CHAPTER 30

IPSec VPN SPA のモニタリングおよびアカウンティングの設定

この章では、Cisco 7600 シリーズ ルータ上の IPSec VPN SPA を使用してモニタリングおよびアカウンティングを設定する方法について説明します。具体的な内容は次のとおりです。

（注） Cisco IOS の IP Security（IPSec）暗号化処理およびポリシーについての詳細は、『*Cisco IOS Security Configuration Guide*』および『*Cisco IOS Security Command Reference*』を参照してください。

システム イメージおよびコンフィギュレーション ファイルの管理については、『*Cisco IOS Configuration Fundamentals Configuration Guide*』および『*Cisco IOS Configuration Fundamentals Command Reference*』を参照してください。

この章で使用するコマンドの詳細については、『*Cisco IOS Software Releases 12.2SR Command Reference*s』および『*Cisco IOS Software Releases 12.2SX Command References*』を参照してください。また、関連する CiscoIOS Release12.2 ソフトウェア コマンド リファレンスおよびマスター インデックスも参照してください。詳細については、「関連資料」（p.lv）を参照してください。

ヒント IPSec VPN SPA を使用して Virtual Private Network（VPN; バーチャル プライベート ネットワーク）を正しく設定するために、設定の概要および注意事項にすべて目を通してから設定作業を始めてください。

IPSec VPN SPA のモニタリングおよびアカウンティングの概要

この章では、IPSec VPN をモニタおよび管理するために使用できる IPSec 機能の一部について説明します。次のような機能があります。

- IPSec VPN モニタリング機能 — VPN のトラブルシューティングおよびエンドユーザ インターフェイスの監視に使用できる VPN セッションのモニタリング拡張機能が提供されます。
- IPSec VPN アカウンティング機能 — セッションが開始および停止された時間を示すセッション アカウンティング レコードが生成されます。
- Cisco VRF 対応 IPSec の IPSec および Internet Key Exchange（IKE; インターネット キー エクスチェンジ）MIB（管理情報ベース）サポート機能 — MIB を使用して VPN Routing and Forwarding（VRF; VPN ルーティングおよび転送）対応 IPSec を管理できるようにします。

IPSec VPN セッションのモニタリングおよび管理

IPSec VPN モニタリング機能により、VPN のトラブルシューティングおよびエンドユーザ インターフェイスの監視に使用できる VPN セッションのモニタリング拡張機能が提供されます。暗号セッションは、2 つの暗号化エンドポイント間の IPSec 接続（フロー）の集合です。2 つの暗号化エンドポイントがキーイング プロトコルとして IKE を使用している場合、これらのエンドポイントは相互に IKE ピアとして機能します。暗号セッションは通常、1 つの IKE SA（セキュリティ アソシエーション）（制御トラフィック用）および最低 2 つの IPSec SA（データ トラフィック用、各方向に 1 つずつ）で構成されます。キーの再生成中、または両端から同時にセットアップ要求が発行された場合、IKE SA および IPSec SA の重複や、同じセッションの IKE SA または IPSec SA の重複が発生することがあります。

セッションのモニタリング拡張機能は次のとおりです。

- コンフィギュレーション ファイルの IKE ピアに関する記述を指定する機能
- 暗号化セッション ステータスのサマリー リスト
- 暗号化セッションのアップまたはダウン ステータスの Syslog 通知
- 共通の CLI（コマンドライン インターフェイス）を使用して IKE SA と IPSec SA の両方を削除する機能

IPSec VPN セッションのモニタリングおよび管理の設定時の注意事項および制約事項

IPSec VPN モニタリングを設定する場合は、次の注意事項および制約事項に従ってください。

- ルータ上で次のいずれかの暗号イメージを実行している必要があります。
  - s72033-adventerprisek9_wan-mz (Supervisor Engine 720)
  - s72033-advipservicesk9_wan-mz (Supervisor Engine 720)
  - s3223-adventerprisek9_wan-mz (Supervisor Engine 32)
  - s3223-advipservicesk9_wan-mz (Supervisor Engine 32)

IKE ピアの記述の追加

IPSec VPN セッションに IKE ピアの記述を追加するには、グローバル コンフィギュレーション モードから次の作業を行います。

	コマンド	説明
ステップ 1	`Router(config)# crypto isakmp peer {ip-address ip-address}`	IPSec ピアによるアグレッシブ モードのトンネル アトリビュートに関する Authentication, Authorization, Accounting（AAA; 認証、認可、アカウンティング）の IKE クエリーをイネーブルにし、ISAKMP ピア コンフィギュレーション モードを開始します。 • *ip-address* — ピアの IP アドレス
ステップ 2	`Router(config-isakmp-peer)# description description`	IKE ピアに関する記述を追加します。 • *description* — ピアを特定する記述

ピアの記述の確認

ピアの記述を確認するには、**show crypto isakmp peer** コマンドを入力します。

```
Router# show crypto isakmp peer

Peer: 10.2.2.9 Port: 500
Description: connection from site A
flags: PEER_POLICY
```

アドレス 10.2.2.9 のピアが接続し、セッションがアップ状態になると、Syslog ステータスが次のように表示されます。

```
%CRYPTO-5-SESSION_STATUS: Crypto tunnel is UP. Peer 10.2.2.9:500 Description:
connection from site A Id: ezvpn
```

暗号化セッション ステータスのサマリー リストの表示

すべてのアクティブな VPN セッションの一覧を表示するには、**show crypto session** コマンドを入力します。次の内容が表示されます。

- インターフェイス
- IKE ピアの記述（該当する場合）
- IPSec SA を作成したピアに対応付けられている IKE SA
- セッションのフローを処理している IPSec SA

同じピアに対して、複数の IKE または IPSec SA が確立される場合があります。その場合、ピアに対応付けられた IKE SA や、セッションのフローを処理している IPSec SA ごとに異なる値を使用して、IKE ピアの記述が繰り返されます。

このコマンドのバリエーションである **show crypto session detail** を使用して、セッションに関して、より詳しい情報を取得することもできます。

次に、**detail** キーワードを使用しない **show crypto session** コマンドの出力例を示します。

```
Router# show crypto session

Crypto session current status

Interface: FastEthernet0/1
Session status: UP-ACTIVE
Peer: 172.0.0.2/500
IKE SA: local 172.0.0.1/500 remote 172.0.0.2/500 Active
IPSEC FLOW: permit ip 10.10.10.0/255.255.255.0 10.30.30.0/255.255.255.0
Active SAs: 2, origin: crypto map
```

次に、**detail** キーワードを使用した ***show crypto session*** コマンドの出力例を示します。

```
Router# show crypto session detail

Interface: Tunnel0
Session status: UP-ACTIVE
Peer: 10.1.1.3 port 500 fvrf: (none) ivrf: (none)
Desc: this is my peer at 10.1.1.3:500 Green
Phase1_id: 10.1.1.3
IKE SA: local 10.1.1.4/500 remote 10.1.1.3/500 Active
Capabilities:(none) connid:3 lifetime:22:03:24
IPSEC FLOW: permit 47 host 10.1.1.4 host 10.1.1.3
Active SAs: 0, origin: crypto map
Inbound: #pkts dec'ed 0 drop 0 life (KB/Sec) 0/0
Outbound: #pkts enc'ed 0 drop 0 life (KB/Sec) 0/0
IPSEC FLOW: permit ip host 10.1.1.4 host 10.1.1.3
Active SAs: 4, origin: crypto map
Inbound: #pkts dec'ed 4 drop 0 life (KB/Sec) 4605665/2949
Outbound: #pkts enc'ed 4 drop 1 life (KB/Sec) 4605665/2949
```

暗号化セッションのアップまたはダウン ステータスに関する Syslog 通知

暗号化セッションのアップまたはダウン ステータスに関する Syslog 通知機能により、暗号化セッションがアップまたはダウンになるたびに、Syslog に対する通知が行われます。セッション ステータスの Syslog ロギングをイネーブルにするには、コンフィギュレーション モードで **crypto logging session** および **crypto logging ezvpn** コマンドを入力します。

次に、暗号化セッションがアップ状態になったときの Syslog 通知の例を示します。

```
%CRYPTO-5-SESSION_STATUS: Crypto session is UP. Peer 10.6.6.1:500 fvrf=name10
ivrf=name20 Description: SJC24-2-VPN-Gateway Id: 10.5.5.2
```

次に、暗号化セッションがダウン状態になったときの Syslog 通知の例を示します。

```
%CRYPTO-5-SESSION_STATUS: Crypto session is DOWN. Peer 10.6.6.1:500 fvrf=name10
ivrf=name20 Description: SJC24-2-VPN-Gateway Id: 10.5.5.2
```

暗号化セッションのクリア

Cisco IOS の以前のソフトウェア リリースでは、IKE SA および IPSec SA の両方を 1 つのコマンドでクリアすることはできませんでした。その代わりに、**clear crypto isakmp** コマンドを入力して IKE を、**clear crypto ipsec** コマンドを使用して IPSec をクリアする必要がありました。**clear crypto session** コマンドを使用すると、1 つのコマンドで IKE と IPSec の両方をクリアできます。特定の暗号化セッションまたはすべてのセッションのサブセット（特定のリモート サイトへの 1 つのトンネルなど）をクリアするには、セッション固有のパラメータ（ローカルまたはリモートの IP アドレス、ローカルまたはリモートのポート、Front door VRF [FVRF] 名、Inside VRF [IVRF] 名など）を指定する必要があります。通常、削除すべきトンネル 1 つを指定するには、リモート IP アドレスを使用します。

clear crypto session コマンドを入力するとき、パラメータとしてローカル IP アドレスを指定すると、その IP アドレスをローカルの暗号化エンドポイント（IKE ローカル アドレス）として共有するすべてのセッション（および各セッションの IKE SA と IPSec SA）がクリアされます。**clear crypto session** コマンドを入力するときパラメータを指定しないと、ルータ上のすべての IPSec SA および IKE SA がクリアされます。

暗号化セッションをクリアするには、特権 EXEC モードで、ルータのコマンドラインから **clear crypto session** コマンドを入力します。このコマンドを使用する場合、コンフィギュレーション ファイル内のコンフィギュレーション ステートメントは必要ありません。

```
Router# clear crypto session
```

IPSec VPN モニタリングに関する詳しい設定情報は、次の URL を参照してください。

http://www.cisco.com/univercd/cc/td/doc/product/software/ios123/123newft/123t/123t_4/gt_ipsvm.htm

IPSec VPN モニタリングの設定例は、「Cisco VRF 対応 IPSec の IPSec および IKE MIB サポートの設定例」（p.30-14）を参照してください。

IPSec VPN アカウンティングの設定

IPSec VPN アカウンティング機能により、セッションが開始および停止された時間を示すセッション アカウンティング レコードが生成されます。

VPN セッションは、IKE SA およびその IKE SA によって作成された 1 つまたは複数の SA ペアとして定義されます。セッションは最初の IPSec ペアが作成された時点で開始し、すべての IPSec SA が削除された時点で停止します。IPSec アカウンティングを設定した場合、IKE フェーズの完了後にセッションのアカウンティング開始レコードが生成されます。キーの再生成を行っても、新しいアカウンティング レコードは生成されません。

セッション識別情報およびセッション使用情報が、標準的な Remote Authentication Dial-In User Service（RADIUS）アトリビュートおよびベンダー固有のアトリビュート（VSA）を使用して、RADIUS サーバに渡されます。

IPSec VPN アカウンティングをイネーブルにするには、グローバル コンフィギュレーション モードから次の作業を行います。

	コマンド	説明
ステップ 1	`Router(config)# aaa new-model`	アカウンティング サーバへの暫定アカウンティング レコードの定期的な送信をイネーブルにします。
ステップ 2	`Router(config)# aaa authentication login` *list-name* ***group radius***	ログイン時の RADIUS サーバによる AAA 認証を設定します。 • *list-name* — ユーザがログインした時点でアクティブにされる認証方式のリスト名として使用するストリング • ***group radius*** — すべての RADIUS サーバのリストを認証に使用します。
ステップ 3	`Router(config)# aaa authorization network` *list-name* ***group radius***	Serial Line Internet Protocol（SLIP; シリアル ライン インターネット プロトコル）、PPP（ポイントツーポイント プロトコル）、PPP Network Control Program（NCP; ネットワーク コントロール プログラム）、AppleTalk Remote Access（ARA）など、すべてのネットワーク関連サービス要求について許可を実行します。 • *list-name* — ユーザがログインした時点でアクティブにされる許可方式のリスト名として使用するストリング • ***group radius*** — すべての RADIUS サーバのリストを認証に使用します。

	コマンド	説明
ステップ 4	`Router(config)# aaa accounting network` *list-name* ***start-stop*** [***broadcast***] ***group radius***	RADIUS を使用している場合、課金またはセキュリティの目的で、要求されたネットワーク関連サービスの AAA アカウンティングをイネーブルにします。 • *list-name* — アカウンティング方式のリスト名として使用するストリング • ***start-stop*** — プロセスの始まりにアカウンティング「開始」通知を、プロセスの終わりにアカウンティング「停止」通知を送信します。アカウンティング「開始」レコードがバックグラウンドで送信されます。アカウンティング サーバがアカウンティング「開始」通知を受信したかどうかにかかわらず、要求されたユーザ プロセスが開始されます。 • ***broadcast*** — （任意）複数の AAA サーバへのアカウンティング レコードの送信をイネーブルにします。同時に、各グループの最初のサーバにアカウンティング レコードを送信します。最初のサーバが使用不可能な場合、そのグループで定義されているバックアップ サーバへのフェールオーバーが行われます。 • ***group radius*** — **aaa group server radius** コマンドで定義されたすべての RADIUS サーバのリストを認証に使用します。
ステップ 5	`Router(config)# aaa session-id common`	コール内の各 AAA アカウンティング サービス タイプに同じセッション ID を使用するか、それともアカウンティング サービス タイプごとに異なるセッション ID を割り当てるかを指定します。 • ***common*** — 対象となるコールに関して送信されるすべてのセッション ID 情報が同じになるようにします。デフォルトの動作は common です。
ステップ 6	`Router(config)# crypto isakmp profile` *profile-name*	IPSec ユーザ セッションを監査し、ISAKMP プロファイル コンフィギュレーション モードを開始します。 • *profile-name* — ユーザ プロファイルの名前。ユーザ プロファイルに RADIUS サーバを対応付けるには、ユーザ プロファイル名を指定する必要があります。
ステップ 7	`Router(conf-isa-prof)# vrf` *ivrf*	VRF インスタンス名にオンデマンドのアドレス プールを対応付けます。 • *ivrf* — IPSec トンネルをマッピングする VRF
ステップ 8	`Router(conf-isa-prof)# match identity group` *group-name*	ピアからのアイデンティティを ISAKMP プロファイルと照合します。 • *group-name* — 識別子（ID）タイプ ID_KEY_ID と一致する Unity グループ。Unity およびメイン モードの Rivest, Shamir, and Adelman（RSA）シグニチャを使用する場合、*group-name* 引数は DN（認定者名）の Organizational Unit（OU）フィールドと一致します。

	コマンド	説明
ステップ 9	`Router(conf-isa-prof)# client authentication list list-name`	ISAKMP プロファイルに IKE 拡張認証（XAUTH）を設定します。 • *list-name* — ユーザがログインした時点でアクティブにされる認証方式のリスト名として使用するストリング。このリスト名は、AAA の設定時に定義したリスト名と同じである必要があります。
ステップ 10	`Router(conf-isa-prof)# isakmp authorization list list-name`	ISAKMP プロファイル内の AAA サーバを使用し、IKE 共有秘密パラメータおよびその他のパラメータを設定します。共有秘密などのパラメータは一般に、モード コンフィギュレーション（MODECFG）によってリモート ピアにプッシュされます。 • *list-name* — コンフィギュレーション モード アトリビュートまたはアグレッシブ モードの事前共有キーとして使用される AAA 許可リスト
ステップ 11	`Router(conf-isa-prof)# client configuration address [initiate | respond]`	ISAKMP プロファイルで IKE モード コンフィギュレーション（MODECFG）を設定します。 • ***initiate*** — ルータは各ピアの IP アドレスの設定を試みます。 • ***respond*** — ルータは任意の要求元ピアからの IP アドレスの要求を受け付けます。
ステップ 12	`Router(conf-isa-prof)# accounting list-name`	この Internet Security Association and Key Management Protocol（ISAKMP）プロファイルを使用して、接続するすべてのピアについて AAA アカウンティング サービスをイネーブルにします。 • *list-name* — クライアント アカウンティング リストの名前
ステップ 13	`Router(conf-isa-prof)# exit`	ISAKMP プロファイル コンフィギュレーション モードを終了し、グローバル コンフィギュレーション モードに戻ります。
ステップ 14	`Router(config)# crypto dynamic-map dynamic-map-name dynamic-seq-num`	ダイナミック クリプト マップ テンプレートを作成し、暗号マップ コンフィギュレーション コマンド モードを開始します。 • *dynamic-map-name* — ポリシー テンプレートとして使用するダイナミック クリプト マップ セットの名前 • *dynamic-seq-num* — ダイナミック クリプト マップ エントリに割り当てるシーケンス番号
ステップ 15	`Router(config-crypto-map)# set transform-set transform-set-name`	暗号マップ テンプレートとともに使用できるトランスフォーム セットを指定します。トランスフォーム セットは、IPSec セキュリティ プロトコルおよびアルゴリズムを定義します。トランスフォーム セットおよび使用できる値については、『*Cisco IOS Security Command Reference*』に記載されています。 • *transform-set-name* — トランスフォーム セットの名前

	コマンド	説明
ステップ 16	`Router(config-crypto-map)# set isakmp-profile profile-name`	ISAKMP プロファイル名を設定します。 • *profile-name* — ISAKMP プロファイルの名前
ステップ 17	`Router(config-crypto-map)# reverse-route [remote-peer]`	VPN リモート トンネルのエンドポイントの後ろに、宛先へのルート (IP アドレス) を挿入できるようにします。トンネル エンドポイント自体へのルートも含めることができます（暗号マップの **remote-peer** キーワードを使用)。 • ***remote-peer*** — パブリック IP アドレスおよび IPSec トンネルの宛先アドレスのルートが、ルーティング テーブルに挿入されます。
ステップ 18	`Router(config-crypto-map)# exit`	暗号マップ コンフィギュレーション モードを終了し、グローバル コンフィギュレーション モードに戻ります。
ステップ 19	`Router(config)# crypto map map-name ipsec-isakmp dynamic dynamic-map-name`	ダイナミックに作成される暗号マップ設定のテンプレートとなる暗号プロファイルを作成します。 • *map-name* — 暗号マップ セットの識別名 • *dynamic-map-name* — ポリシー テンプレートとして使用するダイナミック クリプト マップ セットの名前
ステップ 20	`Router(config)# radius-server host ip-address [auth-port auth-port-number] [acct-port acct-port-number]`	RADIUS サーバ ホストを指定します。 • *ip-address* — RADIUS サーバ ホストの IP アドレス • *auth-port-number* —（任意）認証要求の UDP 宛先ポート番号。0 に設定する場合、ホストは認証に使用されません。指定しない場合、デフォルトでポート番号 1645 が使用されます。 • *acct-port-number* —（任意）アカウンティング要求の UDP 宛先ポート番号。0 に設定する場合、ホストはアカウンティングに使用されません。指定しない場合、デフォルトでポート番号 1646 が使用されます。
ステップ 21	`Router(config)# radius-server key string`	ルータと RADIUS デーモンの間のすべての RADIUS 通信に使用される認証および暗号キーを設定します。 • *string* — 暗号化されない（平文の）共有キー
ステップ 22	`Router(config)# interface type slot/[subslot]/port`	インターフェイス タイプを設定し、インターフェイス コンフィギュレーション モードを開始します。
ステップ 23	`Router(config-if)# crypto map map-name`	事前に定義した暗号マップ セットをインターフェイスに適用します。 • *map-name* — 暗号マップ セットの識別名

アカウンティング アップデートの設定

セッションが「アップ」の状態でアカウンティング アップデートを送信するには、グローバル コンフィギュレーション モードで、次のオプションの **aaa accounting update periodic** コマンドを入力します。

```
Router(config)# aaa accounting update periodic number
```

number は、アカウンティング サーバにアカウンティング レコードを送信する間隔（分）を表す整数です。

IPSec VPN アカウンティングに関する詳しい設定情報は、次の URL を参照してください。

http://www.cisco.com/en/US/docs/ios/12_2t/12_2t15/feature/guide/ft_evpna.html

IPSec VPN アカウンティングの設定例は、「IPSec VPN アカウンティングの設定例」（p.30-12）を参照してください。

Cisco VRF 対応 IPSec の IPSec および IKE MIB サポートの設定

Cisco VRF 対応 IPSec の IPSec および IKE MIB サポート機能を使用すると、MIB を使用して VRF 対応 IPSec を管理できます。この機能の利点は、VRF 対応 IPSec MIB によって、VRF ごとに詳細な IPSec 統計情報およびパフォーマンス メトリックが提供されることです。

（注）　Cisco VRF 対応 IPSec 機能に対する IPSec および IKE MIB のサポートは、Cisco IOS Release 12.2(33)SRA でのみサポートされています。

Cisco VRF 対応 IPSec の IPSec および IKE MIB サポート機能でサポートされる MIB

Cisco VRF 対応 IPSec の IPSec および IKE MIB サポート機能では、次の MIB がサポートされます。

- CISCO-IPSEC-FLOW-MONITOR-MIB
- CISCO-IPSEC-MIB
- CISCO-IPSEC-POLICY-MAP-MIB — 引き続きサポートされます。ただし、この MIB は特定の VPN VRF インスタンスでなくルータ全体に適用されるため、VRF 対応ではありません。したがって、この MIB に属する Object Identifier（OID; オブジェクト識別子）のポーリングは、グローバル VRF コンテキストに関して実現されます。

Cisco VRF 対応 IPSec の IPSec および IKE MIB サポートの設定

この機能には、特別な設定は必要ありません。SNMP（簡易ネットワーク管理プロトコル）フレームワークを使用して、MIB による VRF 対応 IPSec を管理できます。

Cisco VRF 対応 IPSec の IPSec および IKE MIB サポートの詳細については、次の URL を参照してください。

http://www.cisco.com/en/US/products/ps6441/products_feature_guide09186a00804ff67b.html

SNMP の設定例は、「Cisco VRF 対応 IPSec の IPSec および IKE MIB サポートの設定例」（p.30-14）を参照してください。

設定例

ここでは、次の設定例を示します。

VPN セッションのモニタリングおよび管理の設定例

以下に、IKE ピアで IPSec VPN モニタリングを設定する例を示します。

```
Router(config)# crypto isakmp peer address 10.2.2.9
Router(config-isakmp-peer)# description connection from site
```

IPSec VPN アカウンティングの設定例

ここでは、IPSec VPN アカウンティングの初期設定と、アカウンティング アップデートの設定例を示します。

IPSec VPN アカウンティングの設定例

以下に、IPSec VPN アカウンティング機能をイネーブルにする例を示します。

```
!
! Step 1: add a aaa new-model
!
aaa new-model

!
! Step 2: specifiy a radius server
!
aaa group server radius ar1
 server-private <radius-ip-address> auth-port 1812 acct-port 1813 key <radius-key>

!
! Step 3: specify list of authentication, authorization, and accounting
!
aaa authentication login test_list group ar1 local
aaa authorization network test_list group ar1 local
aaa accounting network test_list start-stop group ar1

!
! Step 4: specify crypto ipsec transform set, isakmp client,
!         and crypto isakmp profile
!
crypto ipsec transform-set ts esp-3des esp-sha-hmac
crypto isakmp client configuration group test
 key world
 pool pool1
crypto isakmp profile test_prof
   vrf ivrf1
   match identity group test
   client authentication list test_list
   isakmp authorization list test_list
   client configuration address respond
   accounting test_list

!
! Step 5: specify dynamic crypto map
!
crypto dynamic-map dmap 10
 set transform-set ts
 set isakmp-profile test_prof
 reverse-route

!
! Step 6: specify crypto map local-address to secure egress interface
!         and apply it to the vlan interface
!
crypto map mymap local-address <secure-egress-interface>
crypto map mymap 10 ipsec-isakmp dynamic dmap

interface Vlan100
 ip vrf forwarding ivrf1
 ip address <ip address> < mask>
 crypto map mymap
 crypto engine slot <x/y> inside
!
! <inside> keyword only applies to SXF release
!
```

IPSec VPN アカウンティング アップデートの設定例

以下に、セッションが「アップ」状態で、アップデートを送信する例を示します。

```
Router(config)# aaa accounting update periodic 1-2147483647
```

IPSec VPN モニタリングの設定例

以下に、IKE ピアで IPSec VPN モニタリングを設定する例を示します。

```
Router(config)# crypto isakmp peer address 10.2.2.9
Router(config-isakmp-peer)# description connection from site
```

Cisco VRF 対応 IPSec の IPSec および IKE MIB サポートの設定例

次の SNMP 例は、VRF が 2 つ存在する一般的なハブ構成に対応しています。出力は、IPSec SA をポーリングした場合に表示される出力です。ルータ 3745b は VRF 対応ルータです。

以下に、2 つの VRF（vrf1 および vrf2）を設定した場合の出力例を示します。

2 つの VRF の設定

```
Router3745b# show running-config
Building configuration...
Current configuration : 6567 bytes
!
version 12.4
service timestamps debug datetime msec localtime
service timestamps log uptime
no service password-encryption
!
hostname ipsecf-3745b
!
boot-start-marker
boot-end-marker
!
no logging console
enable password lab
!
no aaa new-model
!
resource policy
!
memory-size iomem 5
clock timezone PST -8
clock summer-time PDT recurring
ip subnet-zero
ip cef
!
ip vrf vrf1
rd 1:101
context vrf-vrf1-context
route-target export 1:101
route-target import 1:101
!
ip vrf vrf2
rd 2:101
context vrf-vrf2-context
route-target export 2:101
route-target import 2:101
!
no ip domain lookup
!
crypto keyring vrf1-1 vrf vrf1
pre-shared-key address 10.1.1.1 255.255.255.0 key vrf1-1
crypto keyring vrf2-1 vrf vrf2
pre-shared-key address 10.1.2.1 255.255.255.0 key vrf2-1
!
crypto isakmp policy 1
authentication pre-share
!
```

```
crypto isakmp policy 50
authentication pre-share
crypto isakmp key global1-1 address 10.1.151.1
crypto isakmp key global2-1 address 10.1.152.1
crypto isakmp profile vrf1-1
keyring vrf1-1
match identity address 10.1.1.1 255.255.255.255 vrf1
crypto isakmp profile vrf2-1
keyring vrf2-1
match identity address 10.1.2.1 255.255.255.255 vrf2
!
crypto ipsec security-association lifetime kilobytes 99000
crypto ipsec security-association lifetime seconds 5000
!
crypto ipsec transform-set tset ah-sha-hmac esp-des esp-sha-hmac
!
crypto map global1-1 10 ipsec-isakmp
set peer 10.1.151.1
set transform-set tset
match address 151
!
crypto map global2-1 10 ipsec-isakmp
set peer 10.1.152.1
set transform-set tset
match address 152
!
crypto map vrf1-1 10 ipsec-isakmp
set peer 10.1.1.1
set transform-set tset
set isakmp-profile vrf1-1
match address 101
!
crypto map vrf2-1 10 ipsec-isakmp
set peer 10.1.2.1
set transform-set tset
set isakmp-profile vrf2-1
match address 102
!
interface FastEthernet0/0
ip address 10.1.38.25 255.255.255.0
no ip mroute-cache
duplex auto
speed auto
!
interface Serial0/0
no ip address
shutdown
clock rate 2000000
!
interface FastEthernet0/1
no ip address
no ip mroute-cache
shutdown
duplex auto
speed auto
!
interface Serial0/1
no ip address
shutdown
clock rate 2000000
!
interface Serial1/0
no ip address
encapsulation frame-relay
no ip route-cache cef
no ip route-cache
no ip mroute-cache
no keepalive
serial restart-delay 0
```

```
clock rate 128000
no frame-relay inverse-arp
!
interface Serial1/0.1 point-to-point
ip vrf forwarding vrf1
ip address 10.3.1.1 255.255.255.0
no ip route-cache
frame-relay interface-dlci 21
!
interface Serial1/0.2 point-to-point
ip vrf forwarding vrf2
ip address 10.3.2.1 255.255.255.0
no ip route-cache
frame-relay interface-dlci 22
!
interface Serial1/0.151 point-to-point
ip address 10.7.151.1 255.255.255.0
no ip route-cache
frame-relay interface-dlci 151
!
interface Serial1/0.152 point-to-point
ip address 10.7.152.1 255.255.255.0
no ip route-cache
frame-relay interface-dlci 152
!
interface Serial1/1
no ip address
no ip mroute-cache
shutdown
serial restart-delay 0
!
interface Serial1/2
no ip address
encapsulation frame-relay
no ip route-cache cef
no ip route-cache
no ip mroute-cache
no keepalive
serial restart-delay 0
no frame-relay inverse-arp
!
interface Serial1/2.1 point-to-point
ip vrf forwarding vrf1
ip address 10.1.1.2 255.255.255.0
no ip route-cache
frame-relay interface-dlci 21
crypto map vrf1-1
!
interface Serial1/2.2 point-to-point
ip vrf forwarding vrf2
ip address 10.1.2.2 255.255.255.0
no ip route-cache
frame-relay interface-dlci 22
crypto map vrf2-1
!
interface Serial1/2.151 point-to-point
ip address 10.5.151.2 255.255.255.0
no ip route-cache
frame-relay interface-dlci 151
crypto map global1-1
!
interface Serial1/2.152 point-to-point
ip address 10.5.152.2 255.255.255.0
no ip route-cache
frame-relay interface-dlci 152
crypto map global2-1
!
interface Serial1/3
no ip address
```

```
no ip mroute-cache
shutdown
serial restart-delay 0
!
ip default-gateway 10.1.38.1
ip classless
ip route 10.1.1.6 255.255.255.255 10.1.151.1
ip route 10.2.1.6 255.255.255.255 10.1.152.1
ip route 10.6.2.1 255.255.255.255 10.7.151.2
ip route 10.6.2.2 255.255.255.255 10.7.152.2
ip route 172.19.216.110 255.255.255.255 FastEthernet0/0
ip route vrf vrf1 10.20.1.1 255.255.255.255 10.1.1.1
ip route vrf vrf1 10.22.1.1 255.255.255.255 10.30.1.1
ip route vrf vrf2 10.20.2.1 255.255.255.255 10.1.2.1
ip route vrf vrf2 10.22.2.1 255.255.255.255 10.30.1.2
!
ip http server
no ip http secure-server
!
ip access-list standard vrf-vrf1-context
ip access-list standard vrf-vrf2-context
!
access-list 101 permit ip host 10.22.1.1 host 10.20.1.1
access-list 102 permit ip host 10.22.2.1 host 10.20.2.1
access-list 151 permit ip host 10.6.2.1 host 10.1.1.6
access-list 152 permit ip host 10.6.2.2 host 10.2.1.6
snmp-server group abc1 v2c context vrf-vrf1-context read view_vrf1 notify
*tv.FFFFFFFF.FFFFFFFF.FFFFFFFF.F access vrf-vrf1-context
snmp-server group abc2 v2c context vrf-vrf2-context read view_vrf2 notify
*tv.FFFFFFFF.FFFFFFFF.FFFFFFFF.F access vrf-vrf2-context
snmp-server view view_vrf1 iso included
snmp-server view view_vrf2 iso included
snmp-server community abc1 RW
snmp-server community global1 RW
snmp-server community abc2 RW
snmp-server community global2 RW
snmp-server enable traps tty
snmp-server enable traps config
snmp-server host 172.19.216.110 version 2c abc1
snmp-server host 172.19.216.110 vrf vrf1 version 2c abc1 udp-port 2001  ipsec isakmp
snmp-server host 172.19.216.110 version 2c abc2
snmp-server host 172.19.216.110 vrf vrf2 version 2c abc2 udp-port 2002  ipsec isakmp
snmp-server context vrf-vrf1-context
snmp-server context vrf-vrf2-context
!
snmp mib community-map  abc1 context vrf-vrf1-context
snmp mib community-map  abc2 context vrf-vrf2-context
!
control-plane
!
line con 0
exec-timeout 0 0
line aux 0
line vty 0 4
login
!
webvpn context Default_context
ssl authenticate verify all
!
no inservice
!
end
```

両方の VRF のクリア

次に、abc1 と abc2 の両方の VRF を「クリア」して、すべてのカウンタを既知の値に初期化した場合の出力例を示します。

以下に、VRF abc1 をクリアする例を示します。

```
orcas:2> setenv SR_MGR_CONF /users/green1
orcas:3> setenv SR_UTIL_SNMP_VERSION v2c
orcas:5> setenv SR_UTIL_COMMUNITY abc1
orcas:6> setenv SR_MGR_CONF_DIR /users/green1
orcas:7> /auto/sw/packages/snmpr/10.14.2.0/solaris2bin/getmany -v2c 10.1.38.25
cipSecMIBObjects
cipSecMibLevel.0 = 1
cikeGlobalActiveTunnels.0 = 0
cikeGlobalPreviousTunnels.0 = 0
cikeGlobalInOctets.0 = 0
cikeGlobalInPkts.0 = 0
cikeGlobalInDropPkts.0 = 0
cikeGlobalInNotifys.0 = 0
cikeGlobalInP2Exchgs.0 = 0
cikeGlobalInP2ExchgInvalids.0 = 0
cikeGlobalInP2ExchgRejects.0 = 0
cikeGlobalInP2SaDelRequests.0 = 0
cikeGlobalOutOctets.0 = 0
cikeGlobalOutPkts.0 = 0
cikeGlobalOutDropPkts.0 = 0
cikeGlobalOutNotifys.0 = 0
cikeGlobalOutP2Exchgs.0 = 0
cikeGlobalOutP2ExchgInvalids.0 = 0
cikeGlobalOutP2ExchgRejects.0 = 0
cikeGlobalOutP2SaDelRequests.0 = 0
cikeGlobalInitTunnels.0 = 0
cikeGlobalInitTunnelFails.0 = 0
cikeGlobalRespTunnelFails.0 = 0
cikeGlobalSysCapFails.0 = 0
cikeGlobalAuthFails.0 = 0
cikeGlobalDecryptFails.0 = 0
cikeGlobalHashValidFails.0 = 0
cikeGlobalNoSaFails.0 = 0
cipSecGlobalActiveTunnels.0 = 0
cipSecGlobalPreviousTunnels.0 = 0
cipSecGlobalInOctets.0 = 0
cipSecGlobalHcInOctets.0 = 0x00
cipSecGlobalInOctWraps.0 = 0
cipSecGlobalInDecompOctets.0 = 0
cipSecGlobalHcInDecompOctets.0 = 0x00
cipSecGlobalInDecompOctWraps.0 = 0
cipSecGlobalInPkts.0 = 0
cipSecGlobalInDrops.0 = 0
cipSecGlobalInReplayDrops.0 = 0
cipSecGlobalInAuths.0 = 0
cipSecGlobalInAuthFails.0 = 0
cipSecGlobalInDecrypts.0 = 0
cipSecGlobalInDecryptFails.0 = 0
cipSecGlobalOutOctets.0 = 0
cipSecGlobalHcOutOctets.0 = 0x00
cipSecGlobalOutOctWraps.0 = 0
cipSecGlobalOutUncompOctets.0 = 0
cipSecGlobalHcOutUncompOctets.0 = 0x00
cipSecGlobalOutUncompOctWraps.0 = 0
cipSecGlobalOutPkts.0 = 0
cipSecGlobalOutDrops.0 = 0
cipSecGlobalOutAuths.0 = 0
cipSecGlobalOutAuthFails.0 = 0
cipSecGlobalOutEncrypts.0 = 0
cipSecGlobalOutEncryptFails.0 = 0
cipSecGlobalProtocolUseFails.0 = 0
ipSecGlobalNoSaFails.0 = 0
```

```
cipSecGlobalSysCapFails.0 = 0
cipSecHistTableSize.0 = 200
cipSecHistCheckPoint.0 = ready(1)
cipSecFailTableSize.0 = 200
cipSecTrapCntlIkeTunnelStart.0 = enabled(1)
cipSecTrapCntlIkeTunnelStop.0 = enabled(1)
cipSecTrapCntlIkeSysFailure.0 = disabled(2)
cipSecTrapCntlIkeCertCrlFailure.0 = disabled(2)
cipSecTrapCntlIkeProtocolFail.0 = disabled(2)
cipSecTrapCntlIkeNoSa.0 = disabled(2)
cipSecTrapCntlIpSecTunnelStart.0 = enabled(1)
cipSecTrapCntlIpSecTunnelStop.0 = enabled(1)
cipSecTrapCntlIpSecSysFailure.0 = disabled(2)
cipSecTrapCntlIpSecSetUpFailure.0 = disabled(2)
cipSecTrapCntlIpSecEarlyTunTerm.0 = disabled(2)
cipSecTrapCntlIpSecProtocolFail.0 = disabled(2)
cipSecTrapCntlIpSecNoSa.0 = disabled(2)
```

以下に、VRF abc2 をクリアする例を示します。

```
orcas:8> setenv SR_UTIL_COMMUNITY abc2
orcas:9> /auto/sw/packages/snmpr/14.2.0.0/solaris2bin/getmany -v2c 10.1.38.25
cipSecMIBObjects
cipSecMibLevel.0 = 1
cikeGlobalActiveTunnels.0 = 0
cikeGlobalPreviousTunnels.0 = 0
cikeGlobalInOctets.0 = 0
cikeGlobalInPkts.0 = 0
cikeGlobalInDropPkts.0 = 0
cikeGlobalInNotifys.0 = 0
cikeGlobalInP2Exchgs.0 = 0
cikeGlobalInP2ExchgInvalids.0 = 0
cikeGlobalInP2ExchgRejects.0 = 0
cikeGlobalInP2SaDelRequests.0 = 0
cikeGlobalOutOctets.0 = 0
cikeGlobalOutPkts.0 = 0
cikeGlobalOutDropPkts.0 = 0
cikeGlobalOutNotifys.0 = 0
cikeGlobalOutP2Exchgs.0 = 0
cikeGlobalOutP2ExchgInvalids.0 = 0
cikeGlobalOutP2ExchgRejects.0 = 0
cikeGlobalOutP2SaDelRequests.0 = 0
cikeGlobalInitTunnels.0 = 0
cikeGlobalInitTunnelFails.0 = 0
cikeGlobalRespTunnelFails.0 = 0
cikeGlobalSysCapFails.0 = 0
cikeGlobalAuthFails.0 = 0
cikeGlobalDecryptFails.0 = 0
cikeGlobalHashValidFails.0 = 0
cikeGlobalNoSaFails.0 = 0
cipSecGlobalActiveTunnels.0 = 0
cipSecGlobalPreviousTunnels.0 = 0
cipSecGlobalInOctets.0 = 0
cipSecGlobalHcInOctets.0 = 0x00
cipSecGlobalInOctWraps.0 = 0
cipSecGlobalInDecompOctets.0 = 0
cipSecGlobalHcInDecompOctets.0 = 0x00
cipSecGlobalInDecompOctWraps.0 = 0
cipSecGlobalInPkts.0 = 0
cipSecGlobalInDrops.0 = 0
cipSecGlobalInReplayDrops.0 = 0
cipSecGlobalInAuths.0 = 0
cipSecGlobalInAuthFails.0 = 0
cipSecGlobalInDecrypts.0 = 0
cipSecGlobalInDecryptFails.0 = 0
cipSecGlobalOutOctets.0 = 0
cipSecGlobalHcOutOctets.0 = 0x00
cipSecGlobalOutOctWraps.0 = 0
```

```
cipSecGlobalOutUncompOctets.0 = 0
cipSecGlobalHcOutUncompOctets.0 = 0x00
cipSecGlobalOutUncompOctWraps.0 = 0
cipSecGlobalOutPkts.0 = 0
cipSecGlobalOutDrops.0 = 0
cipSecGlobalOutAuths.0 = 0
cipSecGlobalOutAuthFails.0 = 0
cipSecGlobalOutEncrypts.0 = 0
cipSecGlobalOutEncryptFails.0 = 0
cipSecGlobalProtocolUseFails.0 = 0
cipSecGlobalNoSaFails.0 = 0
cipSecGlobalSysCapFails.0 = 0
cipSecHistTableSize.0 = 200
cipSecHistCheckPoint.0 = ready(1)
cipSecFailTableSize.0 = 200
cipSecTrapCntlIkeTunnelStart.0 = enabled(1)
cipSecTrapCntlIkeTunnelStop.0 = enabled(1)
cipSecTrapCntlIkeSysFailure.0 = disabled(2)
cipSecTrapCntlIkeCertCrlFailure.0 = disabled(2)
cipSecTrapCntlIkeProtocolFail.0 = disabled(2)
cipSecTrapCntlIkeNoSa.0 = disabled(2)
cipSecTrapCntlIpSecTunnelStart.0 = enabled(1)
cipSecTrapCntlIpSecTunnelStop.0 = enabled(1)
cipSecTrapCntlIpSecSysFailure.0 = disabled(2)
cipSecTrapCntlIpSecSetUpFailure.0 = disabled(2)
cipSecTrapCntlIpSecEarlyTunTerm.0 = disabled(2)
cipSecTrapCntlIpSecProtocolFail.0 = disabled(2)
cipSecTrapCntlIpSecNoSa.0 = disabled(2)
orcas:10>
orcas:10>
orcas:10>
```

VRF abc1 への ping

以下に、VRF abc1 に ping を送信する例を示します。

```
Router3745a# ping
Protocol [ip]:
Target IP address: 10.22.1.1
Repeat count [5]:
Datagram size [100]:
Timeout in seconds [2]:
Extended commands [n]: y
Source address or interface: 10.20.1.1
Type of service [0]:
Set DF bit in IP header? [no]:
Validate reply data? [no]:
Data pattern [0xABCD]:
Loose, Strict, Record, Timestamp, Verbose[none]:
Sweep range of sizes [n]:
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.22.1.1, timeout is 2 seconds:
Packet sent with a source address of 10.20.1.1
```

VRF abc1 のポーリング

以下に、VRF abc1 にポーリングした場合の出力例を示します。

（注）　ping 送信後に、カウンタはゼロ以外の値を示します。

```
orcas:10>
orcas:12> setenv SR_UTIL_COMMUNITY abc1
orcas:13> /auto/sw/packages/snmpr/10.14.2.0/solaris2bin/getmany -v2c 10.1.38.25
cipSecMIBObjects
cipSecMibLevel.0 = 1
cikeGlobalActiveTunnels.0 = 1
cikeGlobalPreviousTunnels.0 = 0
cikeGlobalInOctets.0 = 336
cikeGlobalInPkts.0 = 2
cikeGlobalInDropPkts.0 = 0
cikeGlobalInNotifys.0 = 1
cikeGlobalInP2Exchgs.0 = 2
cikeGlobalInP2ExchgInvalids.0 = 0
cikeGlobalInP2ExchgRejects.0 = 0
cikeGlobalInP2SaDelRequests.0 = 0
cikeGlobalOutOctets.0 = 344
cikeGlobalOutPkts.0 = 2
cikeGlobalOutDropPkts.0 = 0
cikeGlobalOutNotifys.0 = 0
cikeGlobalOutP2Exchgs.0 = 1
cikeGlobalOutP2ExchgInvalids.0 = 0
cikeGlobalOutP2ExchgRejects.0 = 0
cikeGlobalOutP2SaDelRequests.0 = 0
cikeGlobalInitTunnels.0 = 0
cikeGlobalInitTunnelFails.0 = 0
cikeGlobalRespTunnelFails.0 = 0
cikeGlobalSysCapFails.0 = 0
cikeGlobalAuthFails.0 = 0
cikeGlobalDecryptFails.0 = 0
cikeGlobalHashValidFails.0 = 0
cikeGlobalNoSaFails.0 = 0
cikePeerLocalAddr.1.15.48.49.48.46.48.48.49.46.48.48.49.46.48.48.50.1.15.48.49.48.46.4
8.48
.49.46.48.48.49.46.48.48.49.1 = 0a 01  01 02
cikePeerRemoteAddr.1.15.48.49.48.46.48.48.49.46.48.48.49.46.48.48.50.1.15.48.49.48.46.
48.4
8.49.46.48.48.49.46.48.48.49.1 = 0a 01  01 01
cikePeerActiveTime.1.15.48.49.48.46.48.48.49.46.48.48.49.46.48.48.50.1.15.48.49.48.46.
48.4
8.49.46.48.48.49.46.48.48.49.1 = 13743
cikePeerActiveTunnelIndex.1.15.48.49.48.46.48.48.49.46.48.48.49.46.48.48.50.1.15.48.49
.48.
46.48.48.49.46.48.48.49.46.48.48.49.1 = 1
cikeTunLocalType.1 = ipAddrPeer(1)
cikeTunLocalValue.1 = 010.001.001.002
cikeTunLocalAddr.1 = 0a 01  01 02
cikeTunLocalName.1 = ipsecf-3745b
cikeTunRemoteType.1 = ipAddrPeer(1)
cikeTunRemoteValue.1 = 010.001.001.001
cikeTunRemoteAddr.1 = 0a 01  01 01
cikeTunRemoteName.1 =
cikeTunNegoMode.1 = main(1)
cikeTunDiffHellmanGrp.1 = dhGroup1(2)
cikeTunEncryptAlgo.1 = des(2)
cikeTunHashAlgo.1 = sha(3)
cikeTunAuthMethod.1 = preSharedKey(2)
cikeTunLifeTime.1 = 86400
cikeTunActiveTime.1 = 13752
cikeTunSaRefreshThreshold.1 = 0
```

```
cikeTunTotalRefreshes.1 = 0
cikeTunInOctets.1 = 336
cikeTunInPkts.1 = 2
cikeTunInDropPkts.1 = 0
cikeTunInNotifys.1 = 1
cikeTunInP2Exchgs.1 = 2
cikeTunInP2ExchgInvalids.1 = 0
cikeTunInP2ExchgRejects.1 = 0
cikeTunInP2SaDelRequests.1 = 0
cikeTunOutOctets.1 = 344
cikeTunOutPkts.1 = 2
cikeTunOutDropPkts.1 = 0
cikeTunOutNotifys.1 = 0
cikeTunOutP2Exchgs.1 = 1
cikeTunOutP2ExchgInvalids.1 = 0
cikeTunOutP2ExchgRejects.1 = 0
cikeTunOutP2SaDelRequests.1 = 0
cikeTunStatus.1 = active(1)
cikePeerCorrIpSecTunIndex.1.15.48.49.48.46.48.48.49.46.48.48.49.46.48.48.50.1.15.48.49
.48.
46.48.48.49.46.48.48.49.46.48.48.49.1.1 = 1
cipSecGlobalActiveTunnels.0 = 1
cipSecGlobalPreviousTunnels.0 = 0
cipSecGlobalInOctets.0 = 400
cipSecGlobalHcInOctets.0 = 0x0190
cipSecGlobalInOctWraps.0 = 0
cipSecGlobalInDecompOctets.0 = 400
cipSecGlobalHcInDecompOctets.0 = 0x0190
cipSecGlobalInDecompOctWraps.0 = 0
cipSecGlobalInPkts.0 = 4
cipSecGlobalInDrops.0 = 0
cipSecGlobalInReplayDrops.0 = 0
cipSecGlobalInAuths.0 = 4
cipSecGlobalInAuthFails.0 = 0
cipSecGlobalInDecrypts.0 = 4
cipSecGlobalInDecryptFails.0 = 0
cipSecGlobalOutOctets.0 = 704
cipSecGlobalHcOutOctets.0 = 0x02c0
cipSecGlobalOutOctWraps.0 = 0
cipSecGlobalOutUncompOctets.0 = 704
cipSecGlobalHcOutUncompOctets.0 = 0x02c0
cipSecGlobalOutUncompOctWraps.0 = 0
cipSecGlobalOutPkts.0 = 4
cipSecGlobalOutDrops.0 = 0
cipSecGlobalOutAuths.0 = 4
cipSecGlobalOutAuthFails.0 = 0
cipSecGlobalOutEncrypts.0 = 4
cipSecGlobalOutEncryptFails.0 = 0
cipSecGlobalProtocolUseFails.0 = 0
cipSecGlobalNoSaFails.0 = 0
cipSecGlobalSysCapFails.0 = 0
cipSecTunIkeTunnelIndex.1 = 1
cipSecTunIkeTunnelAlive.1 = true(1)
cipSecTunLocalAddr.1 = 0a 01  01 02
cipSecTunRemoteAddr.1 = 0a 01  01 01
cipSecTunKeyType.1 = ike(1)
cipSecTunEncapMode.1 = tunnel(1)
cipSecTunLifeSize.1 = 99000
cipSecTunLifeTime.1 = 5000
cipSecTunActiveTime.1 = 13749
cipSecTunSaLifeSizeThreshold.1 = 64
cipSecTunSaLifeTimeThreshold.1 = 10
cipSecTunTotalRefreshes.1 = 0
cipSecTunExpiredSaInstances.1 = 0
cipSecTunCurrentSaInstances.1 = 4
cipSecTunInSaDiffHellmanGrp.1 = dhGroup1(2)
cipSecTunInSaEncryptAlgo.1 = des(2)
cipSecTunInSaAhAuthAlgo.1 = hmacSha(3)
cipSecTunInSaEspAuthAlgo.1 = hmacSha(3)
```

```
cipSecTunInSaDecompAlgo.1 = none(1)
cipSecTunOutSaDiffHellmanGrp.1 = dhGroup1(2)
cipSecTunOutSaEncryptAlgo.1 = des(2)
cipSecTunOutSaAhAuthAlgo.1 = hmacSha(3)
cipSecTunOutSaEspAuthAlgo.1 = hmacSha(3)
cipSecTunOutSaCompAlgo.1 = none(1)
cipSecTunInOctets.1 = 400
cipSecTunHcInOctets.1 = 0x0190
cipSecTunInOctWraps.1 = 0
cipSecTunInDecompOctets.1 = 400
cipSecTunHcInDecompOctets.1 = 0x0190
cipSecTunInDecompOctWraps.1 = 0
cipSecTunInPkts.1 = 4
cipSecTunInDropPkts.1 = 0
cipSecTunInReplayDropPkts.1 = 0
cipSecTunInAuths.1 = 4
cipSecTunInAuthFails.1 = 0
cipSecTunInDecrypts.1 = 4
cipSecTunInDecryptFails.1 = 0
cipSecTunOutOctets.1 = 704
cipSecTunHcOutOctets.1 = 0x02c0
cipSecTunOutOctWraps.1 = 0
cipSecTunOutUncompOctets.1 = 704
cipSecTunHcOutUncompOctets.1 = 0x02c0
cipSecTunOutUncompOctWraps.1 = 0
cipSecTunOutPkts.1 = 4
cipSecTunOutDropPkts.1 = 0
cipSecTunOutAuths.1 = 4
cipSecTunOutAuthFails.1 = 0
cipSecTunOutEncrypts.1 = 4
cipSecTunOutEncryptFails.1 = 0
cipSecTunStatus.1 = active(1)
cipSecEndPtLocalName.1.1 =
cipSecEndPtLocalType.1.1 = singleIpAddr(1)
cipSecEndPtLocalAddr1.1.1 = 16 01  01 01
cipSecEndPtLocalAddr2.1.1 = 16 01  01 01
cipSecEndPtLocalProtocol.1.1 = 0
cipSecEndPtLocalPort.1.1 = 0
cipSecEndPtRemoteName.1.1 =
cipSecEndPtRemoteType.1.1 = singleIpAddr(1)
cipSecEndPtRemoteAddr1.1.1 = 14 01  01 01
cipSecEndPtRemoteAddr2.1.1 = 14 01  01 01
cipSecEndPtRemoteProtocol.1.1 = 0
cipSecEndPtRemotePort.1.1 = 0
cipSecSpiDirection.1.1 = in(1)
cipSecSpiDirection.1.2 = out(2)
cipSecSpiDirection.1.3 = in(1)
cipSecSpiDirection.1.4 = out(2)
cipSecSpiValue.1.1 = 3891970674
cipSecSpiValue.1.2 = 1963217493
cipSecSpiValue.1.3 = 3691920464
cipSecSpiValue.1.4 = 3458912974
cipSecSpiProtocol.1.1 = ah(1)
cipSecSpiProtocol.1.2 = ah(1)
cipSecSpiProtocol.1.3 = esp(2)
cipSecSpiProtocol.1.4 = esp(2)
cipSecSpiStatus.1.1 = active(1)
cipSecSpiStatus.1.2 = active(1)
cipSecSpiStatus.1.3 = active(1)
cipSecSpiStatus.1.4 = active(1)
cipSecHistTableSize.0 = 200
cipSecHistCheckPoint.0 = ready(1)
cipSecFailTableSize.0 = 200
cipSecTrapCntlIkeTunnelStart.0 = enabled(1)
cipSecTrapCntlIkeTunnelStop.0 = enabled(1)
cipSecTrapCntlIkeSysFailure.0 = disabled(2)
cipSecTrapCntlIkeCertCrlFailure.0 = disabled(2)
cipSecTrapCntlIkeProtocolFail.0 = disabled(2)
cipSecTrapCntlIkeNoSa.0 = disabled(2)
```

```
cipSecTrapCntlIpSecTunnelStart.0 = enabled(1)
cipSecTrapCntlIpSecTunnelStop.0 = enabled(1)
cipSecTrapCntlIpSecSysFailure.0 = disabled(2)
cipSecTrapCntlIpSecSetUpFailure.0 = disabled(2)
cipSecTrapCntlIpSecEarlyTunTerm.0 = disabled(2)
cipSecTrapCntlIpSecProtocolFail.0 = disabled(2)
cipSecTrapCntlIpSecNoSa.0 = disabled(2)
orcas:14>
orcas:14>
orcas:14>
```

VRF abc2 のポーリング

以下に、VRF abc2 にポーリングした場合の出力例を示します。

（注）　ping が完了しているのは、VRF abc1 のみです。したがって、VRF abc2 のカウンタは初期状態のままです。

```
setenv SR_UTIL_COMMUNITY abc2
orcas:15>
orcas:15> /auto/sw/packages/snmpr/10.14.2.0/solaris2bin/getmany -v2c 10.1.38.25
cipSecMIBObjects
cipSecMibLevel.0 = 1
cikeGlobalActiveTunnels.0 = 0
cikeGlobalPreviousTunnels.0 = 0
cikeGlobalInOctets.0 = 0
cikeGlobalInPkts.0 = 0
cikeGlobalInDropPkts.0 = 0
cikeGlobalInNotifys.0 = 0
cikeGlobalInP2Exchgs.0 = 0
cikeGlobalInP2ExchgInvalids.0 = 0
cikeGlobalInP2ExchgRejects.0 = 0
cikeGlobalInP2SaDelRequests.0 = 0
cikeGlobalOutOctets.0 = 0
cikeGlobalOutPkts.0 = 0
cikeGlobalOutDropPkts.0 = 0
cikeGlobalOutNotifys.0 = 0
cikeGlobalOutP2Exchgs.0 = 0
cikeGlobalOutP2ExchgInvalids.0 = 0
cikeGlobalOutP2ExchgRejects.0 = 0
cikeGlobalOutP2SaDelRequests.0 = 0
cikeGlobalInitTunnels.0 = 0
cikeGlobalInitTunnelFails.0 = 0
cikeGlobalRespTunnelFails.0 = 0
cikeGlobalSysCapFails.0 = 0
cikeGlobalAuthFails.0 = 0
cikeGlobalDecryptFails.0 = 0
cikeGlobalHashValidFails.0 = 0
cikeGlobalNoSaFails.0 = 0
cipSecGlobalActiveTunnels.0 = 0
cipSecGlobalPreviousTunnels.0 = 0
cipSecGlobalInOctets.0 = 0
cipSecGlobalHcInOctets.0 = 0x00
cipSecGlobalInOctWraps.0 = 0
cipSecGlobalInDecompOctets.0 = 0
cipSecGlobalHcInDecompOctets.0 = 0x00
cipSecGlobalInDecompOctWraps.0 = 0
cipSecGlobalInPkts.0 = 0
cipSecGlobalInDrops.0 = 0
cipSecGlobalInReplayDrops.0 = 0
cipSecGlobalInAuths.0 = 0
cipSecGlobalInAuthFails.0 = 0
cipSecGlobalInDecrypts.0 = 0
```

```
cipSecGlobalInDecryptFails.0 = 0
cipSecGlobalOutOctets.0 = 0
cipSecGlobalHcOutOctets.0 = 0x00
cipSecGlobalOutOctWraps.0 = 0
cipSecGlobalOutUncompOctets.0 = 0
cipSecGlobalHcOutUncompOctets.0 = 0x00
cipSecGlobalOutUncompOctWraps.0 = 0
cipSecGlobalOutPkts.0 = 0
cipSecGlobalOutDrops.0 = 0
cipSecGlobalOutAuths.0 = 0
cipSecGlobalOutAuthFails.0 = 0
cipSecGlobalOutEncrypts.0 = 0
cipSecGlobalOutEncryptFails.0 = 0
cipSecGlobalProtocolUseFails.0 = 0
cipSecGlobalNoSaFails.0 = 0
cipSecGlobalSysCapFails.0 = 0
cipSecHistTableSize.0 = 200
cipSecHistCheckPoint.0 = ready(1)
cipSecFailTableSize.0 = 200
cipSecTrapCntlIkeTunnelStart.0 = enabled(1)
cipSecTrapCntlIkeTunnelStop.0 = enabled(1)
cipSecTrapCntlIkeSysFailure.0 = disabled(2)
cipSecTrapCntlIkeCertCrlFailure.0 = disabled(2)
cipSecTrapCntlIkeProtocolFail.0 = disabled(2)
cipSecTrapCntlIkeNoSa.0 = disabled(2)
cipSecTrapCntlIpSecTunnelStart.0 = enabled(1)
cipSecTrapCntlIpSecTunnelStop.0 = enabled(1)
cipSecTrapCntlIpSecSysFailure.0 = disabled(2)
cipSecTrapCntlIpSecSetUpFailure.0 = disabled(2)
cipSecTrapCntlIpSecEarlyTunTerm.0 = disabled(2)
cipSecTrapCntlIpSecProtocolFail.0 = disabled(2)
cipSecTrapCntlIpSecNoSa.0 = disabled(2)
orcas:16>
```