも、早い段階で (たとえば、4ユニットのクラスタアップグレー

ドの途中で）新しい制御を手動で選択する方が良いですか。

A.コントロールは最後に更新する必要があります。これは、制御ノードが新しいバージョンを実行する場合、すべてのノードが新しいバージョンを実行しない限り、プールの配布を開始しないためです。また、アップグレードを実行すると、新しいバージョンを持つすべてのデータノードは、古いバージョンを実行している場合にコントロールノードからのプール配布メッセージを無視します。

これを詳細に説明するために、A、B、C、およびDの4つのノードを制御とするクラスタ展開を検討します。一般的な無中断アップグレード手順を次に示します。

1. 各ノードに新しいバージョンをダウンロードします。
2. ユニット'D'をリロードします。すべての接続、xlateはバックアップノードに移動します。
3. ユニット「D」が起動し、以下が表示されます。

a. PATの設定を処理します

b.各PAT IPをポートブロックに分割する

c.すべてのポートブロックが未割り当て状態である

d.コントロールから受信した古いバージョンのクラスタPATメッセージを無視します。

e.すべてのPAT接続をプライマリにリダイレクトします。

4. 同様に、新しいバージョンで他のノードを起動します。

5. ユニット'A'コントロールをリロードします。制御用のバックアップがないため、既存の接続はすべてドロップされます

6. 新しいコントロールは、新しい形式でポートブロックの配布を開始します

7. ユニット「A」が再結合し、ポートブロックの配信メッセージを受け入れて処理できる

フラグメント処理

症状

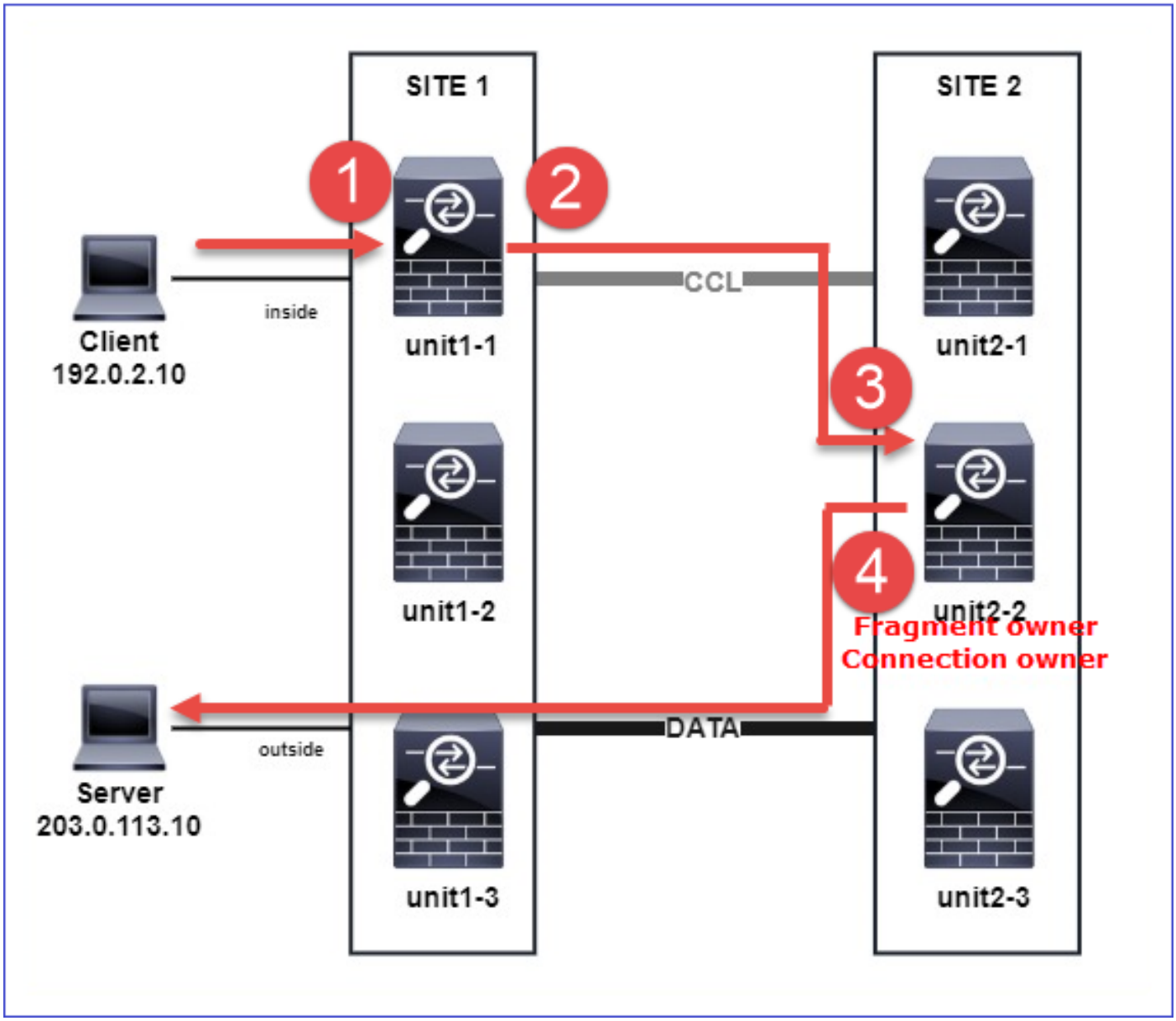
サイト間クラスタの導入では、1つの特定のサイトで処理する必要があるフラグメント化されたパケット（サイトローカルトラフィック）は、他のサイトのユニットに送信できます。これは、これらのサイトの1つがフラグメント所有者を持つことができるためです。

クラスタ論理では、フラグメント化されたパケットを持つ接続に対して、フラグメント所有者というロールが追加で定義されています。

フラグメント化されたパケットの場合、フラグメントを受信するクラスタユニットは、フラグメントの発信元IPアドレス、宛先IPアドレス、およびパケットIDのハッシュに基づいてフラグメント所有者を決定します。その後、すべてのフラグメントは、クラスタ制御リンクを介してフラグメント所有者に転送されます。スイッチのロードバランシングハッシュで使用される5タプルを含んでいるのは最初のフラグメントだけなので、フラグメントは異なるクラスタユニットにロード

バランスできます。その他のフラグメントには送信元ポートと宛先ポートが含まれず、他のクラスタユニットにロードバランスできます。フラグメントの所有者は、送信元/宛先IPアドレスとポートのハッシュに基づいてディレクタを決定できるように、パケットを一時的に再構成します。新しい接続の場合は、フラグメントの所有者が接続の所有者になります。既存の接続の場合、フラグメント所有者はすべてのフラグメントをクラスタ制御リンク経由で接続所有者に転送します。その後、接続の所有者がすべてのフラグメントを再構成します。

クライアントからサーバへのフラグメント化ICMPエコー要求のフローについて、次のトポロジを検討します。



操作の順序を理解するために、トレースオプションを使用して設定された内部、外部、およびクラスタ制御リンクインターフェイスで、クラスタ全体のパケットキャプチャがあります。 さらに、reinject-hideオプションを使用したパケットキャプチャが内部インターフェイスで設定されています。

<#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
```

クラスタ内の操作順序：

1. サイト1のユニット1-1は、フラグメント化されたICMPエコー要求パケットを受信します。

```
<#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は、サイト2のユニット–2-2をフラグメント所有者として選択し、フラグメント化されたパケットを送信します。
ユニット1-1からユニット2-2に送信されるパケットの宛先MACアドレスは、ユニット2-2のCCLリンクのMACアドレスです。

```
<#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. ユニット2-2は、フラグメント化されたパケットを受信して再構成し、フローのオーナーになります。

<#root>

```
firepower#

cluster exec unit unit-2-2 show capture capccl 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. ユニット2-2は、セキュリティポリシーに基づいてパケットを許可し、外部インターフェイスを介してサイト2からサイト1にパケットを送信する。

<#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

観察/警告

- ディレクタの役割とは異なり、フラグメントの所有者を特定のサイト内にローカライズすることはできません。フラグメントの所有者は、新しい接続のフラグメント化されたパケットを最初に受信したユニットによって決定され、任意のサイトに配置できます。
- フラグメントの所有者が接続の所有者になることもできるので、パケットを宛先ホストに転送するには、出力インターフェイスを解決して、宛先ホストまたはネクストホップのIPアドレスとMACアドレスを検出できる必要があります。これは、ネクストホップも宛先ホストへの到達可能性を持っている必要があることを前提としています。
- フラグメント化されたパケットを再構成するために、ASA/FTDは名前付きインターフェイスごとにIPフラグメント再構成モジュールを維持します。IPフラグメント再構成モジュールの動作データを表示するには、show fragmentコマンドを使用します。
  <#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
  ```

  クラスタ展開では、フラグメント所有者または接続所有者が、フラグメント化されたパケットをフラグメントキューに入れます。フラグメントキューのサイズは、fragment size <size> <nameif>コマンドで設定されたSizeカウンタの値（デフォルトは200）によって制限されます。フラグメントキューのサイズがSizeの2/3に達すると、フラグメントキューのしきい値を超えたと見なされ、現在のフラグメントチェーンに含まれない新しいフラグメントはすべて廃棄されます。この場合、Fragment queue threshold exceededが増加し、syslogメッセージFTD-3-209006が生成されます。

<#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 203.0.11

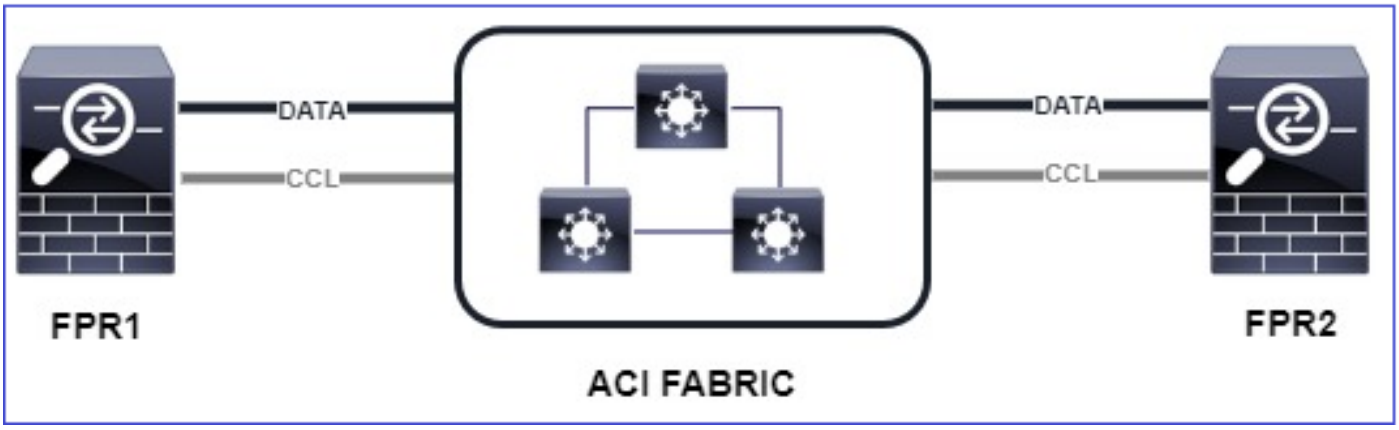
回避策として、Firepower Management Center > Devices > Device Management > [Edit Device] > Interfaces > [Interface] > Advanced > Security Configuration > Override Default Fragment Settingでサイズを増やし、設定を保存してポリシーを展開します。次に、show fragmentコマンド出力のキューカウンタと、syslogメッセージFTD-3-209006の発生を監視します。

ACIの問題

ACIポッドでのアクティブなL4チェックサム検証によるクラスタ経由の接続の断続的な問題

症状

- ACIポッドに導入されたASA/FTDクラスタを介した接続が断続的に発生する問題。
- クラスタ内にユニットが1台しかない場合は、接続の問題は発生しません。
- 1つのクラスタユニットからクラスタ内の他の1つ以上のユニットに送信されたパケットは、FXOSおよびターゲットユニットのデータプレーンキャプチャでは見えません。



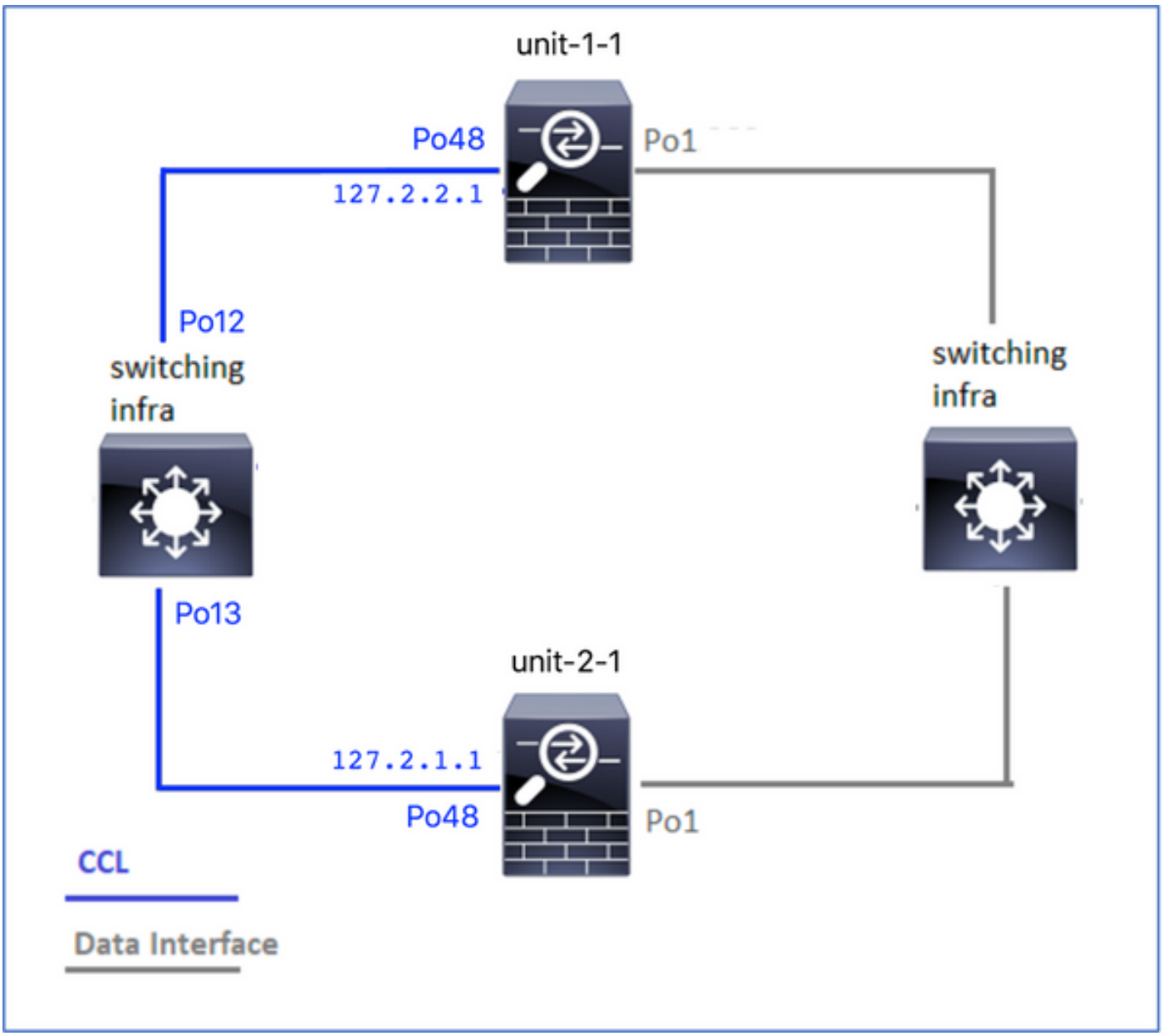
緩和

- クラスタ制御リンクを経由するリダイレクトされたトラフィックには正しいL4チェックサムがなく、これは正常な動作です。クラスタ制御リンクパス上のスイッチは、L4チェックサムを検証しません。L4チェックサムを確認するスイッチにより、トラフィックがドロップされる可能性があります。ACIファブリックスイッチの設定をチェックし、クラスタ制御リンク経由で送受信されたパケットに対してL4チェックサムが実行されていないことを確認します。

クラスタコントロールプレーンの問題

ユニットがクラスタに参加できない

CCLのMTUサイズ



症状

ユニットがクラスタに参加できず、次のメッセージが表示されます。

```
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
```

検証/緩和

- FTDでshow interfaceコマンドを使用して、クラスタ制御リンクインターフェイスのMTUが、データインターフェイスのMTUよりも少なくとも100バイト高いことを確認します。

<#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.

- サイズオプションを指定してCCL経由でpingを実行し、CCL MTU上で設定されているかどうかを、パス内のすべてのデバイスで正しく設定されているかどうかを確認します。

<#root>

firepower#

ping 127.2.1.1 size 9184

- スイッチでshow interfaceコマンドを使用して、MTU設定を確認します

<#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

クラスタユニット間のインターフェイスの不一致

症状

ユニットがクラスタに参加できず、次のメッセージが表示されます。

```
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 erro
All data interfaces have been shutdown due to clustering being disabled. To recover either enable clust
```

検証/緩和

各シャーシのFCM GUIに口グインして、Interfacesタブに移動し、すべてのクラスタメンバーのインターフェイス設定が同じであるかどうかを確認します。

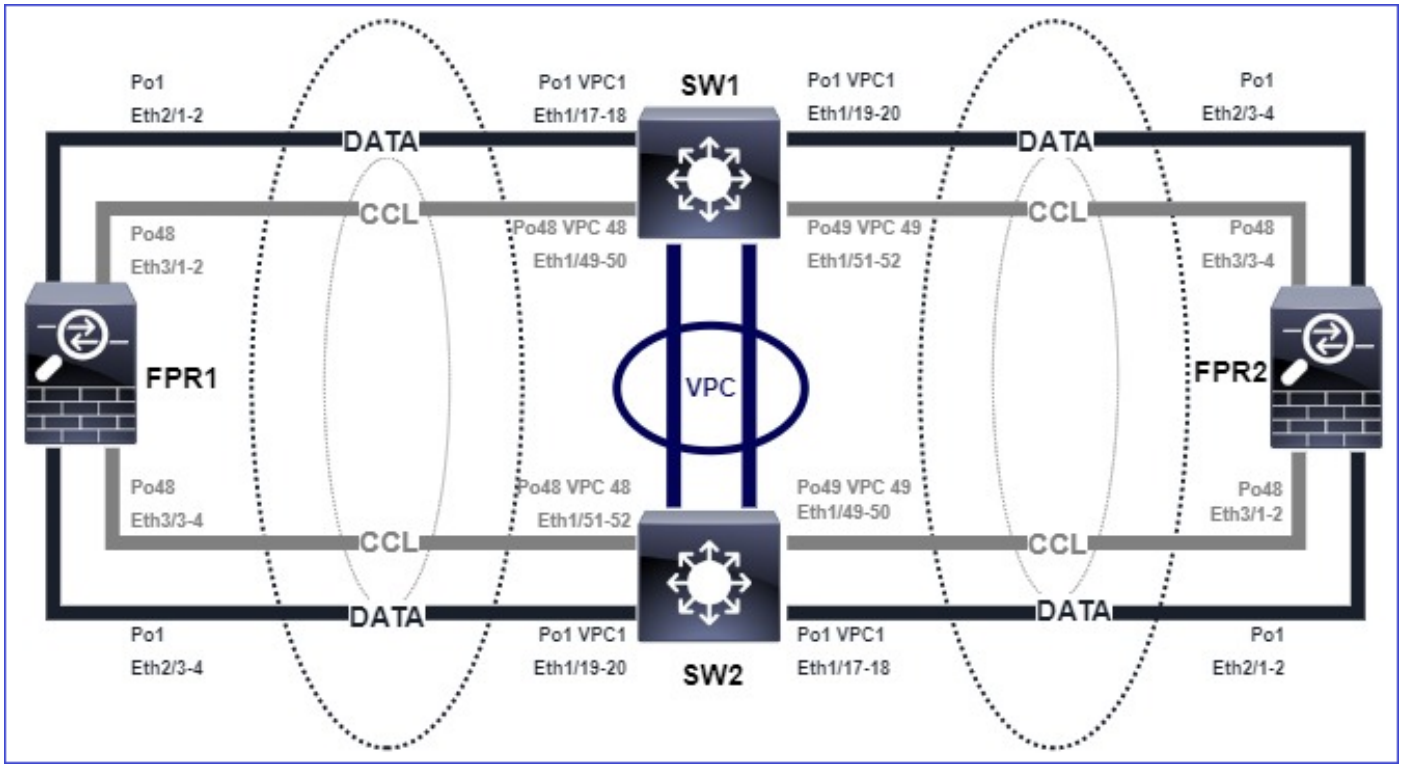
- 論理デバイスに割り当てられたインターフェイス
- インターフェイスの管理速度
- インターフェイスの管理デュプレックス
- インターフェイスのステータス

データ/ポートチャネルインターフェイスの問題

CCL経由の到達可能性の問題によるスプリットブレイン

症状

クラスタ内には複数の制御ユニットがあります。このトポロジを参照してください。



Chassis 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
```

Chassis 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
```

検証

- pingコマンドを使用して、制御ユニットのクラスタ制御リンク(CCL)IPアドレス間の接続を確認します。

<#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)
```

- ARPテーブルをチェックします。

<#root>

```
firepower# show arp


cluster 127.2.2.3 0015.c500.026f 1
cluster 127.2.2.2 0015.c500.029f 1
```

- コントロールユニットで、CCLインターフェイスのキャプチャを設定および確認します。

<#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
```

緩和

- CCLポートチャネルインターフェイスが、スイッチ上の個別のポートチャネルインターフェイスに接続されていることを確認します。
- Nexusスイッチで仮想ポートチャネル(vPC)を使用する場合は、CCLポートチャネルインタ

ーフェイスが別のvPCに接続されていること、およびvPC設定の整合性失敗ステータスが表示されていないことを確認します。
- CCLポートチャネルインターフェイスが同じブロードキャストドメインにあり、CCL VLANが作成されてインターフェイスで許可されていることを確認します。

スイッチの設定例を次に示します。

```
<#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

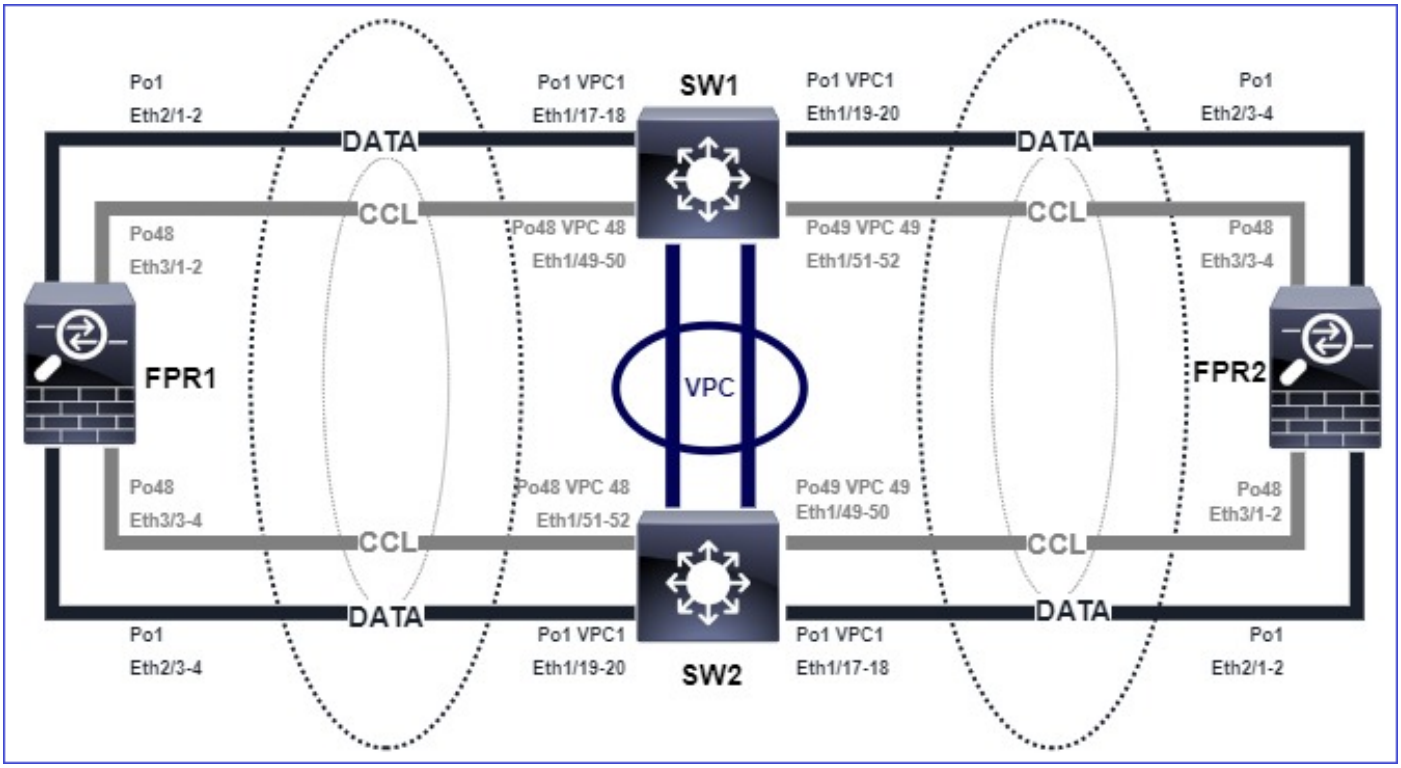
49 Po49 up success success 48

データポートチャネルインターフェイスの中断によるクラスタの無効化

症状

1つ以上のデータポートチャネルインターフェイスが一時停止しています。管理上有効になっているデータインターフェイスが中断されると、インターフェイスのヘルスチェックに失敗したため、同じシャーシ内のすべてのクラスタユニットがクラスタから除外されます。

このトポロジを参照してください。



検証

- コントロールユニットのコンソールを確認します。

<#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.

- 該当するユニットでshow cluster historyコマンドとshow cluster info trace module hcコマンドの出力を確認します。

<#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
```

- fxosコマンドシェルでshow port-channel summaryコマンドの出力を調べます。

```
<#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)
```

緩和

- すべてのシャーシのクラスタグループ名とパスワードが同じであることを確認します。
- すべてのシャーシとスイッチで、ポートチャネルインターフェイスの物理メンバーインターフェイスが管理上有効になっており、同じデュプレックス/速度設定であることを確認します。
- サイト内クラスタでは、すべてのシャーシ内の同じデータポートチャネルインターフェイスが、スイッチ上の同じポートチャネルインターフェイスに接続されていることを確認します。
- Nexusスイッチで仮想ポートチャネル(vPC)を使用する場合は、vPC設定の整合性ステータスが「失敗」になっていないことを確認します。
- サイト内クラスタでは、すべてのシャーシで同じデータポートチャネルインターフェイスが同じvPCに接続されていることを確認します。

クラスタの安定性の問題

FXOSトレースバック

症状

ユニットがクラスタから離脱します。

検証/緩和

- show cluster historyコマンドを使用して、装置がいつクラスタを離れたかを確認します

<#root>

```
firepower#
show cluster history
```

- FXOSにトレースバックがあるかどうかを確認するには、次のコマンドを使用します

<#root>

```
FPR4150#
connect local-mgmt

FPR4150 (local-mgmt)#
dir cores
```

- ユニットがクラスタを離れた時間に生成されたコアファイルを収集し、TACに提供します。

ディスクがいっぱいです

クラスタユニットの/ngfwパーティションのディスク使用率が94 %に達すると、ユニットはクラスタを終了します。ディスク使用率のチェックは3秒ごとに行われます。

<#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
```

この場合、show cluster historyの出力は次のように表示されます。

```
<#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
```

または

```
14:07:26 CEST May 18 2021
SECONDARY DISABLED Received control message DISABLE (application health check failure)
```

障害を確認するもう1つの方法は次のとおりです。

```
<#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
```

また、ディスクが100 %以下の場合、解放されるディスク領域があるまで、ユニットがクラスタに参加して戻ることが困難な場合があります。

オーバーフロー保護

5分ごとに、各クラスタユニットはローカルおよびピアユニットでCPUとメモリの使用率をチェックします。使用率がシステムのしきい値（LINA CPU 50 %またはLINAメモリ59 %）を超える場合、次のような情報メッセージが表示されます。

- Syslog(FTD-6-748008)
- ファイルlog/cluster_trace.log。例：

<#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 [

CPU 50% | Memory 59%

]. 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

このメッセージは、ユニットに障害が発生した場合に他のユニットのリソースがオーバーサブスクライブされる可能性があることを示しています。

簡易モード

6.3より前のFMCリリースの動作

- FMCで各クラスタノードを個別に登録します。
- 次に、FMCで論理クラスタを形成します。
- 新しいクラスタノードを追加するたびに、ノードを手動で登録する必要があります。

6.3以降のFMC

- 簡易モード機能を使用すると、1回の手順でクラスタ全体をFMCに登録できます（クラスタの任意の1ノードを登録するだけです）。

サポートされる最小マネージャ	管理対象デバイス	サポートされる管理対象デバイスの最小バージョンが必要	注意事項
FMC 6.3	FP9300および	6.2.0	これはFMC機能のみで

	FP4100上のFTDクラスタのみ		す

警告: FTDでクラスタが形成されたら、自動登録が開始されるまで待機する必要があります。クラスタノードを手動で登録するのではなく（[デバイスの追加]）、[調整]オプションを使用してください。

症状

ノード登録の失敗

- 制御ノードの登録が何らかの理由で失敗すると、クラスタはFMCから削除されます。

緩和

何らかの理由でデータノードの登録が失敗した場合は、次の2つのオプションがあります。

1. クラスタに展開するたびに、FMCは登録する必要があるクラスタノードがあるかどうかを確認し、これらのノードの自動登録を開始します。
2. Cluster Summaryタブ(Devices > Device Management > Cluster tab > View Cluster Statusリンク)の下に、Reconcileオプションがあります。 調整アクションがトリガーされると、FMCは登録する必要があるノードの自動登録を開始します。

関連情報

- Firepower Threat Defenseのクラスタリング
- Firepower 4100/9300シャーシ用ASAクラスタ
- Firepower 4100/9300シャーシのクラスタリングについて
- Firepower NGFWクラスタリングの詳細 – BRKSEC-3032
- ネットワークの問題を効果的にトラブルシューティングするための Firepower ファイアウォールキャプチャの分析

翻訳について
シスコは世界中のユーザにそれぞれの言語でサポート コンテンツを提供するために、機械と人による翻訳を組み合わせて、本ドキュメントを翻訳しています。ただし、最高度の機械翻訳であっても、専門家による翻訳のような正確性は確保されません。シスコは、これら翻訳の正確性について法的責任を負いません。原典である英語版（リンクからアクセス可能）もあわせて参照することを推奨します。