



## テストとトラブルシューティング

---

この章では、ASA のトラブルシューティング方法と基本接続のテスト方法について説明します。

- [イネーブルパスワードと Telnet パスワードの回復](#) (1 ページ)
- [デバッグ メッセージの表示](#) (5 ページ)
- [パケットキャプチャ](#) (6 ページ)
- [クラッシュ ダンプの表示](#) (13 ページ)
- [コア ダンプの表示](#) (13 ページ)
- [CPU 使用率とレポート](#) (13 ページ)
- [設定のテスト](#) (19 ページ)
- [接続のモニタリング](#) (33 ページ)
- [ASA Virtual での同意トークンを使用したルートシェルアクセスのサポート](#) (33 ページ)
- [テストおよびトラブルシューティングの履歴](#) (35 ページ)

## イネーブルパスワードと Telnet パスワードの回復

ASA 仮想 および ISA 3000 モデルでは、イネーブルパスワードまたは Telnet パスワードを忘れた場合に回復できます。CLI を使用してタスクを実行する必要があります。



- 
- (注) その他のプラットフォームでは、パスワードを忘れた場合に回復することはできません。工場出荷時のデフォルト設定に戻すことは可能で、パスワードをデフォルトにリセットできます。Firepower 4100/9300 の場合は、『[FXOS configuration guide](#)』を参照してください。他のモデルについては、『[FXOS troubleshooting guide](#)』を参照してください。
- 

## ISA 3000 でのパスワードの回復

ISA 3000 のパスワードの回復には、次の手順を実行します。

## 手順

- ステップ1** ASA のコンソール ポートに接続します。
- ステップ2** ASA の電源を切ってから、再び電源をオンにします。
- ステップ3** スタートアップ後、ROMMONモードに入るようにプロンプトが表示されたら、**Escape** キーを押します。
- ステップ4** コンフィギュレーション レジスタ値をアップデートするには、次のコマンドを入力します。

```
rommon #1> confreg 0x41
```

```
You must reset or power cycle for new config to take effect
```

ASA で現在のコンフィギュレーションレジスタ値と構成オプションのリストが表示されます。後で回復できるように、現在のコンフィギュレーションのレジスタ値を記録します。

```
Configuration Register: 0x00000041
```

```
Configuration Summary
```

```
[ 0 ] password recovery  
[ 1 ] display break prompt  
[ 2 ] ignore system configuration  
[ 3 ] auto-boot image in disks  
[ 4 ] console baud: 9600  
boot: ..... auto-boot index 1 image in disks
```

- ステップ5** 次のコマンドを入力して、ASA をリロードします。

```
rommon #2> boot  
Launching BootLoader...  
Boot configuration file contains 1 entry.  
  
Loading disk0:/asa932-226-k8.bin... Booting...Loading...
```

ASA は、スタートアップコンフィギュレーションの代わりにデフォルトコンフィギュレーションをロードします。

- ステップ6** 次のコマンドを入力して、特権 EXEC モードにアクセスします。

```
ciscoasa# enable
```

- ステップ7** パスワードの入力を求められたら、**Enter** キーを押します。  
パスワードは空白です。

- ステップ8** 次のコマンドを入力して、スタートアップコンフィギュレーションをロードします。

```
ciscoasa# copy startup-config running-config
```

**ステップ 9** 次のコマンドを入力して、グローバル コンフィギュレーション モードにアクセスします。

```
ciscoasa# configure terminal
```

**ステップ 10** 次のコマンドを入力して、デフォルト コンフィギュレーションで必要に応じてパスワードを変更します。

```
ciscoasa(config)# password password  
ciscoasa(config)# enable password password  
ciscoasa(config)# username name password password
```

**ステップ 11** 次のコマンドを入力して、デフォルト コンフィギュレーションをロードします。

```
ciscoasa(config)# no config-register
```

デフォルト コンフィギュレーションのレジスタ値は 0x1 です。コンフィギュレーション レジスタの詳細については、[コマンドリファレンス](#)を参照してください。

**ステップ 12** 次のコマンドを入力して、新しいパスワードをスタートアップコンフィギュレーションに保存します。

```
ciscoasa(config)# copy running-config startup-config
```

---

## ASA 仮想 のパスワードまたはイメージの回復

ASA 仮想 のパスワードまたはイメージを回復するには、次の手順を実行します。

### 手順

---

**ステップ 1** 実行コンフィギュレーションを ASA 仮想 のバックアップ ファイルにコピーします。

```
copy running-config filename
```

例 :

```
ciscoasa# copy running-config backup.cfg
```

**ステップ 2** ASA 仮想 を再起動します。

```
reload
```

**ステップ 3** [GNUGRUB] メニューから、下矢印を押し、**コンフィギュレーションをロードしないオプション**で <filename> を選択し、Enter キーを押します。ファイル名は、ASA 仮想 のデフォルトのブートイメージのファイル名です。デフォルトのブートイメージは、**fallback** コマンドによっ

て自動的にブートされることはありません。その後、選択したブートイメージをロードします。

```
GNU GRUB version 2.0(12)4
bootflash:/asa100123-20-smp-k8.bin
bootflash: /asa100123-20-smp-k8.bin with no configuration load
```

例：

```
GNU GRUB version 2.0(12)4
bootflash: /asa100123-20-smp-k8.bin with no configuration load
```

**ステップ4** 実行コンフィギュレーションにバックアップ コンフィギュレーション ファイルをコピーします。

**copy filename running-config**

例：

```
ciscoasa (config)# copy backup.cfg running-config
```

**ステップ5** パスワードのリセット。

**enable password password**

例：

```
ciscoasa(config)# enable password cisco123
```

**ステップ6** 新しい設定を保存します。

**write memory**

例：

```
ciscoasa(config)# write memory
```

---

## ISA 3000 ハードウェアのパスワード回復の無効化



---

(注) ASA 仮想、Cisco Secure Firewall モデルでパスワード回復をディセーブルにすることはできません。

---

権限のないユーザーがパスワード回復メカニズムを使用して ASA を危険にさらすことがないように、パスワード回復をディセーブルにするには、次の手順を実行します。

## 始める前に

ASA で、**no service password-recovery** コマンドを使用すると ROMMON モードに入って、コンフィギュレーションの変更を防ぐことができます。ROMMON モードに入ると、ASA では、すべてのフラッシュ ファイル システムの消去を求めるプロンプトが表示されます。最初に消去を実行しないと、ROMMON モードを開始できません。フラッシュ ファイル システムを消去しない場合、ASA はリロードされます。パスワード回復は ROMMON モードの使用と既存のコンフィギュレーションの保持に依存しているため、この消去によって、パスワードの回復ができなくなります。ただし、パスワードを回復できなくすることで、不正なユーザーがコンフィギュレーションを表示したり、別のパスワードを挿入したりすることがなくなります。この場合、システムを動作可能な状態に回復するには、新しいイメージとバックアップコンフィギュレーション ファイル（入手できる場合）をロードします。

**service password-recovery** コマンドは、コンフィギュレーション ファイルに通知用としてのみ表示されます。CLI プロンプトに対してコマンドを入力すると、設定は NVRAM に保存されます。設定を変更する唯一の方法は、CLI プロンプトでコマンドを入力することです。このコマンドの異なるバージョンで新規コンフィギュレーションをロードしても、設定は変更されません。（パスワード回復の準備段階で）スタートアップ時にスタートアップ コンフィギュレーションを無視するよう ASA が設定されている場合にパスワード回復をディセーブルにすると、通常どおりスタートアップ コンフィギュレーションをロードするように ASA の設定が変更されます。フェールオーバーを使用し、スタートアップ コンフィギュレーションを無視するようスタンバイ装置が設定されている場合は、**no service password-recovery** コマンドでスタンバイ装置に複製したときに、コンフィギュレーション レジスタに同じ変更が加えられます。

## 手順

---

パスワード回復をディセーブルにします。

**no service password-recovery**

例：

```
ciscoasa (config)# no service password-recovery
```

---

# デバッグメッセージの表示

デバッグ出力は CPU プロセスで高プライオリティが割り当てられているため、デバッグ出力を行うとシステムが使用できなくなることがあります。このため、特定の問題のトラブルシューティングを行う場合や、Cisco TAC とのトラブルシューティングセッションの間に限り **debug** コマンドを使用してください。さらに、ネットワークトラフィック量やユーザ数が少ない期間に **debug** コマンドを使用することをお勧めします。デバッグをこのような時間帯に行うと、**debug** コマンド処理のオーバーヘッドの増加によりシステムの使用に影響が及ぶ可能性が少な

くなります。デバッグメッセージを有効にするには、コマンドリファレンスの **debug** コマンドを参照してください。

## パケットキャプチャ

パケットキャプチャは、接続の問題のトラブルシューティングまたは不審なアクティビティのモニタリングを行うときに役立つことがあります。パケットキャプチャサービスを使用する場合は、Cisco TAC に連絡することをお勧めします。

## パケットキャプチャのガイドライン

### コンテキストモード

- コンテキスト内のクラスタ制御リンクでキャプチャを設定できます。この場合、そのクラスタ制御リンクで送信されるコンテキストに関連付けられているパケットだけがキャプチャされます。
- VLAN ごとに設定できるキャプチャは 1 つだけです。共有 VLAN の複数のコンテキストでキャプチャを設定した場合は、最後に設定したキャプチャだけが使用されます。
- 最後に設定した（アクティブ）キャプチャを削除した場合は、別のコンテキストで事前に設定したキャプチャがあっても、アクティブになるキャプチャはありません。キャプチャをアクティブにするには、キャプチャを削除して追加し直す必要があります。
- キャプチャを指定したインターフェイスに着信するすべてのトラフィックがキャプチャされます。これには、共有 VLAN 上の他のコンテキストへのトラフィックも含まれます。したがって、ある VLAN のコンテキスト A でのキャプチャをイネーブルにしたときに、その VLAN がコンテキスト B でも使用される場合は、コンテキスト A とコンテキスト B の両方の入力トラフィックがキャプチャされます。
- 出力トラフィックの場合は、アクティブキャプチャのあるコンテキストのトラフィックだけがキャプチャされます。唯一の例外は、ICMP 検査をイネーブルにしない（したがって、ICMP トラフィックのセッションが高速パスにない）場合です。この場合は、共有 VLAN のすべてのコンテキストで入力と出力の ICMP トラフィックがキャプチャされます。

### その他のガイドライン

- ASA が不正な形式の TCP ヘッダーを持つパケットを受信し、ASP が *invalid-tcp-hdr-length* であるというドロップ理由でそのパケットをドロップする場合、そのパケットを受信したインターフェイス上の **show capture** コマンド出力は、そのパケットを表示しません。
- IP トラフィックだけをキャプチャできます。ARP などの非 IP パケットはキャプチャできません。
- インライン SGT タグ付きパケットの場合、キャプチャされたパケットに含まれている追加 CMD ヘッダーを、PCAP ビューアが認識しないことがあります。

- パケットキャプチャには、システムを変更する、またはインスペクションのために接続に挿入されるパケット、NAT、TCPの正規化、パケットの内容を調整するその他の機能が含まれます。
- データパスに挿入された仮想パケットの寿命のトレースは、データパスでの物理パケットの処理を正確に反映していません。この違いは、ソフトウェアバージョン、構成、および挿入された仮想パケットのタイプによって異なります。違いが生じる原因となる可能性がある構成の設定を次に示します。
  - 同じホストに対して2つ以上のNATステートメントが存在する。
  - 接続の順方向と逆方向のフローでプロトコルが異なる（順方向のフローがUDPまたはTCPで、逆方向のフローがICMPである場合など）。
  - ICMPエラーインスペクションが有効になっている。

## パケットのキャプチャ

パケットをキャプチャするには、次の手順を実行します。

### 手順

- ステップ1** パケットスニффイングおよびネットワーク障害の切り分けのためにパケットキャプチャ機能をイネーブルにします。

```
capture capture_name [type {asp-drop [all | drop-code] | tls-proxy | raw-data | isakmp [ikev1 | ikev2] | inline-tag [tag] | webvpn user webvpn-user}] [access-list access_list_name] [interface {interface_name | asa_dataplane | asa_mgmt_plane | cplane}] [buffer buf_size] [ethernet-type type] [reinject-hide] [packet-length bytes] [circular-buffer] [trace [trace-count number]] [real-time [dump] [detail]] [file-size] [headers-only] [match protocol {host source-ip | source-ip mask | any | any4|any6}] [operator src_port] [host dest_ip | dest_ip mask | any | any4|any6] [operator dest_port]
```

例：

```
ciscoasa# capture capttest interface inside
```

キャプチャするすべてのパケットのインターフェイスを設定する必要があります。複数のタイプのトラフィックをキャプチャするには、複数の **capture** ステートメントで同じ *capture\_name* を使用します。

**type asp-drop** キーワードは、高速セキュリティパスでドロップされるパケットをキャプチャします。クラスタでは、ドロップされた、ユニット間の転送データパケットもキャプチャされます。マルチ コンテキスト モードでは、このオプションがシステム実行スペースで発行されると、すべてのドロップされたデータパケットがキャプチャされます。このオプションがコンテキストで発行されたときは、ドロップされたデータパケットのうち、そのコンテキストに属するインターフェイスから入ったものだけがキャプチャされます。

**type raw-data** キーワードは、着信パケットと発信パケットをキャプチャします。この設定は、デフォルトです。

**inline-tag tag** のキーワードと引数のペアは、特定の SGT 値のタグを指定します。指定しない場合は、任意の SGT 値を持つタグ付きパケットをキャプチャします。

**buffer** キーワードは、パケットを保存するために使用するバッファサイズを定義します。このバイトバッファがいっぱいになると、パケットキャプチャは停止します。クラスタ内で使用されるときは、これはユニットあたりのサイズです（全ユニットの合計ではありません）。

**circular-buffer** キーワードを指定すると、バッファがいっぱいになったときに、バッファが先頭から順に上書きされます。

**ethernet-type** キーワードは、キャプチャするイーサネットタイプを設定します。サポートされるイーサネットタイプには、8021Q、ARP、IP、IP6、LACP、PPPOED、PPPOES、RARP、および VLAN があります。802.1Q タイプと VLAN タイプでは例外が発生します。802.1Q タグは自動的にスキップされ、照合には内部イーサネットタイプが使用されます。IP はデフォルトのイーサネットタイプです。

**interface** キーワードは、パケットキャプチャを使用するインターフェイスの名前を設定します。

データプレーン上のパケットをキャプチャするには、**asa\_dataplane** キーワードを使用します。

キャプチャファイルのサイズを設定するには、**file-size** キーワードを使用します。ファイルサイズは 32 - 10000 MB です。

データを含まないパケットの L2、L3、および L4 ヘッダーだけをキャプチャする場合は、**headers-only** コマンドを使用します。

**match** キーワードは、一致するプロトコルおよび送信元と宛先 IP アドレス、およびオプションのポートをキャプチャします。このキーワードは、1つのコマンドで3回まで使用できます。

**any** キーワードは、IPv4 トラフィックだけをキャプチャします。**any4** および **any6** キーワードを使用して、一致する IPv4 および IPv6 ネットワークトラフィックを個別にキャプチャできます。**operator** には次のいずれかを指定できます。

- lt : より小さい
- gt : より大きい
- eq : 等しい

**real-time** キーワードを指定すると、キャプチャしたパケットがリアルタイムで連続して表示されます。

**reinject-hide** キーワードを指定すると、再注入されたパケットはキャプチャされません。これは、クラスタリング環境にのみ適用されます。

(注)

ACL の最適化が設定されている場合、**access-list** コマンドはキャプチャでは使用できません。**access-group** コマンドのみ使用できます。この場合、**access-list** コマンドを使用しようとするときエラーが表示されます。

**ステップ 2** クラスタ制御リンクのトラフィックをキャプチャします。

```
capture capture_name { type lacp interface interface_id [ buffer buf_size] [ packet-length bytes]
[circular-buffer] [ real-time [ dump] [ detail]
```

```
capture capture_name interface cluster [ buffer buf_size] [cp-cluster] [ ethernet-type type] [
packet-length bytes] [circular-buffer] [trace [ trace-count number]] [real-time [ dump] [ detail]] [trace]
[ match protocol { host source-ip | source-ip mask | any | any4|any6} [ operator src_port] { host dest_ip
| dest_ip mask | any | any4|any6} [ operator dest_port]]
```

例：

```
ciscoasa# capture ccl type lacp interface GigabitEthernet0/0
ciscoasa# capture ccl interface cluster match udp any eq 49495 any
ciscoasa# capture ccl interface cluster match udp any any eq 49495
```

次の2つの方法でクラスタ制御リンクのトラフィックをキャプチャできます。クラスタ制御リンクのすべてのトラフィックをキャプチャするには、インターフェイス名に **cluster** キーワードを使用します。cLACPパケットのみをキャプチャするには **type lacp** を指定し、インターフェイス名ではなく物理インターフェイス ID を指定します。クラスタ制御リンク上のパケットには、コントロールプレーンパケットとデータプレーンパケットの2種類があり、どちらも、転送されたデータトラフィックとクラスタLUメッセージが含まれています。IPアドレスヘッダーの TTL フィールドは、この2種類のパケットを区別できるように符号化されます。転送されたデータパケットがキャプチャされる場合は、デバッグのためにクラスタリングトレラもキャプチャファイルに出力されます。

キーワード **cp-cluster** はクラスタ制御リンク（およびデータプレーンパケットなし）でコントロールプレーンパケットのみをキャプチャできるようになりました。このオプションは、マルチコンテキストモードのシステムで、ACLを使用してトラフィックを照合できない場合に役立ちます。

**ステップ 3** クラスタ全体のパケットをキャプチャします。

```
cluster exec capture capture_name arguments
```

**ステップ 4** スイッチの出力通信パケットをキャプチャします（Cisco Secure Firewall 4200 および 6100 モデルデバイスでのみサポートされています）。

```
capture capture_name switch interface interface_name direction egress
```

（注）

**both** 引数を使用して、スイッチの出力トラフィックと入力トラフィックの両方のキャプチャを作成します。

**ステップ 5** パケットキャプチャを停止します。

```
no capture capture_name
```

リアルタイムパケットキャプチャを終了するには、**Ctrl+c** を入力します。キャプチャを完全に削除するには、このコマンドの **no** 形式を使用します。リアルタイムオプションは、**raw-data** キャプチャおよび **asp-drop** キャプチャにのみ適用されます。

**ステップ 6** バッファからパケットを削除せずに手動でパケットキャプチャを停止する場合：

**capture name stop**

**ステップ 7** 再度キャプチャを開始する場合：

**no capture namestop**

**ステップ 8** クラスタ ユニットで永続的なパケット トレースをキャプチャします。

**cluster exec capture\_test persist**

**ステップ 9** 永続的なパケット トレースをクリアします。

**cluster exec clear packet-trace**

**ステップ 10** 復号化された IPsec パケットをキャプチャします。

**cluster exec capture\_test include-decryptd**

**ステップ 11** キャプチャをクリアします。

**clear capture capture\_name**

**例****コントロール プレーン パケット**

コントロールプレーンと通信するすべてのパケットは TTL が 255 に設定されており、ポート番号 49495 がクラスタリング コントロール プレーン リッスン ポートに使用されます。次の例では、クラスタリング環境の LACP キャプチャを作成する方法を示します。

```
ciscoasa# capture lacp type lacp interface GigabitEthernet0/0
```

次の例では、クラスタリングリンクでの制御パスパケットのキャプチャを作成する方法を示します。

```
ciscoasa# capture cp interface cluster match udp any eq 49495 any
ciscoasa# capture cp interface cluster match udp any any eq 49495
```

**データ プレーン パケット**

データ パケットには、1つのユニットから別のユニット（その接続の所有者）に転送されるパケットと、クラスタ LU メッセージが含まれます。通常のクラスタ LU 更新メッセージは、TTL が 254 に設定されており、TTL が 253 に設定された特別な LU パケットがあります。この特別な LU パケットは TCP のみで、ディレクタが新しいフローの所有者を選択した場合にのみ発生します。ディレクタは CLU\_FULL アップデート パケットとともに要求パケットを送り返します。LU パケットには、元のパケットの L3/L4 ヘッダーが書き込まれます。これにより、受信者側で潜在的な競合状態が発生するのを回避できます。転送されるデータ パケットは、TTL が 4 未満に設定されません。次の例では、クラスタ制御リンクでデータパスパケットのキャプチャを作成する

方法を示します。クラスタ間データプレーンの「flow logical update」メッセージをすべてキャプチャするには、ポート 4193 を使用します。

```
ciscoasa# access-list ccl extended permit udp any any eq 4193
ciscoasa# access-list ccl extended permit udp any eq 4193 any
ciscoasa# capture dp interface cluster access-list ccl
```

## パケットキャプチャの表示

CLIでパケットキャプチャをブラウザ上に表示したり、任意のサーバーにキャプチャをダウンロードしたりすることができます。

### 手順

**ステップ 1** CLIでキャプチャを表示するには：

```
[cluster exec] show capture [capture_name] [ access-list access_list_name] [ count number] [decode]
[detail] [dump] [ packet-number number]
```

例：

```
ciscoasa# show capture capin

 8 packets captured

 1: 03:24:35.526812      192.168.10.10 > 203.0.113.3: icmp: echo request
 2: 03:24:35.527224      203.0.113.3 > 192.168.10.10: icmp: echo reply
 3: 03:24:35.528247      192.168.10.10 > 203.0.113.3: icmp: echo request
 4: 03:24:35.528582      203.0.113.3 > 192.168.10.10: icmp: echo reply
 5: 03:24:35.529345      192.168.10.10 > 203.0.113.3: icmp: echo request
 6: 03:24:35.529681      203.0.113.3 > 192.168.10.10: icmp: echo reply
 7: 03:24:57.440162      192.168.10.10 > 203.0.113.3: icmp: echo request
 8: 03:24:57.440757      203.0.113.3 > 192.168.10.10: icmp: echo reply
```

**access-list** キーワードは、特定のアクセスリストIDのIPフィールドまたはより高位のフィールドに基づいて、パケットに関する情報を表示します。

**cluster exec** キーワードを使用すると、あるユニットで **show capture** コマンドを発行し、他のすべてのユニットでそのコマンドを同時に実行できます。

**count** キーワードは、指定したデータのパケット数を表示します。

**decode** キーワードは、**isakmp** タイプのキャプチャがインターフェイスに適用される場合に役立ちます。当該のインターフェイスを通過する ISAKMP データは、復号化の後にすべてキャプチャされ、フィールドをデコードした後にその他の情報とともに表示されます。パケットのデコード出力は、パケットのプロトコルによって異なります。通常、このコマンドは、ICMP、UDP、およびTCPプロトコルのIPデコードをサポートします。バージョン9.10(1)から、このコマンドはGREおよびIPinIPのIPデコードもサポートします。

**detail** キーワードは、各パケットの追加のプロトコル情報を表示します。

**dump** キーワードは、データ リンク経由で転送されたパケットの 16 進ダンプを表示します。

**packet-number** キーワードは、指定したパケット番号で表示を開始します。

**ステップ 2** ブラウザでパケットキャプチャを表示するには：

**https://ip\_of\_asaladmin/capture/capture\_name/pcap**

**pcap** キーワードを省略すると、**show capture capture\_name** コマンド出力に相当する内容のみが表示されます。

マルチ コンテキスト モードでは、システム実行スペースでのみ **copy capture** コマンドを使用できます。

**ステップ 3** パケットキャプチャをサーバーにコピーします。この例では FTP を示します。

**[cluster exec] copy /pcap capture:[context-name]/capture\_name ftp://username:password@server\_ip/path**

**pcap** キーワードを省略すると、**show capture capture\_name** コマンド出力に相当する内容のみが表示されます。

(注)

パケットキャプチャをディスクにコピーする場合は、キャプチャファイル名が 63 文字以下であることを確認してください。ファイル名が 63 文字を超える場合、パケットキャプチャは成功しますが、ディスクへのキャプチャのコピーは失敗します。

## 例

次の例は、**asp-drop** タイプのキャプチャを示します。

```
ciscoasa# capture asp-drop type asp-drop acl-drop
ciscoasa# show capture asp-drop

 2 packets captured

1: 04:12:10.428093      192.168.10.10.34327 > 10.94.0.51.15868: S
 2669456341:2669456341(0) win 4128 <mss 536> Drop-reason: (acl-drop)
  Flow is denied by configured rule
2: 04:12:12.427330      192.168.10.10.34327 > 10.94.0.51.15868: S
 2669456341:2669456341(0) win 4128 <mss 536> Drop-reason: (acl-drop)
  Flow is denied by configured rule
2 packets shown

ciscoasa# show capture asp-drop

 2 packets captured

1: 04:12:10.428093      192.168.10.10.34327 > 10.94.0.51.15868: S
 2669456341:2669456341(0) win 4128 <mss 536> Drop-reason: (acl-drop)
  Flow is denied by configured rule
2: 04:12:12.427330      192.168.10.10.34327 > 10.94.0.51.15868: S
 2669456341:2669456341(0) win 4128 <mss 536> Drop-reason: (acl-drop)
  Flow is denied by configured rule
2 packets shown
```

次の例は、ethernet タイプのキャプチャを示します。

```
ciscoasa# capture arp ethernet-type arp interface inside
ciscoasa# show cap arp

22 packets captured

  1: 05:32:52.119485      arp who-has 10.10.3.13 tell 10.10.3.12
  2: 05:32:52.481862      arp who-has 192.168.10.123 tell 192.168.100.100
  3: 05:32:52.481878      arp who-has 192.168.10.50 tell 192.168.100.10
  4: 05:32:53.409723      arp who-has 10.106.44.135 tell 10.106.44.244
  5: 05:32:53.772085      arp who-has 10.106.44.108 tell 10.106.44.248
  6: 05:32:54.782429      arp who-has 10.106.44.135 tell 10.106.44.244
  7: 05:32:54.784695      arp who-has 10.106.44.1 tell 11.11.11.112:
```

## クラッシュ ダンプの表示

ASA か ASA 仮想 がクラッシュした場合に、クラッシュダンプ情報を表示できます。クラッシュダンプの内容を調べる必要がある場合は、Cisco TAC に連絡することを推奨します。コマンドリファレンスで **show crashdump** コマンドを参照してください。

## コア ダンプの表示

コア ダンプは、プログラムが異常終了（クラッシュ）したときの、実行中のプログラムのスナップショットです。コアダンプは、エラーを診断またはデバッグするため、および障害を後からオフサイトで分析できるように、クラッシュを保存するために使用されます。ASA か ASA 仮想でのアプリケーションまたはシステムクラッシュをトラブルシューティングするために、コアダンプ機能を有効にするよう Cisco TAC から要請される場合があります。コマンドリファレンスで **coredump** コマンドを参照してください。

## CPU 使用率とレポート

CPU 使用率レポートには、指定された時間内に使用された CPU の割合の要約が表示されます。通常、コアはピーク時以外には合計 CPU 容量の約 30 ~ 40% で動作し、ピーク時は約 60 ~ 70% の容量で動作します。

## の vCPU 使用率 ASA 仮想

CPU 使用率の統計を表示するには、ASA 仮想 で **show cpu usage** コマンドを使用します。ASA 仮想の vCPU 使用率では、データパス、制御ポイント、および外部プロセスで使用されている vCPU の量を表示します。

(VMware、Azure、OCI などの) クラウドサービスプロバイダーによって報告される vCPU 使用率には、示されている ASA 仮想 使用率に加えて、以下が含まれます。

- ASA 仮想 のアイドル時間
- ASA VM に使用された %SYS オーバーヘッド
- vSwitch、vNIC および pNIC の間を移動するパケットのオーバーヘッド。このオーバーヘッドは非常に大きくなる場合があります。

## CPU 使用率の例

報告された vCPU の使用率が大幅に異なる例を次に示します。

- ASA 仮想 のレポート : 40%
- DP : 35%
- 外部プロセス : 5%
- vSphere のレポート : 95%
- ASA (ASA 仮想 レポートとして) : 40%
- ASA アイドル ポーリング : 10%
- オーバーヘッド : 45%

オーバーヘッドは、ハイパーバイザ機能の実行、および vSwitch を使用した NIC と vNIC の間のパケット転送に使用されています。

ASA 仮想のためのオーバーヘッドとして、ESXi サーバが追加のコンピューティングリソースを使用する場合がありますため、使用率は 100% を超えることがあります。

## VMware の CPU 使用率のレポート

vSphere で [VM Performance] タブをクリックし、[Advanced] をクリックすると [Chart Options] ドロップダウンリストが表示されます。ここには VM の各ステート (%USER、%IDLE、%SYS など) の vCPU 使用率が表示されます。この情報は、VMware の観点から CPU リソースが使用されている場所を理解するのに役立ちます。

ESXi サーバのシェル (ホストへの接続に SSH を使用してシェルにアクセスします) では、esxtop を使用できます。Esxtop は Linux の **top** コマンドに似た操作性と外観を持ち、次の内容を含む vSphere のパフォーマンスに関する VM のステート情報を提供します。

- vCPU、メモリ、ネットワーク使用率の詳細
- 各 VM のステートごとの vCPU 使用率
- メモリ (実行中に「M」と入力) とネットワーク (実行中に「N」と入力) に加えて、統計情報と RX ドロップ数

## ASA 仮想 と vCenter のグラフ

ASA 仮想 と vCenter の CPU 使用率の数値には違いがあります。

- vCenter のグラフの数値は常に ASA 仮想 の数値よりも大きくなります。
- vCenter ではこの値は %CPU usage と呼ばれ、ASA 仮想 ではこの値は %CPU utilization と呼ばれます。

用語「%CPU utilization」と「%CPU usage」は別のものを意味しています。

- CPU utilization は、物理 CPU の統計情報を提供します。
- CPU usage は CPU のハイパースレッディングに基づいた論理 CPU の統計情報を提供します。しかし、1 つの vCPU のみが使用されるため、ハイパースレッディングは動作しません。

vCenter は CPU % usage を次のように計算します。

アクティブに使用された仮想 CPU の量。使用可能な CPU の合計に対する割合として指定されます。

この計算は、ホストから見た CPU 使用率であり、ゲストオペレーティングシステムから見た CPU 使用率ではありません。また、これは仮想マシンで使用可能なすべての仮想 CPU の平均 CPU 使用率になります。

たとえば、1 個の仮想 CPU を搭載した 1 つの仮想マシンが、4 個の物理 CPU を搭載した 1 台のホストで実行されており、その CPU 使用率が 100% の場合、仮想マシンは、1 個の物理 CPU をすべて使用しています。仮想 CPU の使用率は、「MHz 単位の使用率 / 仮想 CPU の数 x コア周波数」として計算されます。

使用率を MHz で比較すると、vCenter と ASA 仮想 両方の数値は一致します。vCenter グラフから、MHz % CPU 使用率は  $60 / (2499 \times 1 \text{ vCPU}) = 2.4$  と求められます。

## Amazon CloudWatch CPU 使用率レポート

メトリックエクスペローラを表示して、タグとプロパティでリソースをモニターできます。特定のインスタンスの CPU 使用率の統計を表示するには、次の手順を実行します。

### 手順

- ステップ 1** [CloudWatch] コンソールを開き、ナビゲーションペインで [メトリクス (Metrics)] を選択します。
- ステップ 2** EC2 メトリクスの名前空間を選択し、[インスタンスごとのメトリクス (Per-instance Metrics)] ディメンションを選択します。
- ステップ 3** 検索フィールドに **CPUtilization** と入力して Enter を押します。必要なインスタンスの行を選択し、そのインスタンスの **CPUtilization** メトリックのグラフを表示します。

詳細については、[Amazon CloudWatch のドキュメント](#)を参照してください。

---

## ASA 仮想 と Amazon CloudWatch のグラフ

Amazon CloudWatch のグラフの数値は、CPU 使用率の計算方法が ASA 仮想 と CloudWatch で異なるため、数値よりも大きくなっています。

ASA 仮想 がポーリングモードで実行されている場合、各 CPU は、省電力モードやその他のアイドル状態に入る代わりに、軽量コマンドのループを実行します。これにより、インテルの電源状態によってオンオフを切り替えたりクロックを調整したりするのではなく、各コアが常にアクティブに保たれてパフォーマンスが向上します。

ASA 仮想 内では、このアクティビティはアイドル動作であると認識され、CPU 使用率が正しく計算されます。ただし、Amazon CloudWatch では、すべての CPU サイクルに実行する命令があるため、アイドル状態の動作は通常の CPU アクティビティのように見えます。これにより、CloudWatch では高い CPU 使用率（85 ~ 90%）が表示されます。

## Azure の CPU 使用率レポート

Azure Monitor から VM Insights を使用して、監視対象の VM すべての CPU 使用率を表示するには、次の手順を実行します。

### 手順

---

- ステップ 1** Azure ポータルに移動し、[監視 (Monitor)] を選択してから [ソリューション (Solutions)] セクションで [仮想マシン (Virtual Machines)] を選択します。
  - ステップ 2** [パフォーマンス (Performance)] タブを選択して [CPU使用率 (CPU Utilization %)] グラフを表示します。このグラフには、平均プロセッサ使用率が最も高い上位5つのマシンが表示されます。
- 

特定の Azure VM から直接 CPU 使用率グラフを表示するには、次の手順を実行します。

### 手順

---

- ステップ 1** Azure ポータルに移動し、[仮想マシン (Virtual Machines)] を選択します。
- ステップ 2** VM のリストから VM を選択します。
- ステップ 3** [モニタリング (Monitoring)] セクションで、[Insights] を選択します。
- ステップ 4** [パフォーマンス (Performance)] タブを選択します。

詳細については、「[How to chart performance with VM insights](#)」[英語]を参照してください。

## ASA 仮想 と Azure のグラフ

ASA 仮想 と Azure の CPU 使用率の数値には違いがあります。Azure は、使用可能な CPU の合計に対する割合として指定される、アクティブに使用されている仮想 CPU の量として CPU 使用率を計算するため、Azure のグラフの数値は常に ASA 仮想 の数値より高くなります。

この計算は、ホストから見た CPU 使用率であり、ゲストオペレーティングシステムから見た CPU 使用率ではありません。また、これは仮想マシンで使用可能なすべての仮想 CPU の平均 CPU 使用率になります。

たとえば、1 個の仮想 CPU を搭載した 1 つの仮想マシンが、4 個の物理 CPU を搭載した 1 台のホストで実行されており、その CPU 使用率が 100% の場合、仮想マシンは、1 個の物理 CPU をすべて使用しています。仮想 CPU の使用率は、「MHz 単位の使用率/仮想 CPU の数 X コア周波数」として計算されます。

Azure は、ゲスト OS によって要求される CPU の量にもレート制限を適用します。ASA 仮想 が 40% の CPU 使用率を報告し、ハイパーバイザが 90% の CPU 使用率を報告しているシナリオについて考えてみましょう。ここで ASA 仮想 がさらなる処理能力を求めた場合、CPU 使用率が 80% を超え、ハイパーバイザが 95% を超える CPU 使用率を報告する可能性があります。これにより、ASA 仮想 がポーリングモードで軽量コマンドのループを実行しているだけでアイドル動作を示していたとしても、ハイパーバイザは ASA 仮想 CPU をスロットリングすることになります。

## Hyper-V CPU 使用率レポート

使用可能なクラウドサーバーの CPU、RAM、およびディスク容量の構成情報の表示に加えて、ディスク、I/O、およびネットワーク情報も表示できます。この情報を使用して、ニーズに適したクラウドサーバーを決定してください。コマンドライン nova クライアントまたは **Cloud Control Panel** インターフェイスを使用して、使用可能なサーバーを表示できます。

コマンドラインで、次のコマンドを実行します。

```
nova flavor-list
```

使用可能なすべてのサーバー構成が表示されます。リストには、次の情報が含まれています。

- ID : サーバー構成 ID
- 名前 : RAM サイズとパフォーマンスタイプでラベル付けされた構成名
- Memory\_MB : 構成の RAM の量
- ディスク : GB 単位のディスクサイズ (汎用クラウドサーバーの場合、システムディスクのサイズ)
- エフェメラル : データディスクのサイズ

- スワップ：スワップ領域のサイズ
- VCPU：構成に関連付けられた仮想 CPU の数
- RXTX\_Factor：サーバーに接続された PublicNet ポート、ServiceNet ポート、および分離されたネットワーク（クラウドネットワーク）に割り当てられる帯域幅の量（Mbps 単位）
- Is\_Public：未使用

## ASA Virtual と Hyper-V のグラフ

ASA Virtual と Hyper-V の CPU 使用率の数値には違いがあります。

- Hyper-V のグラフの数値は ASA Virtual の数値よりも常に大きくなります。
- Hyper-V ではこの値は %CPU usage と呼ばれ、ASA Virtual ではこの値は %CPU utilization と呼ばれます。

用語「%CPU utilization」と「%CPU usage」は別のものを意味しています。

- CPU utilization は、物理 CPU の統計情報を提供します。
- CPU usage は CPU のハイパースレッディングに基づいた論理 CPU の統計情報を提供します。しかし、1つの vCPU のみが使用されるため、ハイパースレッディングは動作しません。

Hyper-V では %CPU usage は次のように計算されます。

アクティブに使用された仮想 CPU の量。使用可能な CPU の合計に対する割合として指定されます。

この計算は、ホストから見た CPU 使用率であり、ゲストオペレーティングシステムから見た CPU 使用率ではありません。また、これは仮想マシンで使用可能なすべての仮想 CPU の平均 CPU 使用率になります。

たとえば、1 個の仮想 CPU を搭載した 1 つの仮想マシンが、4 個の物理 CPU を搭載した 1 台のホストで実行されており、その CPU 使用率が 100% の場合、仮想マシンは、1 個の物理 CPU をすべて使用しています。仮想 CPU の使用率は、「MHz 単位の使用率 / 仮想 CPU の数 x コア周波数」として計算されます。



(注) 正確な CPU 使用率を得るには、ASA Virtual レポートを調べることをお勧めします。

## OCI CPU 使用率レポート

コンピューティングインスタンスメトリック `oci_computeagent` を使用して、OCI の CPU 使用率を表示できます。CpuUtilization メトリックは、CPU からのアクティビティレベルを表示し、

合計時間に対する割合として表されます。単一のコンピューティングインスタンスのメトリックグラフを表示するには、次の手順を実行します。

#### 手順

- ステップ1 ナビゲーションメニューを開き、[コンピューティング (Compute)] の下の [インスタンス (Instances)] をクリックします。
- ステップ2 インスタンスをクリックし、[リソース (Resources)] の下の [メトリック (Metrics)] をクリックします。
- ステップ3 メトリック名前空間リストで [oci\_computeagent] を選択します。  
詳細については、[コンピューティング インスタンス メトリック](#) を参照してください。

## ASA 仮想 と OCI のグラフ

OCI は、使用可能な CPU の合計に対する割合として指定される、アクティブに使用されている仮想 CUP の量として CPU 使用率を計算するため、OCI のグラフの数値は常に ASA 仮想の数値より高くなります。

この計算は、ホストから見た CPU 使用率であり、ゲストオペレーティングシステムから見た CPU 使用率ではありません。また、これは仮想マシンで使用可能なすべての仮想 CPU の平均 CPU 使用率になります。

たとえば、1 個の仮想 CPU を搭載した 1 つの仮想マシンが、4 個の物理 CPU を搭載した 1 台のホストで実行されており、その CPU 使用率が 100% の場合、仮想マシンは、1 個の物理 CPU をすべて使用しています。仮想 CPU の使用率は、「MHz 単位の使用率/仮想 CPU の数 X コア周波数」として計算されます。

## 設定のテスト

ここでは、シングルモード ASA または各セキュリティ コンテキストの接続性のテスト方法、ASA インターフェイスを ping する方法、およびあるインターフェイス上のホストから他のインターフェイス上のホストに ping できるようにする方法について説明します。

## 基本接続のテスト：アドレス向けの ping の実行

ping は、特定のアドレスが使用可能で、応答するかどうかを確認するための単純なコマンドです。次のトピックでは、このコマンドの詳細とそれを使って実行可能なテストについて説明します。

## ping で実行可能なテスト

デバイスを ping すると、そのデバイスにパケットが送信され、デバイスが応答を返します。このプロセスを使用して、ネットワークデバイスは、相互に検出、識別、およびテストすることができます。

ping を使用して、次のテストを実行できます。

- 2つのインターフェイスのループバック テスト：同じ ASA で一方のインターフェイスからもう一方のインターフェイスに ping を外部ループバック テストとして起動すると、双方のインターフェイスの基本的な「アップ」ステータスおよび動作を検証できます。
- ASA の ping：別の ASA のインターフェイスを ping し、そのインターフェイスがアップして応答することを確認できます。
- ASA 経由の ping：ASA の反対側のデバイスを ping することによって、中間 ASA 経由で ping することができます。パケットは、それぞれの方向に移動するときに、2つの中間 ASA のインターフェイスを通過します。このアクションは、中間ユニットのインターフェイス、動作、および応答時間の基本テストになります。
- ネットワーク デバイスの疑わしい動作をテストするための ping：ASA インターフェイスから、正常に機能していないと思われるネットワーク デバイスに ping することができます。インターフェイスが正しく設定されているにもかかわらずエコーが受信されない場合は、デバイスに問題があると考えられます。
- 中間通信をテストするための ping：ASA インターフェイスから、正常に機能することがわかっているネットワーク デバイスに ping することができます。エコーを受信した場合、中間にあるデバイスがすべて正常に動作し、物理的に正しく接続されていることが確認されたこととなります。

## ICMP ping と TCP ping の選択

ASA には、ICMP エコー要求パケットを送信して、エコー応答パケットを受信する従来の ping が付属しています。これは、標準ツールで、すべての仲介ネットワークデバイスで ICMP トラフィックが許可される場合にうまく機能します。ICMP ping を使用して、IPv4/IPv6 アドレスまたはホスト名を ping することができます。

ただし、ICMP を禁止しているネットワークもあります。ご使用のネットワークがこれに該当する場合は、代わりに、TCP ping を使用してネットワーク接続をテストできます。TCP ping では、ping から TCP SYN パケットが送信され、応答で SYN-ACK が受信された段階でその ping が成功したと見なされます。また、TCP ping では、IPv4 アドレスまたはホスト名は ping できますが、IPv6 アドレスは ping できません。

正常な ICMP または TCP ping とは、使用されているアドレスが有効で特定のタイプのトラフィックに反応することを意味しているにすぎません。これは基本接続が機能していることを意味します。デバイス上で動作する他のポリシーで、特定のタイプのトラフィックがデバイスを通過できないようにすることができます。

## ICMP の有効化

デフォルトでは、セキュリティの高いインターフェイスからセキュリティの低いインターフェイスへの ping を実行できます。リターントラフィックを通過させるように ICMP インспекションをイネーブルにすることだけが必要です。セキュリティの低いインターフェイスから高いインターフェイスに ping するには、トラフィックを許可する ACL を適用する必要があります。

ASA インターフェイスを ping する場合は、そのインターフェイスに適用された ICMP ルールによって、エコー要求パケットとエコー応答パケットが許可される必要があります。ICMP ルールは省略可能です。このルールを設定しなかった場合は、インターフェイスへのすべての ICMP トラフィックが許可されます。

この手順では、ASA インターフェイスの ICMP ping をイネーブルにするため、または、ASA 経由の ping 用に構成する必要がある ICMP コンフィギュレーションのすべてについて説明します。

### 手順

**ステップ 1** ICMP ルールでエコー要求/エコー応答が許可されることを確認します。

ICMP ルールは、省略可能で、インターフェイスに直接送信される ICMP パケットに適用されます。ICMP ルールを適用しなかった場合は、すべての ICMP アクセスが許可されます。この場合は、アクションが不要です。

ただし、ICMP ルールを実装する場合は、少なくとも以下の「inside」をご使用のデバイスのインターフェイス名に置き換えたものが各インターフェイスに含まれていることを確認します。

```
ciscoasa(config)# icmp permit 0.0.0.0 0.0.0.0 echo inside
ciscoasa(config)# icmp permit 0.0.0.0 0.0.0.0 echo-reply inside
```

**ステップ 2** アクセスルールで ICMP が許可されることを確認します。

ASA 経由でホストを ping する場合は、アクセスルールで ICMP トラフィックの送受信が許可される必要があります。アクセスルールは、少なくとも、エコー要求/エコー応答 ICMP パケットを許可する必要があります。これらのルールはグローバルルールとして追加することができます。

アクセスルールがインターフェイスに適用されている、または、グローバルに適用されている場合は、次のようなルールを関連 ACL に追加するだけです。

```
ciscoasa(config)# access-list outside_access_in extended permit icmp any
anyecho
ciscoasa(config)# access-list outside_access_in extended permit icmp any
anyecho-reply
```

または、すべての ICMP を許可するだけです。

```
ciscoasa(config)# access-list outside_access_in extended permit icmp any any
```

アクセスルールを使用しない場合は、必要な他のタイプのトラフィックも許可する必要があります。これは、インターフェイスにアクセスルールを適用すると、暗黙の **deny** が追加されるため、他のすべてのトラフィックが破棄されるためです。ACL をインターフェイスに適用する、または、グローバルに適用するには、**access-group** コマンドを使用します。

単にテスト目的でルールを追加する場合は、**access-list** コマンドの **no** 形式を使用して ACL からルールを削除できます。ACL 全体をテストするだけの場合は、**no access-group** コマンドを使用してインターフェイスから ACL を削除します。

### ステップ3 ICMP インспекションをイネーブルにします。

インターフェイスの **ping** とは対照的に、ASA 経由で **ping** する場合は、ICMP インспекションが必要です。インспекションを使用すれば、リターントラフィック（つまり、エコー応答パケット）を **ping** を開始したホストに返すことができるうえ、パケットあたり 1 つの応答の存在が保証されるため、特定のタイプの攻撃を防止することができます。

ICMP インспекションは、デフォルトのグローバルインспекションポリシーでイネーブルにできます。

```
ciscoasa(config)# policy-map global_policy
ciscoasa(config-pmap)# class inspection_default
ciscoasa(config-pmap-c)# inspect icmp
```

## ホストの ping

デバイスを **ping** するには、**ping 10.1.1.1** や **ping www.example.com** のように IP アドレスやホスト名と一緒に **ping** を入力します。TCP ping の場合は、**ping tcp www.example.com 80** のように **tcp** キーワードと宛先ポートを含めます。通常は、実行する必要のあるテストの範囲にします。

成功した **ping** の出力例：

```
Sending 5, 100-byte ICMP Echos to out-pc, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms
```

**ping** が失敗した場合は、失敗した試行が ? で示され、成功率が 100% 未満になります（すべて失敗した場合は 0% になります）。

```
Sending 5, 100-byte ICMP Echos to 10.132.80.101, timeout is 2 seconds:
????
Success rate is 0 percent (0/5)
```

ただし、**ping** の一部の側面を制御するパラメータを追加することもできます。以下に基本オプションを示します。

- ICMP ping。

**ping** [*if\_name*] *host* [ **repeat count**] [ **timeout seconds**] [ **data pattern**] [ **size bytes**] [ **validate**]

それぞれの説明は次のとおりです。

- *if\_name* は、**ping** の送信元 IP アドレスを指定します。ただし、出力インターフェイスは、データルーティングテーブルを使用したルートルックアップによって決定されます。
- *host* は、**ping** するホストの IPv4 アドレス、IPv6 アドレス、またはホスト名です。
- **repeat count** は、送信するパケット数です。デフォルトは 5 分です。
- **timeout seconds** は、応答がなかった場合にタイムアウトするパケットごとの秒数です。デフォルトは 2 です。
- **data pattern** は、送信するパケットに使用される 16 進数のパターンです。デフォルトは 0xabcd です。
- **size bytes** は、送信するパケットの長さです。デフォルト値は 100 バイトです。
- **validate** は、応答データを検証する必要があることを示します。

- TCP ping。

**ping tcp** [*if\_name*] *host* [*port*] [ **repeat count**] [ **timeout seconds**] [ **source host** [*ports*]

それぞれの説明は次のとおりです。

- *if\_name* は、**ping** の送信元 IP アドレスを指定します。ただし、出力インターフェイスは、データルーティングテーブルを使用したルートルックアップによって決定されます。
- *host* は、**ping** する宛先の IPv4 アドレスまたはホスト名です。TCP ping は IPv6 アドレスと一緒に使用できません。
- *port* は、**ping** するホストの TCP ポートです。
- **repeat** と **timeout** は、上記と同じ意味です。
- **source host port** は、**ping** 用の送信元 IP アドレスおよびポートを示します。ランダムポートを取得するには、ポート 0 を使用します。

- インタラクティブ ping。

**ping**

パラメータを指定せずに **ping** を入力した場合は、インターフェイス、宛先、およびキーワードとして使用できない拡張パラメータを含むその他のパラメータが要求されます。**ping** パケットを細かく制御する必要がある場合は、この方式を使用します。

## ASA 接続の体系的なテスト

ASA 接続のさらに体系的なテストを実行する場合は、次の一般的な手順を使用できます。

### 始める前に

手順で説明した syslog メッセージを確認する場合は、ロギングをイネーブルにします (**logging enable** コマンドまたは ASDM の [Configuration] > [Device Management] > [Logging] > [Logging Setup])。

また、必須ではありませんが、ICMP デバッグをイネーブルにして、外部デバイスから ASA インターフェイスを ping したときのメッセージを ASA コンソールに表示することもできます (ASA を通過する ping に関するデバッグ メッセージは表示されません)。ping メッセージとデバッグメッセージをイネーブルにするのはトラブルシューティング中だけにすることをお勧めします。これらのメッセージはパフォーマンスに影響する可能性があります。次に、ICMP デバッグをイネーブルにして、Telnet または SSH セッションに送信する syslog メッセージを設定し、それらをセッションに送信して、ロギングをイネーブルにする例を示します。または、**logging monitor debug** コマンドの代わりに、**logging buffer debug** コマンドを使用してログメッセージをバッファに送信し、後で **show logging** コマンドを使用してそれらを表示することもできます。

```
ciscoasa(config)# debug icmp trace
ciscoasa(config)# logging monitor debug
ciscoasa(config)# terminal monitor
ciscoasa(config)# logging enable
```

この設定では、外部ホスト (209.165.201.2) から ASA の外部インターフェイス (209.165.201.1) への ping が成功すると、次のように表示されます。

```
ciscoasa(config)# debug icmp trace
Inbound ICMP echo reply (len 32 id 1 seq 256) 209.165.201.1 > 209.165.201.2
Outbound ICMP echo request (len 32 id 1 seq 512) 209.165.201.2 > 209.165.201.1
Inbound ICMP echo reply (len 32 id 1 seq 512) 209.165.201.1 > 209.165.201.2
Outbound ICMP echo request (len 32 id 1 seq 768) 209.165.201.2 > 209.165.201.1
Inbound ICMP echo reply (len 32 id 1 seq 768) 209.165.201.1 > 209.165.201.2
Outbound ICMP echo request (len 32 id 1 seq 1024) 209.165.201.2 > 209.165.201.1
Inbound ICMP echo reply (len 32 id 1 seq 1024) 209.165.201.1 > 209.165.201.2
```

この出力では、ICMP パケット長 (32 バイト)、ICMP パケット識別子 (1)、および ICMP シーケンス番号 (ICMP シーケンス番号は 0 から始まり、要求が送信されるたびに増分されます) が示されています。

テストが終了したら、デバッグをディセーブルにします。この設定をそのままにしておくと、パフォーマンスとセキュリティのリスクが高まります。テストのためだけにロギングをイネーブルにした場合は、それもディセーブルにできます。

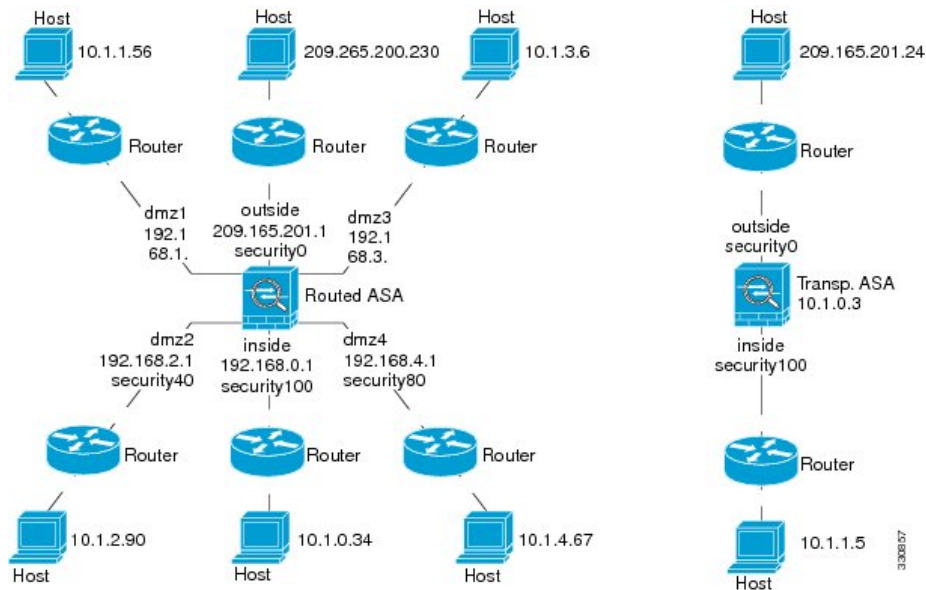
```
ciscoasa(config)# no debug icmp trace
ciscoasa(config)# no logging monitor debug
```

```
ciscoasa(config)# no terminal monitor
ciscoasa(config)# no logging enable
```

手順

**ステップ1** インターフェイス名、セキュリティレベル、およびIPアドレスを示すシングルモードのASAまたはセキュリティコンテキストの図を作成します。図には、直接接続されたすべてのルータ、およびASAをpingするルータの反対側にあるホストも含める必要があります。

図1: インターフェイス、ルータ、およびホストを含むネットワーク図



**ステップ2** 直接接続されたルータから各ASAインターフェイスをpingします。トランスペアレントモードでは、BVI IPアドレスをpingします。このテストでは、ASAインターフェイスがアクティブであること、およびインターフェイスコンフィギュレーションが正しいことを確認します。

ASAインターフェイスがアクティブではない場合、インターフェイスコンフィギュレーションが正しくない場合、またはASAとルータの間でスイッチがダウンしている場合、pingは失敗する可能性があります(次の図を参照)。この場合は、パケットがASAに到達しないので、デバッグメッセージやsyslogメッセージは表示されません。

図2: ASAインターフェイスでのpingの失敗

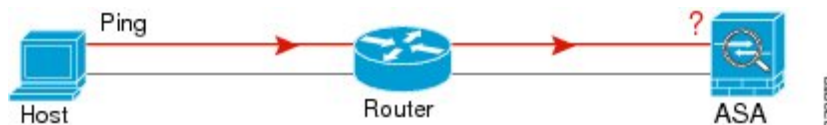
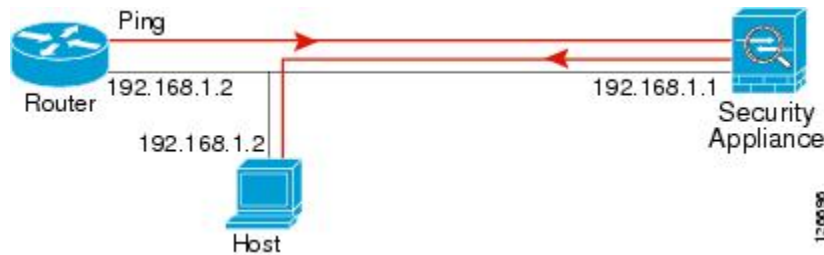


図 3: IP アドレッシングの問題による ping の失敗



ping 応答がルータに戻されない場合は、スイッチ ループまたは冗長 IP アドレスが存在する可能性があります (次の図を参照)。

**ステップ 3** リモート ホストから各 ASA インターフェイスを ping します。トランスペアレント モードでは、BVI IP アドレスを ping します。このテストでは、直接接続されたルータがホストと ASA の間でパケットをルーティングできるかどうか、および ASA がパケットを正確にルーティングしてホストに戻せるかどうかを確認します。

中間ルータを通してホストに戻るルートが ASA がない場合、ping は失敗する可能性があります (次の図を参照)。この場合は、デバッグメッセージは ping が成功したことを示しますが、ルーティングの失敗を示す syslog メッセージ 110001 が表示されます。

図 4: ASA の戻りルート未設定による ping の失敗



**ステップ 4** ASA インターフェイスから既知のネットワーク デバイスへの ping は正しく機能しています。

- ping を受信しない場合は、送信ハードウェアまたはインターフェイスのコンフィギュレーションに問題がある可能性があります。
- ASA のインターフェイスが正しく設定されているにもかかわらず、「既知の正常な」デバイスからエコー応答を受信しない場合は、インターフェイスハードウェアの受信機能に問題があると考えられます。「既知の正常な」受信機能を持つ別のインターフェイスで、同じ「既知の正常な」デバイスに対して ping を送信してエコーを受信できる場合、最初のインターフェイスのハードウェアの受信機能に問題があると確認されたことになります。

**ステップ 5** ホストまたはルータから発信元インターフェイスを介して別のインターフェイス上の別のホストまたはルータに ping します。確認が必要なすべてのインターフェイス ペアに対して、このステップを繰り返します。NAT を使用する場合は、このテストを行うと NAT が正しく動作していることがわかります。

ping が成功すると、ルーテッドモードのアドレス変換 (305009 または 305011) と ICMP 接続が確立されたこと (302020) を確認する syslog メッセージが表示されます。show xlate コマンドまたは show conns コマンドを入力してこの情報を表示することもできます。

NAT が正しく設定されていないことが原因で、ping に失敗することもあります。この場合、NAT が失敗したことを示す syslog メッセージが表示されます (305005 または 305006)。ping

が外部ホストから内部ホストへ送信され、スタティック変換が存在しない場合は、メッセージ 106010 が表示されます。

図 5: ASA のアドレス変換の問題による ping の失敗



## ホストまでのルートの追跡

IP アドレスへのトラフィックの送信で問題が発生している場合は、ホストまでのルートを追跡することによってネットワークパスに問題がないかどうかを確認できます。

### 手順

- ステップ 1 [トレースルート上の ASA の表示 \(27 ページ\)](#) を使用して無効にすることができます。
- ステップ 2 [パケットルートの決定 \(29 ページ\)](#) を使用して無効にすることができます。

## トレースルート上の ASA の表示

デフォルトで、ASA はトレースルート上にホップとして表示されません。これを表示するには、ASA を通過するパケットの存続可能時間を減らして、ICMP 到達不能メッセージのレート制限を増やす必要があります。

### 手順

- ステップ 1 L3/L4 クラスマップを作成して、接続の設定をカスタマイズするトラフィックを識別します。

```
class-map name
match parameter
```

例 :

```
ciscoasa(config)# class-map CONNS
ciscoasa(config-cmap)# match any
```

照合文の詳細については、ファイアウォール設定ガイドのサービスポリシーに関する章を参照してください。

- ステップ 2** クラスマップトラフィックで実行するアクションを設定するポリシーマップを追加または編集して、クラス マップを指定します。

**policy-map** *name* **class** *name*

例 :

```
ciscoasa(config)# policy-map global_policy
ciscoasa(config-pmap)# class CONNS
```

デフォルト設定では、`global_policy` ポリシーマップはすべてのインターフェイスにグローバルに割り当てられます。`global_policy` を編集する場合は、ポリシー名として `global_policy` を入力します。クラス マップの場合、この手順ですでに作成したクラスを指定します。

- ステップ 3** クラスと一致するパケットの存続可能時間 (TTL) を減らします。

**set connection decrement-ttl**

- ステップ 4** 既存のサービス ポリシー (`global_policy` という名前のデフォルト グローバル ポリシーなど) を編集している場合は、このステップを省略できます。それ以外の場合は、1 つまたは複数のインターフェイスでポリシー マップをアクティブにします。

**service-policy** *polycymap\_name* {**global** | **interface** *interface\_name* }

例 :

```
ciscoasa(config)# service-policy global_policy global
```

**global** キーワードはポリシー マップをすべてのインターフェイスに適用し、**interface** はポリシーを1つのインターフェイスに適用します。グローバル ポリシーは1つしか適用できません。インターフェイスのグローバル ポリシーは、そのインターフェイスにサービス ポリシーを適用することで上書きできます。各インターフェイスには、ポリシーマップを1つだけ適用できます。

- ステップ 5** トレース ルートの出力に ASA が表示されるように、ICMP 到達不能メッセージのレート制限を増やします。

**icmp unreachable rate-limit** *rate* **burst-size** *size*

例 :

```
ciscoasa(config)# icmp unreachable rate-limit 50 burst-size 1
```

レート制限は1～100の範囲で設定できます。デフォルトは1です。バーストサイズは動作には影響しませんが、1～10の範囲で設定する必要があります。

例

次の例では、すべてのトラフィックの TTL をグローバルに減らして、ICMP 到達不能制限を 50 に増やします。

```
ciscoasa(config)# class-map global-policy
ciscoasa(config-cmap)# match any
ciscoasa(config-cmap)# exit
ciscoasa(config)# policy-map global_policy
ciscoasa(config-pmap)# class global-policy
ciscoasa(config-pmap-c)# set connection decrement-ttl
ciscoasa(config-pmap-c)# exit
ciscoasa(config)# icmp unreachable rate-limit 50 burst-size 6
```

## パケットルートの決定

traceroute を使用すれば、パケットが宛先に到着するまでのルートを特定できます。traceroute は、無効なポート上の宛先に UDP パケットまたは ICMPv6 エコーを送信することで機能します。ポートが有効でないため、宛先への途中にあるルータは ICMP または ICMPv6 Time Exceeded Message で応答し、そのエラーを ASA に報告します。

traceroute は送信された各プローブの結果を表示します。出力の各行が 1 つの TTL 値に対応します (昇順)。次の表に、出力記号の説明を示します。

出力記号	説明
*	タイムアウトの期間内にプローブへの応答を受信しませんでした。
U	宛先へのルートが存在しません。
nn msec	各ノードで、指定した数のプローブのラウンドトリップにかかる時間 (ミリ秒)。
!N.	ICMP ネットワークに到達できません。ICMPv6 では、アドレスは対象外です。
!H	ICMP ホストに到達できません。
!P	ICMP に到達できません。ICMPv6 では、ポートが到達不能です。
!A	ICMP が管理者によって禁止されています。
?	原因不明の ICMP エラーが発生しました。

手順

宛先までのルートを追跡します。

**traceroute** [*destination\_ip* | *hostname*] [**source** {*source\_ip* | *source-interface*}] [**numeric**] [**timeout** *timeout\_value*] [**probe** *probe\_num*] [**ttl** *min\_ttl* *max\_ttl*] [**port** *port\_value*] [**use-icmp**]

例 :

```
ciscoasa# traceroute 209.165.200.225

Type escape sequence to abort.
Tracing the route to 209.165.200.225

 1 10.83.194.1 0 msec 10 msec 0 msec
 2 10.83.193.65 0 msec 0 msec 0 msec
 3 10.88.193.101 0 msec 10 msec 0 msec
 4 10.88.193.97 0 msec 0 msec 10 msec
 5 10.88.239.9 0 msec 10 msec 0 msec
 6 10.88.238.65 10 msec 10 msec 0 msec
 7 172.16.7.221 70 msec 70 msec 80 msec
 8 209.165.200.225 70 msec 70 msec 70 msec

ciscoasa# traceroute 2002::130

Type escape sequence to abort.
Tracing the route to 2002::130

 1 5000::2 0 msec 0 msec 0 msec
 2 2002::130 10 msec 0 msec 0 msec
```

通常は、宛先 IP アドレスまたはホスト名を含める (**traceroute www.example.com** など) だけです。ただし、必要に応じて、トレースの特性を調整できます。

- **source** {*source\_ip*|*source-interface*} : トレースの送信元として使用するインターフェイスを指定します。インターフェイスは、名前または IP アドレスで指定できます。IPv6 では、送信元インターフェイスを指定できません。送信元 IP アドレスだけを指定できます。IPv6 アドレスは、ASA インターフェイスで IPv6 を有効にしている場合にのみ有効です。トランスペアレントモードでは、管理アドレスを使用する必要があります。
- **numeric** : IP アドレスのみをトレースルートに表示するように指示します。このキーワードを指定しなかった場合は、DNS が設定されていれば、トレースルートでアドレスの DNS 参照が実行され、DNS 名が追加されます。
- **timeout** *timeout\_value* : タイムアウトするまで応答を待機する時間。デフォルトは 3 秒です。
- **probe** *probe\_num* : 各 TTL レベルで送信するプローブの数。デフォルトは 3 です。
- **ttl** *min\_ttl* *max\_ttl* : プローブの最小および最大存続可能時間。デフォルトの最小値は 1 ですが、この値を増やして、既知のホップの表示を抑制することができます。デフォルトの最大値は 30 です。トレースルートは、パケットが宛先に到達するか、または最大値に達すると終了します。
- **port** *port\_value* : 使用する UDP ポート。デフォルトは 33434 です。
- **use-icmp** : プローブの UDP パケットの代わりに ICMP パケットを送信します。

## パケットトレーサを使用したポリシー設定のテスト

送信元と宛先のアドレスおよびプロトコルの特性に基づいてパケットをモデル化することによってポリシー設定をテストできます。トレースは、ポリシー参照を実行してアクセスルールや NAT などをテストし、パケットを許可するか、拒否するかを確認します。

このようにパケットをテストすることによって、ポリシーの結果を確認し、必要に応じて、許可または拒否するトラフィックのタイプが処理されるかどうかをテストできます。設定の確認に加えて、トレーサを使用して許可すべきパケットが拒否されるなどの予期せぬ動作をデバッグできます。

### 手順

**ステップ 1** このコマンドは複雑なため、複数の部分に分けて説明します。トレース用のインターフェイスとプロトコルを選択することから始めます。

```
packet-tracer input ifc_name [vlan-id vlan_id] {icmp | tcp | udp | rawip | sctp} [ inline-tag tag] ...
```

それぞれの説明は次のとおりです。

- **input ifc\_name** : トレースを開始するインターフェイスの名前。ブリッジグループの場合、ブリッジグループメンバーインターフェイスの名前を指定します。
- **vlan-id vlan\_id** : (オプション)。パケットトレーサが（あとでサブインターフェイスにリダイレクトされる）親インターフェイスに入る仮想 LAN。VLAN ID は、入力インターフェイスがサブインターフェイスでない場合にのみ使用可能です。有効な値の範囲は 1 ~ 4096 です。
- **icmp**、**tcp**、**udp**、**rawip**、**sctp** : 使用するプロトコル。「rawip」は未加工の IP、つまり、TCP/UDP 以外の IP パケットです。
- **inline-tag tag** : (オプション)。レイヤ 2 CMD ヘッダーに埋め込まれたセキュリティグループタグの値。有効な値の範囲は 0 ~ 65533 です。

**ステップ 2** 次に、送信元アドレスとプロトコル基準を入力します。

```
...{src_ip | user username | security-group {name name | tag tag} | fqdn fqdn-string}...
```

それぞれの説明は次のとおりです。

- **src\_ip** : パケットトレーサ用の送信元 IPv4 または IPv6 アドレス。
- **user username** : domain\user の形式のユーザー ID。ユーザーに対して最後にマッピングされたアドレス（複数ある場合）がトレースに使用されます。
- **security-group {name name | tag tag}** : TrustSec の IP-SGT 参照に基づく送信元セキュリティグループ。セキュリティグループの名前またはタグ番号を指定できます。
- **fqdn fqdn-string** : 送信元ホストの完全修飾ドメイン名、IPv4 のみ。

**ステップ3** 次に、プロトコルの特性を入力します。

- [ICMP] : ICMP タイプ (1 ~ 255) 、 ICMP コード (0 ~ 255) 、 およびオプションで ICMP 識別子を入力します。各変数に対応する数字 (エコーに対応する 8 など) を使用する必要があります。

*type code... [ident]...*

- TCP/UDP/SCTP : 送信元ポート番号を入力します。

*...src\_port ...*

- [Raw IP] : プロトコル番号 (0 ~ 255) を入力します。

*...protocol ...*

**ステップ4** 最後に、宛先アドレス基準、TCP/UDP トレース用の宛先ポート、およびオプションのキーワードを入力して、**Enter** キーを押します。

*...dmac {dst\_ip | security-group { name name | tag tag } | fqdn fqdn-string} dst\_port [detailed] [xml]*

それぞれの説明は次のとおりです。

- *dst\_ip* : パケット トレース用の宛先 IPv4 または IPv6 アドレス。
- **security-group {name name | tag tag}** : TrustSec の IP-SGT 参照に基づく宛先セキュリティグループ。セキュリティグループの名前またはタグ番号を指定できます。
- **fqdn fqdn-string** : 宛先ホストの完全修飾ドメイン名、IPv4 のみ。
- *dst\_port* : TCP/UDP/SCTP トレース用の宛先ポート。ICMP または未加工 IP トレースの場合はこの値を含めないでください。
- *dmac* : (トランスペアレント モード) 宛先 MAC アドレス。
- **detailed** : 標準出力に加えて詳細なトレース結果情報を提供します。
- **xml** : トレース結果を XML 形式で表示します。

**ステップ5** クラスタユニット全体でパケットをデバッグするには、パケットトレーサの **persist** オプションを入力します。

- **transmit** オプションを使用すると、シミュレートされたパケットが ASA から出られるようにすることができます。
- ACL、VPN フィルタ、IPsec スプーフィング、uRPF などのセキュリティチェックをスキップするには、**bypass-checks** オプションを使用します。
- **decrypted** オプションを使用すると、復号化されたパケットを VPN トンネルに注入し、さらに、VPN トンネルを経由して到着するパケットをシミュレートすることもできます。

**ステップ6** 特定の packets をクラスタユニットで追跡するには、**id** と **origin** を入力します。

- **id** : トレースを開始するユニットによって割り当てられた識別番号。

- **origin** : トレースを開始するクラスタ ユニットを示します。

## 例

次に、HTTP ポート 10.100.10.10 から 10.100.11.11 への TCP パケットをトレースする例を示します。暗黙の拒否アクセスルールによってパケットがドロップされることを示す結果が表示されます。

```
ciscoasa(config)# packet-tracer input outside tcp 10.100.10.10 80
10.100.11.11 80
```

```
Phase: 1
Type: ROUTE-LOOKUP
Subtype: Resolve Egress Interface
Result: ALLOW
Config:
Additional Information:
found next-hop 10.86.116.1 using egress ifc  outside

Phase: 2
Type: ACCESS-LIST
Subtype:
Result: DROP
Config:
Implicit Rule
Additional Information:

Result:
input-interface: outside
input-status: up
input-line-status: up
output-interface: NP Identity Ifc
output-status: up
output-line-status: up
Action: drop
Drop-reason: (acl-drop) Flow is denied by configured rule
```

## 接続のモニタリング

送信元、宛先、プロトコルなどに関する情報を含む現在の接続を表示するには、**show conn all detail** コマンドを使用します。

## ASA Virtual での同意トークンを使用したルートシェルアクセスのサポート

ASA Virtual は、承認ユーザーが管理者パスワードなしでトラブルシューティングまたは診断の目的で Linux ルートシェルにワンタイムアクセスできるようにする新しい同意トークンメカニズムをサポートします。



- (注)
- ユーザーは、コンソールのシリアル端末またはホストの仮想シリアルコンソール経由で、チャレンジを開始し、応答を送信できます。応答が正常に検証されると、ルートシェルが対応するコンソールで開かれます。
  - ユーザーは、SSH または Telnet 端末でチャレンジを生成し、応答を受け入れることができます。応答が確認されると、ルートシェルはコンソール端末でのみ使用可能になり、SSH や Telnet 端末では使用できなくなります。
  - ルートシェルにアクセスするための推奨される方法は、仮想シリアルコンソール経由です。このコンソールでは、ユーザーがチャレンジを生成し、応答を検証し、シリアル端末内で直接ルートシェルを開くことができます。

### Consent Token を使用して Linux ルートシェルにアクセスする手順

次の手順を実行して、チャレンジを生成し、Cisco TAC からの応答を取得し、ルートシェルを開きます。

#### 1. チャレンジを生成する

Cisco ASA Virtual で次の CLI コマンドを実行するとチャレンジ文字列を生成できます。

```
ciscoasa# consent-token generate-challenge shell-access 1
```

出力例：

```
Generating Challenge.....
```

```
Challenge String (Please copy everything between the asterisk lines exclusively):
```

```
*****
wzC0AAQ0FPAQAAEFgAEAAAAAACITpkrFpSLOEBAQzOyEFWlp7wGirO8-SUQUABAAAJGARQMBwPHMFTQVDAgABEFTQXJAs5QV5SzRNOERg=
*****
```

#### 2. このチャレンジを Cisco TAC と共有します。

生成されたチャレンジ文字列を Cisco TAC チームに送信します。TAC エンジニアは、要求と承認を確認した後、対応する承認応答を提供します。

#### 3. 応答を受け入れる

Cisco TAC から受け入れ応答を受信したら、次のコマンドを使用して適用します。

```
ciscoasa# consent-token accept-response shell-access
```

プロンプトが表示されたら、制限時間内で受け入れ応答を貼り付けます。

```
Please input the response within 37 seconds when you are ready .....
/CYANQAAAQYBAAQAAAABAgAEAAAAAAMBYkFrdDkydBUMXlCMStNSWNlMElteUJ6VzhRMVViv05NSX$
Response Signature Verified successfully !
```

検証が成功すると、ルートシェルがコンソール端末で開きます。



**重要** 安全でシームレスな運用のために、すべての Consent Token 交換が Cisco TAC チームと調整され、実行されていることを確認します。

## テストおよびトラブルシューティングの履歴

機能名	プラットフォームリリース	説明
tracerouteのIPv6サポート	9.7(1)	<b>traceroute</b> コマンドが変更され、IPv6 アドレスも受け入れられるようになりました。  次のコマンドが変更されました。 <b>traceroute</b>
ブリッジグループメンバーインターフェイス用のパケットトレーサのサポート	9.7(1)	ブリッジグループメンバーインターフェイスにパケットトレーサを使用できるようになりました。  <b>packet-tracer</b> コマンドに次の2つのオプションが追加されました。 <b>vlan-id</b> および <b>dmac</b>
手動によるパケットキャプチャの開始と停止	9.7(1)	キャプチャを手動で停止および開始できるようになりました。  追加/変更されたコマンド： <b>capture stop</b>

機能名	プラットフォームリリース	説明
強化されたパケットトレーサおよびパケットキャプチャ機能	9.9(1)	<p>パケット トレーサは次の機能で強化されました。</p> <ul style="list-style-type: none"> <li>• パケットがクラスタ ユニット間を通過するときにパケットを追跡します。</li> <li>• シミュレートされたパケットが ASA から出られるようにします。</li> <li>• シミュレートされたパケットのセキュリティ チェックをバイパスします。</li> <li>• シミュレートされたパケットを IPsec/SSL で復号化されたパケットとして扱います。</li> </ul> <p>パケット キャプチャは次の機能で強化されました。</p> <ul style="list-style-type: none"> <li>• パケットを復号化した後にキャプチャします。</li> <li>• トレースをキャプチャし、永続リストに保持します。</li> </ul> <p>新規または変更されたコマンド：<b>cluster exec capture test trace include-decryptd、cluster exec capture test trace persist、cluster exec clear packet-tracer、cluster exec show packet-tracer id、cluster exec show packet-tracer origin、packet-tracer persist、packet-tracer transmit、packet-tracer decryptd、packet-tracer bypass-checks</b></p>

機能名	プラットフォームリリース	説明
ACL を使用せず IPv6 トラフィックを一致させるためのパケットキャプチャのサポート	9.10(1)	<p><b>capture</b> コマンドの <b>match</b> キーワードを使用する場合、<b>any</b> キーワードは IPv4 トラフィックのみ照合します。IPv4 または IPv6 トラフィックをキャプチャするために、<b>any4</b> と <b>any6</b> キーワードを指定できるようになりました。<b>any</b> キーワードでは、引き続き IPv4 トラフィックのみ照合されます。</p> <p>新規/変更されたコマンド：<b>capture match</b></p>
Forepower 9300/4100 の新しい <b>debug telemetry</b> コマンド	9.14(1)	<p><b>debug telemetry</b> コマンドを使用すると、テレメトリに関連するデバッグメッセージが表示されます。このデバッグは、テレメトリレポートの生成時にエラーの原因を特定するために役立ちます。</p> <p>新規/変更されたコマンド：<b>[ no ] debug telemetry</b>、<b>show debug telemetry</b></p>
<b>ping</b> コマンドの変更	9.18(2)	<p>ループバック インターフェイスの <b>ping</b> をサポートするために、<b>ping</b> コマンドの動作が変更されました。コマンドでインターフェイスを指定する場合、送信元 IP アドレスは指定されたインターフェイスの IP アドレスと一致しますが、実際の出カインターフェイスは、データルーティングテーブルを使用したルートルックアップによって決定されます。</p> <p>新規/変更されたコマンド：<b>ping</b></p>
スイッチのパケットキャプチャ	9.20(1)	<p>スイッチの出力および入力トラフィックパケットをキャプチャするように設定できるようになりました。このオプションは、Secure Firewall 4200 モデルデバイスに対してのみ使用できます。</p> <p>新しい/変更されたコマンド：</p> <p><b>capture capture_name switch interface interface_name [ direction { both   egress   ingress } ]</b></p>

機能名	プラットフォームリリース	説明
ASA Virtual での同意トークンを使用したルートシェルアクセスのサポート	9.24(1)	ASA Virtual は、承認ユーザーが管理者パスワードなしでトラブルシューティングまたは診断の目的で Linux ルートシェルにワンタイムアクセスできるようにする新しい同意トークンメカニズムをサポートします。  新規/変更されたコマンド： <b>consent-token generate-challenge shell-access</b> 、 <b>consent-token accept-response shell-access</b>

## 翻訳について

このドキュメントは、米国シスコ発行ドキュメントの参考和訳です。リンク情報につきましては、日本語版掲載時点で、英語版にアップデートがあり、リンク先のページが移動/変更されている場合がありますことをご了承ください。あくまでも参考和訳となりますので、正式な内容については米国サイトのドキュメントを参照ください。

## 翻訳について

このドキュメントは、米国シスコ発行ドキュメントの参考和訳です。リンク情報につきましては、日本語版掲載時点で、英語版にアップデートがあり、リンク先のページが移動/変更されている場合がありますことをご了承ください。あくまでも参考和訳となりますので、正式な内容については米国サイトのドキュメントを参照ください。