統合脅威防御（UTD）のマルチテナントの設定

統合脅威防御（UTD）のマルチテナントは、複数のユーザに Snort IPS と Web フィルタリングを提供します。1 つの Cisco CSR 1000v インスタンスで 1 つ以上のテナントのポリシーを定義できます。各ポリシーには、脅威検知プロファイルと Web フィルタリングプロファイルを設定できます。次の項では、Unified Threat Defense のマルチテナントを設定する方法について説明します。これらの設定手順で使用されるコマンドの多くは、シングルテナントの設定で使用されるものと似ています。「Snort IPS」および「Web フィルタリング」を参照してください。

統合脅威防御（UTD）のマルチテナントに関する情報

Snort IPS および Web フィルタリングのマルチテナントを使用すると、1 つの Cisco CSR 1000v のインスタンスで 1 つ以上のテナントのポリシーを定義できます。この機能は、Cisco IOS XE Everest 16.6.1 で導入されました。

各テナントは、1 つ以上の VPN ルーティングおよび転送テーブル（VRF）を持つ VPN ルーティングおよび転送インスタンスです。統合脅威防御（UTD）のポリシーは、脅威検知プロファイルと Web フィルタリングプロファイルに関連付けられています。複数のテナントが UTD ポリシーを共有できます。

システムログには、テナントごとの統計情報の生成を可能にする VRF の名前が含まれます。

マルチテナントモードで使用する CLI コマンドは、シングルテナントモードで使用するものと似ています（Snort IPS および Web フィルタリング を参照）。マルチテナントでは、サブモードである `utd engine standard multi-tenancy`に入り、UTD ポリシー、Web フィルタリング、

および脅威検知プロファイルを設定します。`utd engine standard multi-tenancy`のサブモードを終了すると、UTD ポリシーが適用されます。

Web フィルタリングと脅威検知（Snort IPS または IDS）の利点については、次の項で説明します。

- Web フィルタリングの利点
- Snort 仮想サービスインターフェイスの概要 （4 ページ）

Web フィルタリングの概要

Web フィルタリングにより、URL ベースのポリシーとフィルタを設定することで、インターネットへのアクセスを制御できます。Web フィルタリングは、悪意のあるもしくは不要な Web サイトをブロックし、ネットワークのセキュリティを強化することで、Web サイトへのアクセスの制御に役立ちます。個々の URL またはドメイン名をブロックリストに載せ、それらに対して許可リストポリシーを設定できます。レピュテーションまたはカテゴリに基づいて URL を許可またはブロックするようにプロビジョニングすることもできます。

Snort IPS の概要

Snort IPS 機能は、Cisco 4000 シリーズサービス統合型ルータおよび Cisco クラウドサービスルータ 1000v シリーズのブランチオフィスで侵入防止システム（IPS）または侵入検知システム（IDS）を実現します。この機能は、Snort エンジンを使用して IPS および IDS 機能を実現します。

Snort は、リアルタイムでトラフィック分析を行い、IP ネットワークで脅威が検出されたときにアラートを生成するオープンソースのネットワーク IPS です。また、プロトコル分析、コンテンツ検索またはマッチングを実行し、バッファオーバーフロー、ステルスポートスキャンなどのさまざまな攻撃やプローブを検出することもできます。Snort エンジンは、Cisco 4000 シリーズサービス統合型ルータおよび Cisco クラウドサービスルータ 1000v シリーズで仮想コンテナサービスとして実行されます。

Snort IPS 機能は、IPS または IDS 機能を提供するネットワーク侵入検知および防止モードで動作します。ネットワーク侵入検知および防止モードでは、Snort は次のアクションを実行します。

- ネットワークトラフィックをモニタし、定義されたルールセットに照らしあわせて分析します。
- 攻撃の分類を行います。
- 一致したルールに照らしあわせてアクションを呼び出します。

要件に応じて、IPS または IDS モードで Snort を有効にできます。IDS モードでは、Snort はトラフィックを検査し、アラートを報告しますが、攻撃を防ぐためのアクションは実行しません。IPS モードでは、侵入検知に加えて、攻撃を防ぐためのアクションを実行します。

Snort IPS はトラフィックをモニタし、イベントを外部ログサーバまたは IOS syslog に報告します。IOS syslog へのロギングを有効にすると、ログメッセージが大量に発生する可能性があるため、パフォーマンスに影響する場合があります。Snort ログに対応する外部のサードパーティ製のモニタリングツールを、ログの収集と分析に使用できます。

Snort IPS ソリューション

Snort IPS ソリューションは、次のエンティティで構成されています。

- Snort センサー：トラフィックをモニタして、設定されたセキュリティポリシー（署名、統計情報、プロトコル分析など）に基づいて異常を検出し、アラートサーバまたはレポートサーバにアラートメッセージを送信します。Snort センサーは、仮想コンテナサービスとしてルータに導入されます。
- 署名ストア：定期的に更新される Cisco 署名パッケージをホストします。これらの署名パッケージは、定期的にもしくはオンデマンドで Snort センサーにダウンロードされます。検証済みの署名パッケージは Cisco.com に掲載されます。設定に基づいて、署名パッケージを Cisco.com またはローカルサーバからダウンロードできます。

  次のドメインは、次の cisco.com から署名パッケージをダウンロードするプロセスにおいてルータによってアクセスされます。

  - api.cisco.com
  - apx.cisco.com
  - cloudsso.cisco.com
  - cloudsso-test.cisco.com
  - cloudsso-test3.cisco.com
  - cloudsso-test4.cisco.com
  - cloudsso-test5.cisco.com
  - cloudsso-test6.cisco.com
  - cloudsso.cisco.com
  - download-ssc.cisco.com
  - dl.cisco.com
  - resolver1.opendns.com
  - resolver2.opendns.com

（注） 署名パッケージを保持するためにローカルサーバから署名パッケージをダウンロードする場合は、HTTP のみに対応します。

Snort センサーが署名パッケージを取得するには、Cisco.com の認証情報を使用して、署名パッケージを Cisco.com からローカルサーバに手動でダウンロードする必要があります。

URL が IP アドレスとして指定されていない場合、Snort コンテナは（ルータに設定された DNS サーバ上で）ドメイン名ルックアップを実行して、Cisco.com によるまたはローカルサーバ上の自動署名更新の場所を解決します。

- アラートまたはレポートサーバ：Snort センサーからアラートイベントを受信します。Snort センサーによって生成されたアラートイベントは、IOS syslog または外部 syslog サーバ、もしくは IOS syslog と外部 syslog サーバの両方に送信できます。Snort IPS ソリューションに付属している外部ログサーバはありません。
- 管理：Snort IPS ソリューションを管理します。管理は、IOS CLI を使用して設定します。Snort センサーには直接アクセスできず、すべての設定は IOS CLI を使用してのみ行えます。

Snort 仮想サービスインターフェイスの概要

Snort センサーは、ルータ上でサービスとして動作します。サービスコンテナは、仮想テクノロジーを使用して、アプリケーション用の Cisco デバイスにホスティング環境を提供します。

Snort トラフィック検査は、インターフェイス単位で、または対応しているすべてのインターフェイスでグローバルに有効にできます。検査対象のトラフィックは Snort センサーに転送され、再度投入されます。侵入検知システム（IDS）では、識別された脅威がログイベントとして報告され、許可されます。ただし、侵入防止システム（IPS）では、ログイベントとともに攻撃を防ぐためのアクションが実行されます。

Snort センサーには 2 つの VirtualPortGroup インターフェイスが必要です。最初の VirtualPortGroup インターフェイスは管理トラフィックに使用され、2 つ目は転送プレーンと Snort 仮想コンテナサービス間のデータトラフィックに使用されます。これらの VirtualPortGroup インターフェイスには、ゲスト IP アドレスを設定する必要があります。管理 VirtualPortGroup インターフェイスに割り当てられた IP サブネットは、署名サーバおよびアラート / 報告サーバと通信できる必要があります。

2 つ目の VirtualPortGroup インターフェイスの IP サブネットは、このインターフェイス上のトラフィックがルータ内部にあるため、カスタマーネットワーク上でルーティング可能であってはなりません。内部サブネットを外部に公開することはセキュリティ上のリスクとなります。2 つ目の VirtualPortGroup サブネットには 192.0.2.0/30 の IP アドレス範囲を使用することをお勧めします。192.0.2.0/24 のサブネットを使用することは、RFC 3330 で定義されています。

仮想サービスが実行されているルータと同じ管理ネットワークで、Snort 仮想コンテナサービスの IP アドレスを割り当てることができます。この設定は、syslog またはアップデートサーバが管理ネットワーク上にあり、他のインターフェイスからアクセスできない場合に役立ちます。

統合脅威防御（UTD）のマルチテナントの設定に関する制約事項

-
- ドメインベースのフィルタリングには対応しません。
- 各 Cisco CSR 1000v インスタンスで最大25のテナントに対応します。
- 最大 25 のポリシーに対応します。
- Cisco CSR 1000v では、最大 50,000 の同時セッションに対応します。
-
- ブロックリストまたは許可リストのルールは、正規表現のパターンのみに対応します。現在、ブロックリストまたは許可リストのルールごとに 64 のパターンに対応しています。ただし、各テナントには複数のルールを設定できます。
- ローカルブロックサーバは、HTTPS ブロックページの提供には対応していません。URL フィルタがブロックページまたはリダイレクトメッセージを挿入しようとする場合、HTTPS トラフィックには対応しません。
- URL にユーザ名とパスワードがある場合、ブロックリストまたは許可リストのパターンと一致する前に、URL フィルタがユーザ名とパスワードを URL から削除することはしません。ただし、カテゴリまたはレピュテーションルックアップにはこの制限はなく、ルックアップの前に URL からユーザ名とパスワードを削除します。
- HTTPS 検査は制限されています。Web フィルタリングでは、サーバ証明書を使用して URL およびドメイン情報を取得します。完全な URL のパスを検査することはできません。
- UTD は、VRF 間シナリオにおいては WCCP および NBAR との相互運用は行いません。
- Snort IPS コマンドの `threat inspection profile` *`profile-name`* は、ID（番号）ではなく英数字のプロファイル名を使用します。

統合脅威防御（UTD）のマルチテナントの設定方法

対応しているデバイスに Unified Threat Defense のマルチテナント機能を導入するには、次のタスクを実行します。

始める前に

マルチテナント用に Web フィルタリングおよび脅威検知をインストールするデバイスをプロビジョニングします。この機能は現在、Cisco CSR 1000v でのみ対応しています。

ライセンスを取得します。UTD は、セキュリティパッケージを実行しているルータでのみ使用でき、サービスを有効にするにはセキュリティライセンスが必要となります。セキュリティライセンスの取得については、シスコサポートにお問い合わせください。

手順の概要

1. 仮想サービスをインストールしてアクティブにします。マルチテナント用の UTD OVA ファイルのインストール （6 ページ）
2. VirtualPortGroup のインターフェイスおよび仮想サービスを設定します。マルチテナント用の VirtualPortGroup インターフェイスと仮想サービスの設定方法 （7 ページ）
3. VRF を設定します。マルチテナント用の VRF の設定方法 （10 ページ）
4. マルチテナント用の脅威検知と Web フィルタリングを設定します。マルチテナント Web フィルタリングおよび脅威検知の設定方法 （11 ページ）

手順の詳細

ステップ 1 仮想サービスをインストールしてアクティブにします。マルチテナント用の UTD OVA ファイルのインストール （6 ページ）

ステップ 2 VirtualPortGroup のインターフェイスおよび仮想サービスを設定します。マルチテナント用の VirtualPortGroup インターフェイスと仮想サービスの設定方法 （7 ページ）

ステップ 3 VRF を設定します。マルチテナント用の VRF の設定方法 （10 ページ）

ステップ 4 マルチテナント用の脅威検知と Web フィルタリングを設定します。マルチテナント Web フィルタリングおよび脅威検知の設定方法 （11 ページ）

マルチテナント用の UTD OVA ファイルのインストール

仮想サービスの OVA ファイルは、仮想マシンの圧縮された「インストール可能な」バージョンを含むオープン仮想アーカイブファイルです。この OVA ファイルをルータにダウンロードしてから、仮想サービスをインストールする必要があります。仮想サービスの OVA ファイルは、ルータにインストールされている Cisco IOS XE リリースイメージには付属していません。OVA ファイルは、ルータのフラッシュメモリに事前にインストールされている場合があります。

OVA ファイルをインストールするには、セキュリティライセンス付きの Cisco IOS XE イメージを使用する必要があります。インストール中に、セキュリティライセンスのチェックが行われます。

仮想サービスのインストール例：

```
Device> enable
Device# virtual-service install name utd package
bootflash:utdsnort.1.0.4_SV2983_XE_16_6.20170623_174453_RELEASE.ova
Device# show virtual-service list

Name Status    Package Name
```

```
------------------------------------------------------------------------------
utd Activated utdsnort.1.0.4_SV2983_XE_16_6.20170
```

仮想サービスのアップグレードの例：

```
Device> enable
Device# virtual-service upgrade name utd package
bootflash:utdsnort.1.0.4_SV2983_XE_16_6.20170623_174453_RELEASE.ova
Device# show virtual-service list

Name Status   Package Name
------------------------------------------------------------------------------
utd Activated utdsnort.1.0.4_SV2983_XE_16_6.20170
```

仮想サービスのアンインストールの例：

```
Device> enable
Device# virtual-service uninstall name utd
Device# show virtual-service list

Virtual Service List:
```

マルチテナント用のVirtualPortGroupインターフェイスと仮想サービスの設定方法

この手順に示すように、マルチテナントの場合、2 つの VirtualPortGroup インターフェイスと両方のインターフェイスのゲスト IP アドレスを設定する必要があります。

（注） データトラフィック用の VirtualPortGroup インターフェイスは、プライベートまたはルーティング不可の IP アドレスを使用する必要があります。このインターフェイスには、IP アドレスの範囲として 192.0.2.0 / 30を 使用することを推奨します。

手順の概要

1. **enable**
2. **configure terminal**
3. **interface VirtualPortGroup** *interface-number*
4. **ip address** *ip-address mask*
5. **exit**
6. **interface VirtualPortGroup** *interface-number*
7. **ip address** *ip-address mask*
8. **exit**
9. **virtual-service** *name*
10. **profile multi-tenancy**
11. **vnic gateway VirtualPortGroup** *interface-number*
12. **guest ip address** *ip-address*
13. **exit**
14. **vnic gateway VirtualPortGroup** *interface-number*

15. **guest ip address** *ip-address*
16. **exit**
17. **activate**
18. **end**
19. **show virtual-service list**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**interface VirtualPortGroup** *interface-number* 例： `Device(config)# interface VirtualPortGroup 0`	インターフェイス設定モードに入り、VirtualPortGroup インターフェイスを設定します。このインターフェイスは、管理インターフェイスの GigabitEthernet0 が使用されていない場合に管理トラフィックに対して使用されます。
ステップ 4	**ip address** *ip-address mask* 例： `Device(config-if)# ip address 10.1.1.1 255.255.255.252`	インターフェイスのプライマリ IP アドレスを設定します。このインターフェイスは、署名アップデートサーバおよび外部ログサーバにルーティング可能である必要があります。
ステップ 5	**exit** 例： `Device(config-if)# exit`	インターフェイス設定モードを終了し、グローバル設定モードに戻ります。
ステップ 6	**interface VirtualPortGroup** *interface-number* 例： `Device(config)# interface VirtualPortGroup 1`	インターフェイスを設定し、インターフェイス設定モードを開始します。VirtualPortGroup インターフェイスを設定します。このインターフェイスは、データトラフィックに使用されます。
ステップ 7	**ip address** *ip-address mask* 例： `Device(config-if)# ip address 192.0.2.1 255.255.255.252`	インターフェイスのプライマリ IP アドレスを設定します。この IP アドレスは、外部ネットワークに対してルーティング不能である必要があります。IP アドレスは、推奨される 192.0.2.0/30 のサブネットから割り当てられます。
ステップ 8	**exit** 例： `Device(config-if)# exit`	インターフェイス 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。

	コマンドまたはアクション	目的
ステップ 9	**virtual-service** *name* 例： `Device(config)# virtual-service utd`	仮想コンテナサービスを設定し、仮想サービス設定モードに入ります。*name* 引数は、仮想コンテナサービスを識別するために使用される論理名です。
ステップ 10	**profile multi-tenancy** 例： `Device(config-virt-serv)#profile multi-tenancy`	リソースプロファイルを設定します。マルチテナントモードの場合（Cisco CSR 1000v のみ）、このプロファイル マルチテナント コマンドを設定する必要があります。
ステップ 11	**vnic gateway VirtualPortGroup** *interface-number* 例： `Device(config-virt-serv)# vnic gateway VirtualPortGroup 0`	仮想サービスの仮想ネットワーク インターフェイス カード（vNIC：virtual network interface card）設定モードに入ります。仮想コンテナサービス用の vNIC ゲートウェイ インターフェイスを作成し、vNIC ゲートウェイ インターフェイスを仮想ポートグループ インターフェイスにマッピングします。これは、手順3で設定したインターフェイスです。
ステップ 12	**guest ip address** *ip-address* 例： `Device(config-virt-serv-vnic)# guest ip address 10.1.1.2`	vNIC ゲートウェイ インターフェイスのゲスト vNIC アドレスを設定します。
ステップ 13	**exit** 例： `Device(config-virt-serv-vnic)# exit`	仮想サービスの vNIC 設定モードを終了し、仮想サービス設定モードに戻ります。
ステップ 14	**vnic gateway VirtualPortGroup** *interface-number* 例： `Device(config-virt-serv)# vnic gateway VirtualPortGroup 1`	仮想サービスの vNIC 設定モードに入ります。仮想コンテナサービス用の vNIC ゲートウェイ インターフェイスを設定し、インターフェイスを仮想ポートグループにマッピングします。手順6で設定されたインターフェイス（*interface-number*）は、ユーザトラフィックをモニタするために Snort エンジンによって使用されます。
ステップ 15	**guest ip address** *ip-address* 例： `Device(config-virt-serv-vnic)# guest ip address 192.0.2.2`	vNIC ゲートウェイ インターフェイスのゲスト vNIC アドレスを設定します。
ステップ 16	**exit** 例： `Device(config-virt-serv-vnic)# exit`	仮想サービスの vNIC 設定モードを終了し、仮想サービス設定モードに戻ります。
ステップ 17	**activate** 例：	仮想コンテナサービスにインストールされたアプリケーションをアクティブにします。

	コマンドまたはアクション	目的
	`Device(config-virt-serv)# activate`	
ステップ 18	**end** 例： `Device(config-virt-serv)# end`	仮想サービス設定モードを終了し、特権EXECモードに戻ります。

ステップ 19	**show virtual-service list**
	例：
	`Device# show virtual-service list`
	
	`Virtual Service List:`
	
	`Name  Status  Package Name`
	`----------`
	`utd  Activated`
	`utdsnort.1.0.4_SV2983_XE_16_6.20170`	

マルチテナント用の VRF の設定方法

この手順では、テナントの VRF を設定するために必要な一般的な手順について説明します。この手順は後にマルチテナント Web フィルタリングおよび脅威検知の設定方法 （11 ページ）で使います。

（注） VRF 間トラフィックの場合、2 つの VRF 間を流れるトラフィックに UTD 用の入力インターフェイスと出力インターフェイスが設定されている場合、セッションを表す VRF を決定するルールが適用されます。選択した VRF の UTD ポリシーは、VRF 間トラフィックのすべてのパケットに適用されます。

手順の概要

1. **vrf definition** *vrf-name*
2. **rd** *route-distinguisher*
3. **address-family ipv4**
4. **exit address-family**
5. VRF ごとに手順 1 ～ 4 を繰り返します。

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**vrf definition** *vrf-name* 例： `Device(config)# vrf definition 100`	VRF 名を定義し、VRF 設定モードに入ります。

	コマンドまたはアクション	目的
ステップ 2	**rd** *route-distinguisher* 例： `Device(config-vrf)# rd 100:1`	ルーティングテーブルと転送テーブルを作成し、ルート識別子を「VRF 名」という名前の VRF インスタンスに関連付けます。ルータはルート識別子を使用して、パケットが属する VRF を識別します。ルート識別子は、次の 2 つのタイプのいずれかとなります。 • 自律システム関連。AS 番号 xxx および任意の番号 y：xxx:y • IP アドレス関連。IP アドレス A.B.C.D および任意の番号 y：A.B.C.D:y
ステップ 3	**address-family ipv4** 例： `Device(config-vrf)# address-family ipv4`	IP バージョン 4 アドレスを使用してルーティングセッションを設定するためのアドレスファミリ設定モードに入ります。
ステップ 4	**exit address-family** 例： `Device(config-vrf-af)# exit`	アドレスファミリ設定モードを終了します。
ステップ 5	VRF ごとに手順 1 ～ 4 を繰り返します。	

マルチテナント Web フィルタリングおよび脅威検知の設定方法

マルチテナント（複数のテナントまたは VRF）の脅威検知（IPS または IDS）および Web フィルタリングを設定するには、次の手順を実行します。

この手順では、ブロックリストと許可リストの定義を最初の手順 1 ー 5 に示します。主な設定手順（マルチテナント用の UTD 標準エンジンの設定モード）は、手順 6 以降に示しています。

（注） シングルテナント用の脅威検知と Web フィルタリングの詳細については、Snort IPS および Web フィルタリング を参照してください。

始める前に

`no utd engine standard` コマンドを使用して、既存のシングルテナントの UTD 設定を削除します。

テナントごとに VRF を事前に設定しておく必要があります（マルチテナント用の VRF の設定方法 （10 ページ） を参照）。

手順

	コマンドまたはアクション	目的
ステップ 1	**parameter-map type regex** *blacklist-name* 例： `Device(config)# parameter-map type regex urlf-blacklist1`	ブロックリストのパラメータマップを定義します。これは、後に手順 17 で使用します。

ステップ 2	**pattern** *URL-name*
	例：
	`Device(config-profile)# pattern www\.cnn\.com`
	`Device(config-profile)# pattern www\.msnbc\.com`
		ブロックリストに登録する URL を定義します。*URL-name* 内のピリオドの前には、必ずエスケープ「\」文字を入れてください。ブロックリストに複数の URL を設定するには、この手順を繰り返します。

ステップ 3	**parameter-map type regex** *whitelist-name* 例： `Device(config-profile)# parameter-map type regex urlf-whitelist1`	許可リストのパラメータマップを定義します。これは、後に手順 20 で使用します。
ステップ 4	**pattern** *URL-name* 例： `Device(config-profile)# pattern www\.nfl\.com`	許可リストに登録する URL を定義します。ブロックリストの URL では、*URL-name* 内のピリオドの前には、必ずエスケープ「\」文字を入れてください。許可リストに複数の URL を設定するには、この手順を繰り返します。
ステップ 5	**exit** 例： `Device(config-profile)# exit`	
ステップ 6	**utd multi-tenancy** 例： `Device(config)# utd multi-tenancy`	このコマンドは、次の `utd engine standard multi-tenancy` コマンドに備えて、スイッチの役割を果たします。
ステップ 7	**utd engine standard multi-tenancy** 例： `Device(config)# utd engine standard multi-tenancy`	マルチテナント用の UTD 標準エンジンの設定モードに入ります。 （注） 後に手順 50 で UTD 標準エンジンの設定モードを終了すると、ポリシー設定が適用されます。
ステップ 8	**web-filter sourcedb** *sourcedb-number* 例： `Device(config)# web-filter sourcedb 1`	Web フィルタリングのソース DB プロファイル（*sourcedb-number* は数字）を設定します。これは、後に手順 29 で使用されます。

<table>
<tr><th></th><th>コマンドまたはアクション</th><th>目的</th></tr>
<tr><td>ステップ 9</td><td>logging level {alerts | critical | debugging | emergencies | errors | informational | notifications | warnings}
例：
<code>Device(config)# logging level errors</code></td><td>Web フィルタリングイベントに関して報告されるシステムメッセージのレベルを設定します。指定したレベル以下のメッセージが報告されます。（各レベルには、次の表に示す数値があります）
表 1: システム メッセージのシビラティ（重大度）
<table><tr><th>レベル</th><th>説明</th></tr><tr><td>0：emergencies</td><td>システムが使用不可</td></tr><tr><td>1：alerts</td><td>即時処理が必要</td></tr><tr><td>2：critical</td><td>クリティカル状態</td></tr><tr><td>3：errors</td><td>エラー状態</td></tr><tr><td>4：warnings</td><td>警告状態</td></tr><tr><td>5：notifications</td><td>正常だが注意を要する状態</td></tr><tr><td>6：informational</td><td>情報メッセージだけ</td></tr><tr><td>7：debugging</td><td>デバッグ実行時にのみ表示</td></tr></table></td></tr>
<tr><td>ステップ 10</td><td>web-filter block local-server profile profile-id
例：
<code>Device(config-utd-multi-tenancy)# web-filter block local-server profile 1</code>
コンテンツのテキストはローカルサーバによって表示されます。</td><td>Webフィルタリングのローカルブロックサーバのプロファイルを設定します。profile-id の値の範囲は 1 ー 255 です。
「ローカルブロックサーバを使用した URL ベースの Web フィルタリングの設定」を参照してください。
（注） マルチテナント用のコマンドを設定する場合、シングルテナントと比較して、最初の <code>utd</code> というキーワードを使用しないでください。</td></tr>
<tr><td>ステップ 11</td><td>block-page-interface loopback id
例：
<code>Device(config-utd-mt-webf-blk-srvr)# block-page-interface loopback 110</code></td><td>ループバックインターフェイスにこのプロファイルを関連付けます。このループバックインターフェイスの IP アドレスは、ブロックローカルサーバの IP アドレスとして使用されます。</td></tr>
<tr><td>ステップ 12</td><td>content text display-text
例：</td><td>ブロックされたページにアクセスした後に表示される警告テキストを指定します。</td></tr>
</table>

	コマンドまたはアクション	目的
	`Device(config-utd-mt-webf-blk-srvr)# content text "Blocked by Web-Filter"`	
ステップ 13	**http-ports** *port-number* 例： `Device(config-utd-mt-webf-blk-srvr)# http-ports 80`	http ポート値は、カンマで区切られたポートの文字列です。nginx HTTP サーバはこれらのポートをリッスンします。

ステップ 14	**web-filter block page profile** *profile-name*
	例：
	`Device(config-utd-multi-tenancy)# web-filter block page profile 1`
	`Device(config-utd-mt-webf-block-urc)# text "this page is blocked"`
		インラインブロックページを使用した URL ベースの Web フィルタリングの設定 を参照してください。ただし、マルチテナント用にここで使用されるコマンドは、シングルテナント用に使用される `utd` キーワードを使用しません）。

ステップ 15	**web-filter url profile** *web-filter-profile-id*
	例：
	`Device(config-utd-multi-tenancy)# web-filter url profile 1`
	`Device(config-utd-mt-webfltr-url)#`
		Web フィルタリングの URL プロファイルである *web-filter-profile-id* を指定します。値は 1 ～ 255 です。このコマンドの後、ブロックリスト、許可リスト、およびカテゴリのアラートを設定できます。詳細については、「インラインブロックページを使用した URL ベースの Web フィルタリングの設定」を参照してください。
		（注） マルチテナント用のコマンドを設定する場合、シングルテナントと比較して、最初の `utd` というキーワードを使用しないでください。

ステップ 16	**blacklist** 例： `Device(config-utd-mt-webfltr-url)# blacklist`	Web フィルタリングのブロックリストの設定モードに入ります。
ステップ 17	**parameter-map regex** *blacklist-name* 例： `Device(config-utd-mt-webf-url-bl)# parameter-map regex urlf-blacklist1`	手順 1 で前に定義したブロックリストを使用して、パラメータマップの正規表現を指定します。

ステップ 18	**exit**
	例：
	`Device(config-utd-mt-webf-url-bl)# exit`
	`Device(config-utd-mt-webfltr-url)#`
		Web フィルタリングのブロックリストの設定モードを終了します。

ステップ 19	**whitelist** 例：	Web フィルタリングの許可リストの設定モードに入ります。

	コマンドまたはアクション	目的
	`Device(config-utd-mt-webfltr-url)# whitelist` `Device(config-utd-mt-webf-url-wl)#`	
ステップ 20	**parameter-map regex** *whitelist-name* 例： `Device(config-utd-mt-webf-url-wl)# parameter-map regex urlf-list1`	手順3で前に定義した許可リストを使用して、パラメータマップの正規表現を指定します。

ステップ 21	**exit**
	例：
	`Device(config-utd-mt-webf-url-wl)# exit`
	`Device(config-utd-mt-webfltr-url)#`
		Web フィルタリングの許可リストの設定モードを終了します。

ステップ 22	**exit**
	例：
	`Device(config-utd-mt-webfltr-url)# exit`
	`Device(config-utd-multi-tenancy)#`
		Web フィルタリングの URL プロファイルモードを終了します。

ステップ 23	**utd global** 例： `Device(config-utd-multi-tenancy)# utd global`	`utd global` に入力されたコマンドは、すべてのテナントまたはポリシーに適用されます。Cisco CSR 1000v インスタンスの場合のコマンド例は、`logginghost syslog` および `threat inspection` などです。

ステップ 24	**logging** {**host** *hostname* | **syslog**}
	例：
	この例では、アラートは指定されたホストのログファイルに記録されます。
	`Device(config-utd-mt-utd-global)# logging host systemlog1`
	例：
	この例では、アラートは IOS syslog に記録されます。
	`Device(config-utd-mt-utd-global)# logging syslog`
		`logging` コマンドは、syslog メッセージの送信先となるホスト名または IOS syslog を指定します。

ステップ 25	**threat inspection** 例： `Device(config-utd-mt-utd-global)# threat inspection`	グローバル脅威検知モードに入ります。
ステップ 26	**signature update server** {**cisco** | **url** *url* } [**username** *username* [**password** *password*]] 例：	署名更新サーバのパラメータを設定します。サーバの詳細で署名更新パラメータを指定する必要があります。署名の更新に www.cisco.com を使用する場合は、ユーザ名とパスワードを入力する必要がありま

	コマンドまたはアクション	目的
	`Device(config-utd-mt-utd-global-threat)# signature update server cisco username abcd password cisco123`	す。署名の更新にローカルサーバを使用する場合は、サーバ設定に基づいてユーザ名とパスワードを指定できます。ルータは、インターネットに接続することでドメイン名を解決できる必要があります。
ステップ 27	**signature update occur-at** {**daily** | **monthly** *day-of-month* | **weekly** *day-of-week*} *hour minute* 例： `Device(config-utd-mt-utd-global-threat)# signature update occur-at daily 0 0`	署名の更新間隔パラメータを設定します。この設定をすることで、午前0時に署名の更新がトリガーされます。
ステップ 28	**web-filter** 例： `Device(config-utd-mt-utd-global-threat)# web-filter`	このコマンドは、次の `sourcedb` コマンドと組み合わせて使用し、Web フィルタリングの URL ソースデータベースを指定します。
ステップ 29	**sourcedb** *sourcedb-number* 例： `Device(config-utd-mt-utd-global-threat)# sourcedb 1`	Web フィルタリングのソースデータベースを割り当てます。アクティブにできるソースデータベースは 1 つだけです。
ステップ 30	**exit** 例： `Device(config-utd-mt-utd-global-threat)# exit`	脅威検知設定モードを終了します。
ステップ 31	**exit** 例： `Device(config-utd-mt-global)# exit`	グローバル更新設定モードを終了します。
ステップ 32	**threat-inspection whitelist profile** *policy-name* 例： `Device(config-utd-multi-tenancy)# threat-inspection whitelist profile wh101`	許可リストのプロファイルを現在設定されているポリシーに関連付けます。同様のコマンドがシングルテナントで使用されますが、`utd` キーワードを使用します。
ステップ 33	**signature id** *id* 例： `Device(config-utd-mt-list)# signature id 101`	以前に脅威として特定した ID である *id* を指定します。たとえば、アラートのログファイルの ID を確認した後などです。 複数の署名 ID に対してこのコマンドを繰り返します。
ステップ 34	**exit** 例： `Device(config-utd-mt-whitelist)# exit`	許可リストの設定モードを終了します。

	コマンドまたはアクション	目的
ステップ 35	**threat-inspection profile** *profile-name* 例： `Device(config-utd-multi-tenancy)# threat-inspection profile 101`	脅威検知プロファイルを設定することで、複数のテナントにより再利用できるようになります。複数の脅威検知プロファイルを設定できます。プロファイル内では、複数の許可リストを設定できます。 `profile-name` は英数字です。
ステップ 36	**threat** {**detection** | **protection** } 例： `Device(config-utd-mt-threat)# threat protection`	Snort エンジンの動作モードとして侵入検知システム（IDS）または侵入防止システム（IPS）を指定します。 デフォルトは **threat detection** です。
ステップ 37	**policy** {**balanced** | **connectivity** | **security**} 例： `Device(config-utd-mt-threat)# policy security`	Snort エンジンのセキュリティポリシーを設定します。 • デフォルトのセキュリティポリシータイプは **balanced** です。
ステップ 38	**logging level**{**alert** | **crit** | **debug** | **emerg** | **err** | **info** | **notice** | **warning**}	次のいずれかのカテゴリのログを表示します。 • alert：アラートレベルのログを表示します（シビラティ（重大度）＝2）。 • crit：クリティカルレベルのログ（シビラティ（重大度）＝3） • debug：すべてのログ（シビラティ（重大度）＝8） • emerg：緊急レベルのログ（シビラティ（重大度）＝1） • err：エラーレベルのログ（シビラティ（重大度）＝4）デフォルト。 • info：情報レベルのログ（シビラティ（重大度）＝7） • notice：通知レベルのログ（シビラティ（重大度）＝6） • warning：警告レベルのログ（シビラティ（重大度）＝5）
ステップ 39	**whitelist profile** *profile-name* 例： `Device(config-utd-mt-threat)# whitelist profile wh101`	また、許可リストプロファイルを別の場所にある許可リストのプロファイルに対してのみ指定することもできます（上記の `threat-inspection whitelist profile` コマンド）。

	コマンドまたはアクション	目的
		（オプション）UTD エンジンで許可リストを有効にします。
ステップ 40	**exit** 例： `Device(config-utd-mt-threat)# exit`	脅威検知モードを終了します。
ステップ 41	脅威検知プロファイルを追加するには、手順 35 ー 40 を繰り返します。	
ステップ 42	**policy** *policy-name* 例： `Device(config-utd-multi-tenancy)# policy pol101`	複数のテナントに関連付けるポリシーを定義します。脅威検知（IPS）および Web フィルタリングのプロファイルがポリシーに追加されます。
ステップ 43	**vrf** [*vrf-name* | global] 例： この例では、2 つのテナント（VRF）と 2 つのポリシーの設定を示します。 `Device(config-utd-mt-policy)# vrf vrf101`	UTD ポリシーを使用する VRF（テナント）ごとに `vrf vrf-name` コマンドを繰り返し入力します。以前に定義されたこれらの VRF については、マルチテナント用の VRF の設定方法 （10 ページ） を参照してください。 または、`vrf global` を使用してグローバル（デフォルト）VRF に関連付け、インターフェイスで VRF を有効にします。
ステップ 44	**all-interfaces** 例： `Device(config-utd-mt-policy)# all-interfaces`	（オプション）VRF のすべてのインターフェイスをポリシーに関連付けます。
ステップ 45	**threat-inspection profile** *profile-name* 例： `Device(config-utd-mt-policy)# threat-inspection profile 101`	（オプション）以前に定義した脅威検知プロファイルにポリシーを関連付けます。手順 35 を参照してください。
ステップ 46	**web-filter url profile** *web-filter-profile-id* 例： `Device(config-utd-mt-policy)# web-filter url profile 1`	（オプション）以前に定義した Web フィルタリングのプロファイルにポリシーを関連付けます。手順 15 を参照してください。
ステップ 47	**fail close** 例： `Device(config-utd-mt-policy)# fail close`	（オプション）エンジン障害時に IPS または IDS パケットをドロップします。デフォルトは `fail open` です。
ステップ 48	**exit**	ポリシー設定モードを終了します。
ステップ 49	各ポリシーに対して手順 42 ー 48 を繰り返します。	

	コマンドまたはアクション	目的
ステップ 50	**exit** 例： `Device(config-utd-multi-tenancy)# exit`	`utd engine standard multi-tenancy`モードを終了します。 ポリシー設定が適用されます。これには数分かかる場合があります。この間は、`utd engine standard multi-tenancy`設定モードのコマンドはそれ以上入力できません。

ステップ 51	**exit**
	例：
	`Device(config)# exit`
	`Device#`	

ステップ 52	**show logging**
	例：
	`Device(config)# show logging`
	`..UTD MT configuration download has started`
	`..UTD MT configuration download has completed`	

ステップ 53	**interface** *sub-interface* 例： `Device(config)# interface GigabitEthernet4.101`	テナント（VRF）に使用するサブインターフェイスを指定します。
ステップ 54	**encapsulation dot1Q** *vlan-id* 例： `Device(config-if)# encapsulation dot1Q 101`	VLAN ID をサブインターフェイスに適用します。
ステップ 55	**ip vrf forwarding** *vrf-name* 例： `Device(config-if)# ip vrf forwarding vrf101`	VRF インスタンスをサブインターフェイスに関連付けます。
ステップ 56	**ip address** *ip-address subnet-mask* 例： `Device(config-if)# ip address 111.0.0.1 255.255.255.0`	VRF のサブインターフェイスの IP アドレスを指定します。
ステップ 57	**ip route** *ip-address subnet-mask sub-interface* 例： この例では、VRF のサブネット GigabitEthernet4.101 は、静的 IP アドレス 111.0.0.0 255.255.255.0 を使用してグローバル ルーティング テーブルにリンクされています。 `Device(config-if)# ip route 111.0.0.0 255.255.255.0 GigabitEthernet4.101`	（オプション）次の手順のこの `ip route` コマンドと `ip route vrf` コマンドはオプションです。VRF とグローバル ルーティング テーブル間の静的ルートを使用してルートリークを設定する場合にこれらの手順を使用できます。 これにより、VRF インターフェイスから VRF サブネットへの静的ルートが設定され、VRF サブネットにグローバル ルーティング テーブルからアクセ

	コマンドまたはアクション	目的
		スできるようになります。ルートリークの設定の詳細については、「MPLS または VPN ネットワークでのルートリーク」を参照してください。
ステップ 58	**ip route vrf** *vrf-name ip-address subnet-mask* **global** 例： `Device(config-if)# ip route vrf vrf101 0.0.0.0 0.0.0.0 5.2.1.1 global`	（オプション）この手順と前の手順は任意となります。VRF とグローバル ルーティング テーブル間の静的ルートを使用してルートリークを設定する場合は、次の手順を使用できます。ルートリークの設定の詳細については、「MPLS または VPN ネットワークでのルートリーク」を参照してください。 グローバル ルーティング テーブルへの静的 VRF のデフォルトルートを指定します。
ステップ 59	**utd enable**	（オプション）インターフェイス上で UTD を有効にします。このコマンドは、`all-interfaces` コマンドが設定されていない場合に使用できます（手順 44 内）。
ステップ 60	各テナント（VRF）のサブインターフェイスを設定するには、手順 53 ー 59 を繰り返します。	
ステップ 61	**exit**	インターフェイス設定モードを終了します。

Web フィルタリングおよび脅威検知（IPS）のプロファイルが適用されました。

設定例：統合脅威防御（UTD） のマルチテナント

この例は、2 つのテナントの UTD にマルチテナントを設定した後の一般的な実行設定を示しています。

（注） 次の例では、パラメータマップである `urlf-blacklist1` および `urlf-whitelist1` について説明します。これらのパラメータマップの設定は、例には示されていません。ブロックリストおよび承認済みリストのパラメータマップの詳細については、「インラインブロックページを使用した URL ベースの Web フィルタリングの設定」を参照してください。

```
utd multi-tenancy
utd engine standard multi-tenancy
 web-filter block page profile 1
  text "This page is blocked"
 web-filter block page profile 2
  text "This page is blocked"
 web-filter url profile 1
  alert all
  blacklist
   parameter-map regex urlf-blacklist1
  whitelist
```

```
  parameter-map regex urlf-whitelist1
 categories block
  social-network
  sports
 block page-profile 1
 log level error
web-filter url profile 2
 alert all
 blacklist
  parameter-map regex urlf-blacklist2
 categories block
  shopping
  news-and-media
  sports
  real-estate
  motor-vehicles
 block page-profile 2
 log level error
 reputation
  block-threshold low-risk
web-filter sourcedb 1
 logging level error
threat-inspection whitelist profile wh101
 signature id 101
threat-inspection profile 101
 threat protection
 policy security
 logging level debug
 whitelist profile wh101
threat-inspection profile 102
 threat detection
 policy security
 logging level debug
utd global
 logging host 172.27.58.211
 logging host 172.27.58.212
 logging host 172.27.56.97
 threat-inspection
  signature update server cisco username abc password
]RDCe[B\^KFI_LgQgCFeBEKWP^SWZMZMb]KKAAB
  signature update occur-at daily 0 0
 web-filter
  sourcedb 1
policy pol102
 vrf vrf102
 all-interfaces
 threat-inspection profile 102
 web-filter url profile 2
policy pol101
 vrf vrf101
 all-interfaces
 threat-inspection profile 101
 web-filter url profile 1
 fail close
```

統合脅威防御エンジンの標準設定の確認

次のコマンドを使用して、設定を確認します。

手順の概要

1. **enable**
2. **show utd multi-tenancy**
3. **show utd engine standard global**
4. **show utd engine standard status**
5. **show utd engine standard statistics**
6. **show utd engine standard statistics daq** [**dp** | **cp**]
7. **show utd engine standard statistics url-filtering** [**engine** | *no*]
8. **show utd engine standard statistics url-filtering vrf name** *vrf-name*
9. **show utd engine standard statistics internal**
10. **show utd engine standard logging event**
11. **show logging** | **include** CONFIG_DOWNLOAD
12. **show utd threat-inspection whitelist** [**profile** *profile-name*]
13. **show utd threat-inspection profile** *profile-name*
14. **show utd** [**policy** *profile-name*]
15. **show utd web-filter url** [**profile** *profile-name*]
16. **show utd web-filter block local-server** [**profile** *profile-name*]
17. **show utd web-filter sourcedb** [**profile** *profile-name*]
18. **show utd engine standard statistics daq dp** [**engine** *engine-num*] [**vrf** [**name** *vrf-name* | **global**]]
19. **show utd engine standard config threat-inspection whitelist** [**profile** *profile-name*]
20. **show utd engine standard config web-filter url profile** *profile-name*
21. **show utd engine standard config** [**vrf name** *vrf-name*]
22. **show utd engine standard config threat-inspection profile** *profile-name*
23. **show utd engine standard threat-inspection signature update status**
24. **show platform software qfp active feature utd config** [**vrf**[{**id** *vrf-id* | **name** *vrf-name* | **global** }]
25. **show platform software utd interfaces**
26. **show platform hardware qfp active feature utd config** [**vrf** {**id** *vrf-id* | **name** *vrf-name* | **global** }]
27. **show platform hardware qfp active feature utd stats** [**clear** | **divert** | **drop** | **general** | **summary**] [**vrf** {**id** *vrf-id* | **name** *vrf-name* | **global** }] [all] [**verbose**]
28. **show platform hardware qfp active feature utd stats summary** [**vrf name** *vrf-name* | **all**]
29. **show platform hardware qfp active feature utd stats drop all**

手順の詳細

ステップ 1 **enable**

例：

```
Device# enable
```

特権 EXEC モードを有効にします。パスワードを入力します（要求された場合）。

ステップ 2 **show utd multi-tenancy**

マルチテナントの現在のステータスを表示します。

例：

```
Device# show utd multi-tenancy
Multitenancy is enabled
```

ステップ 3　show utd engine standard global

UTD エンジン標準のグローバル設定を表示します。

例：

```
Device# show utd engine standard global
UTD Engine Standard Global: enabled
Threat-inspection: enabled
Web-filter: enabled
Logging:
```

ステップ 4　show utd engine standard status

UTD エンジンのステータスが緑色であることを確認します。

例：

```
Device# show utd eng standard status
Engine version       : 1.0.2_SV2983_XE_16_8

Profile              : Multi-tenancy
System memory        :
              Usage  : 3.50 %
              Status : Green
Number of engines    : 1

Engine        Running    CFT flows  Health     Reason
===============================================================
Engine(#1):   Yes        0          Green      None
===============================================================

Overall system status: Green

Signature update status:
========================
Current signature package version: 29.0.c
Last update status: Failed
Last successful update time: None
Last failed update time: Thu Jan 11 13:34:36 2018 PST
Last failed update reason: [Errno 113] No route to host
Next update scheduled at: None
Current status: Idle
```

ステップ 5　show utd engine standard statistics

例：

```
Device# show utd engine standard statistics
*************Engine #1*************
===============================================================================
Memory usage summary:
Total non-mmapped bytes (arena): 80125952
Bytes in mapped regions (hblkhd): 359546880
Total allocated space (uordblks): 68314032
Total free space (fordblks): 11811920
Topmost releasable block (keepcost): 112
===============================================================================
```

```
Packet I/O Totals:
Received: 49088
Analyzed: 49088 (100.000%)
Dropped: 0 ( 0.000%)
Filtered: 0 ( 0.000%)
Outstanding: 0 ( 0.000%)
Injected: 640
================================================================================
Breakdown by protocol (includes rebuilt packets):
Eth: 49394 (100.000%)

<output removed for brevity>

Total: 49394
================================================================================
Action Stats:
Alerts: 65 ( 0.132%)
Logged: 65 ( 0.132%)
Passed: 0 ( 0.000%)
```

ステップ 6　show utd engine standard statistics daq [dp | cp]

Snort DAQ 統計情報を表示します。

例：

```
Device# show utd engine standard statistics daq dp
IOS-XE DAQ Counters(Engine #1):
--------------------------------
Frames received 654101
Bytes received 549106120
RX frames released 654101
Packets after vPath decap 654101
Bytes after vPath decap 516510928
Packets before vPath encap 651686
Bytes before vPath encap 514800669
Frames transmitted 651686
Bytes transmitted 544447557

<output removed for brevity>
```

例：

```
Device# show utd engine standard statistics daq cp
IOS-XE DAQ CP Counters(Engine #1):
-----------------------------------
Packets received :16353210
Bytes received :1112018252
Packets transmitted :16353210
Bytes transmitted :1700733776
Memory allocation :16353212
Memory free :16353210
CFT API error :0
VPL API error :0
Internal error :0
External error :0
Memory error :0
Timer error :0
RX ring full 0
CFT full 0
sPath lib flow handle exhausted 0
Memory status changed to yellow :1
Memory status changed to red :0
```

```
Process restart notifications :0
```

ステップ 7 **show utd engine standard statistics url-filtering** [**engine** | *no*]

すべてのテナントのURL統計情報（ブロックリストのサイトのヒット数、許可リストのサイトのヒット数、カテゴリブロックとレピュテーションブロックによってブロックされたサイトの数を）を表示します。

例：

```
Device# show utd engine standard statistics url-filtering
UTM Preprocessor Statistics
---------------------------
URL Filter Requests Sent:               377226166       379846771       381117940
URL Filter Response Received:           377009606       379622845       380892658
Blacklist Hit Count:                    0               0               0
Whitelist Hit Count:                    0               0               0

Reputation Lookup Count:                376859139       379458008       380706804
Reputation Action Block:                0               0               0
Reputation Action Pass:                 307             280             102
Reputation Action Default Pass:         376858832       379457728       380706702
Reputation Score None:                  376858832       379457728       380706702
Reputation Score Out of Range:          0               0               0

Category Lookup Count:                  376859139       379458008       380706804
Category Action Block:                  0               0               0
Category Action Pass:                   307             280             102
Category Action Default Pass:           376858832       379457728       380706702
Category None:                          376858832       379457728       380706702


Device# show utd engine standard statistics url-filtering engine1
UTM Preprocessor Statistics
---------------------------------------------------
URL Filter Requests Sent:               377226166
URL Filter Response Received:           377009606
Blacklist Hit Count:                    0
Whitelist Hit Count:                    0

Reputation Lookup Count:                376859139
Reputation Action Block:                0
Reputation Action Pass:                 307
Reputation Action Default Pass:         376858832
Reputation Score None:                  376858832
Reputation Score Out of Range:          0

Category Lookup Count:                  376859139
Category Action Block:                  0
Category Action Pass:                   307
Category Action Default Pass:           376858832
Category None:                          376858832
```

ステップ 8 **show utd engine standard statistics url-filtering vrf name** *vrf-name*

追加パラメータの **vrf name** *vrf-name* を使用して、テナントごとの URL の統計情報を表示します。

例：

```
Device# show utd engine standard statistics url-filtering vrf name vrf101
UTM Preprocessor Statistics
```

```
--------------------------
URL Filter Requests Sent: 764
URL Filter Response Received: 764
Blacklist Hit Count: 3
Whitelist Hit Count: 44

Reputation Lookup Count: 764
Reputation Action Block: 0
Reputation Action Pass: 58
Reputation Action Default Pass: 706
Reputation Score None: 706
Reputation Score Out of Range: 0

Category Lookup Count: 764
Category Action Block: 5
Category Action Pass: 53
Category Action Default Pass: 706
Category None: 706
```

ステップ 9 show utd engine standard statistics internal

例：

```
Device# show utd engine standard statistics internal
*************Engine #1*************
================================================================================
Memory usage summary:
Total non-mmapped bytes (arena): 80125952
Bytes in mapped regions (hblkhd): 359546880
Total allocated space (uordblks): 68314032
Total free space (fordblks): 11811920
Topmost releasable block (keepcost): 112
================================================================================
Packet I/O Totals:
Received: 49088
Analyzed: 49088 (100.000%)
Dropped: 0 ( 0.000%)
Filtered: 0 ( 0.000%)
Outstanding: 0 ( 0.000%)
Injected: 640
================================================================================
Breakdown by protocol (includes rebuilt packets):
Eth: 49394 (100.000%)
VLAN: 49394 (100.000%)
IP4: 49394 (100.000%)
Frag: 0 ( 0.000%)
ICMP: 5 ( 0.010%)
UDP: 2195 ( 4.444%)
TCP: 47194 ( 95.546%)

<output removed for brevity>
```

ステップ 10 show utd engine standard logging event

VRF ごとにブロックリストまたは許可リストにあるアラートと URL を含むログを表示します。

例：

```
Device# show utd engine standard logging event

2017/08/04-16:01:49.205959 UTC [**] [Instance_ID: 1] [**] Drop [**]
UTD WebFilter Category/Reputation [**] [URL: www.cricinfo.com] ** [Category: Sports]
** [Reputation: 96] [VRF: vrf101] {TCP} 23.72.180.26:80 -> 111.0.0.254:53509
```

```
2017/08/04-16:02:12.253330 UTC [**] [Instance_ID: 1] [**] Pass [**]
 UTD WebFilter Whitelist [**] [URL: www.espn.go.com/m]
[VRF: vrf101] {TCP} 111.0.0.254:53511 -> 199.181.133.61:80
```

ステップ 11 **show logging** | **include** CONFIG_DOWNLOAD

例：

```
show# logging | include CONFIG_DOWNLOAD
Aug 23 11:34:21.250 PDT: %IOSXE_UTD-4-MT_CONFIG_DOWNLOAD: UTD MT configuration download has started
Aug 23 11:54:18.496 PDT: %IOSXE_UTD-4-MT_CONFIG_DOWNLOAD: UTD MT configuration download has
completed
```

ステップ 12 **show utd threat-inspection whitelist** [**profile** *profile-name*]

すべての許可リストのプロファイルまたは特定の許可リストのプロファイルを表示します。

例：

```
Device# show utd threat-inspection whitelist
Whitelist Profile: wh101
Signature ID: 101
```

例：

```
Device# show utd threat-inspection whitelist profile wh101
Whitelist Profile: wh101
Signature ID: 101
```

ステップ 13 **show utd threat-inspection profile** *profile-name*

プロファイル名で指定された脅威検知プロファイルの詳細を表示します。

例：

```
Device# show utd threat-inspection profile 101
Threat-inspection Profile: 101
Operational Mode: Intrusion Protection
Operational Policy: Security
Logging Level: debug
Whitelist Profile: wh101
```

ステップ 14 **show utd** [**policy** *profile-name*]

すべての UTD ポリシーまたは特定の UTD ポリシーを表示します。

例：

```
Device# show utd policy pol101
Policy name: pol101
VRF name: vrf101, VRF ID: 1
Global Inspection (across above VRFs): Enabled
Threat-inspection profile: 101
Web-filter URL profile: 1
Fail Policy: Fail-open
```

ステップ 15 **show utd web-filter url** [**profile** *profile-name*]

すべての URL プロファイルまたは特定のプロファイルを表示します。

例：

```
Device# show utd web-filter url profile 1
URL Profile: 1
Alert: all
Blacklist Parameter Map Regex: urlf-blacklist1
Whitelist Parameter Map Regex: urlf-whitelist1
Block Categories:
dating
sports
Block Page Profile 1
Log level error
reputation block-threshold high-risk
```

ステップ 16 **show utd web-filter block local-server** [**profile** *profile-name*]

すべてのブロックページのプロファイルまたは特定のブロックページのプロファイルを表示します。

例：

```
Device# show utd web-filter block local-server profile 2
Block Local Server Profile: 2
Content text: "Blocked by Web-Filter"
HTTP ports: 80
```

ステップ 17 **show utd web-filter sourcedb** [**profile** *profile-name*]

すべての sourcedb プロファイルまたは特定の sourcedb プロファイルを表示します。

例：

```
Device# show utd web-filter sourcedb
SourceDB Profile: 1
database update server interval hour 0 minute 0
Fail open
Log level: error
Proxy host port 0

SourceDB Profile: 2
database update server interval hour 0 minute 0
Fail open
Log level: error
Proxy host port 0
```

例：

```
Device# show utd web-filter sourcedb profile 1
SourceDB Profile: 1
database update server interval hour 0 minute 0
Fail open
Log level: error
Proxy host port 0
```

ステップ 18 **show utd engine standard statistics daq dp** [**engine** *engine-num*] [**vrf** [**name** *vrf-name* | **global**]]

すべての VRF または特定の VRF のサービスプレーンのデータ収集（DAQ：Data Acquistion）の統計情報を表示します。

例：

次の例は、VRF vrf101 のサービスプレーンのデータ収集の統計情報を示しています。

```
Device# show utd engine standard statistics daq dp vrf name vrf101
IOS-XE DAQ Counters(Engine #1):
--------------------------------
Frames received 374509
Bytes received 303136342
RX frames released 374509
Packets after vPath decap 374509
Bytes after vPath decap 284405526
Packets before vPath encap 372883
Bytes before vPath encap 283234522
Frames transmitted 372883
Bytes transmitted 300202270

Memory allocation 781856
Memory free 749636
Memory free via timer 29420
Merged packet buffer allocation 0
Merged packet buffer free 0

VPL buffer allocation 0
VPL buffer free 0
VPL buffer expand 0
VPL buffer merge 0
VPL buffer split 0
VPL packet incomplete 0

VPL API error 0
CFT API error 0
Internal error 52
External error 0
Memory error 0
Timer error 0

Kernel frames received 373590
Kernel frames dropped 0

FO cached via timer 0
Cached fo used 0
Cached fo freed 0
FO not found 0
CFT full packets 0
```

ステップ 19 **show utd engine standard config threat-inspection whitelist** [**profile** *profile-name*]

コンテナに保存されている脅威検知許可リストのプロファイルの詳細を表示します。

例：

```
Device# show utd engine standard config threat-inspection whitelist
UTD Engine Standard Configuration:

UTD threat-inspection whitelist profile table entries:
Whitelist profile: wh101
Entries: 1
```

ステップ 20 **show utd engine standard config web-filter url profile** *profile-name*

コンテナに保存されている Web フィルタのプロファイルの詳細を表示します。

例：

```
Device# show utd engine standard config web-filter url profile 1
UTD Engine Standard Configuration:

UTD web-filter profile table entries
Web-filter URL profile: 1
Whitelist:
www.espn.com
www.nbcsports.com
www.nfl.com
Blacklist:
www.cnn.com
Categories Action: Block
Categories:
Social Network
Sports
Block Profile: 1
Redirect URL: http://172.27.56.97/vrf101.html
Reputation Block Threshold: High risk
Alerts Enabled: Whitelist, Blacklist, Categories, Reputation
Debug level: Error
Conditional debug level: Error
```

ステップ 21 **show utd engine standard config** [**vrf name** *vrf-name*]

特定の VRF に関連付けられた UTD ポリシー、脅威検知プロファイル、および Web フィルタプロファイルの詳細を表示します。

例：

```
Device# show utd engine standard config vrf name vrf101
UTD Engine Standard Configuration:

UTD VRF table entries:
VRF: vrf101 (1)
Policy: pol101
Threat Profile: 101
Webfilter Profile: 1
```

ステップ 22 **show utd engine standard config threat-inspection profile** *profile-name*

特定の脅威検知プロファイルの詳細を表示します。

例：

```
Device# show utd engine standard config threat-inspection profile 101
UTD Engine Standard Configuration:

UTD threat-inspection profile table entries:
Threat profile: 101
Mode: Intrusion Prevention
Policy: Security
Logging level: Debug
Whitelist profile: wh101

Description:
Displays the details of a threat-inspection profile stored in the container.
```

ステップ 23 **show utd engine standard threat-inspection signature update status**

現在の署名パッケージのバージョン、以前の署名パッケージのバージョン、および最後のステータス更新の出力を表示します。

例：

```
Device# show utd engine standard threat-inspection signature update status
Current signature package version: 29.0.c
Current signature package name: default
Previous signature package version: None
----------------------------------------
Last update status: Failed
----------------------------------------
Last successful update time: None
Last successful update method: None
Last successful update server: None
Last successful update speed: None
----------------------------------------
Last failed update time: Thu Jan 11 13:34:36 2018 PST
Last failed update method: Manual
Last failed update server: http://172.27.57.252/UTD-STD-SIGNATURE-2983-1-S.pkg
Last failed update reason: [Errno 113] No route to host
----------------------------------------
Last attempted update time: Thu Jan 11 13:34:36 2018 PST
Last attempted update method: Manual
Last attempted update server: http://172.27.57.252/UTD-STD-SIGNATURE-2983-1-S.pkg
----------------------------------------
Total num of updates successful: 0
Num of attempts successful: 0
Num of attempts failed: 1
Total num of attempts: 1
----------------------------------------
Next update scheduled at: None
----------------------------------------
Current status: Idle
```

ステップ 24 **show platform software qfp active feature utd config [vrf[{id** *vrf-id* | **name** *vrf-name* | **global** }]

サービスノードの統計情報を表示します。VRF情報は、マルチテナントの場合にのみ表示できます。データプレーンUTD設定を表示します。次の例では、セキュリティコンテキスト情報が強調表示されています。

例：

```
Device# Global configuration
  NAT64: disabled
  SN threads: 12
  CFT inst_id 0 feat id 0 fo id 0 chunk id 4
  Context Id: 0, Name: Base Security Ctx
   Ctx Flags: (0xf0000)
        Engine: Standard
        SN Redirect Mode : Fail-close, Divert
        Threat-inspection: Enabled, Mode: IPS
        Domain Filtering : Not Enabled
        URL Filtering    : Not Enabled
   SN Health: Green
```

ステップ 25 **show platform software utd interfaces**

例：

```
Device# show platform software utd interfaces

UTD interfaces
All dataplane interfaces
```

ステップ 26 **show platform hardware qfp active feature utd config** [**vrf** {**id** *vrf-id* | **name** *vrf-name* | **global** }]

UTD データパスの設定とステータスを表示します。

例：

```
Device# show platform hardware qfp active feature utd config vrf name vrf101
Global configuration
  NAT64: disabled
  Drop pkts: disabled
  Multi-tenancy: enabled
  Data plane initialized: yes
  SN threads: 12
  CFT inst_id 0 feat id 1 fo id 1 chunk id 8
  SN Health: Green
```

ステップ 27 **show platform hardware qfp active feature utd stats** [**clear** | **divert** | **drop** | **general** | **summary**] [**vrf** {**id** *vrf-id* | **name** *vrf-name* | **global** }] [all] [**verbose**]

ゼロのカウントを含むデータプレーン UTD 統計情報を表示します。

clear：統計情報をクリアします

divert：AppNav リダイレクト統計情報を表示します

drop：ドロップ統計情報を表示します

general：一般統計情報を表示します

summary：サマリー統計情報を表示します

verbose：Verbose 統計情報を表示します

VRF 統計情報ごとの VRF 表示：VRF 情報は、マルチテナントが有効な場合にのみ入力できます。

id：VRF ID に関連付けられた統計情報を表示します

name：指定した名前の VRF に関連付けられた統計情報を表示します

global：グローバル VRF（つまり VRF ID が 0）に関連付けられている統計情報を表示します

例：

```
Device# show platform hardware qfp active feature utd stats

Summary Statistics:
TCP Connections Created 29893
UDP Connections Created 24402
ICMP Connections Created 796
Pkts dropped pkt 258
byt 66365
Pkts entered policy feature pkt 715602
byt 562095214
Pkts entered divert feature pkt 662014
byt 516226302
Pkts slow path pkt 55091
byt 4347864
Pkts Diverted pkt 662014
byt 516226302
```

```
Pkts Re-injected pkt 659094
byt 514305557

Would-Drop Statistics:

Service Node flagged flow for dropping 258

General Statistics:
Non Diverted Pkts to/from divert interface 1022186
Inspection skipped - UTD policy not applicable 1081563

<output removed for brevity>
```

例：

ステップ 28 **show platform hardware qfp active feature utd stats summary** [**vrf name** *vrf-name* | **all**]

show platform hardware qfp active feature utd stats コマンドのサマリーオプションから取得したすべての VRF または特定の VRF に関する情報を表示します。

例：

```
Device# show platform hardware qfp active feature utd stats vrf name vrf101
Security Context: Id:1 Name: 1 : vrf101

Summary Statistics:
TCP Connections Created 18428
UDP Connections Created 13737
ICMP Connections Created 503
Pkts dropped pkt 258
byt 66365
Pkts entered policy feature pkt 407148
byt 296496913
Pkts entered divert feature pkt 383176
byt 283158966
Pkts slow path pkt 32668
byt 2571632
Pkts Diverted pkt 383176
byt 283158966
Pkts Re-injected pkt 381016
byt 281761395


<output removed for brevity>
```

ステップ 29 **show platform hardware qfp active feature utd stats drop all**

show platform コマンドのドロップオプションから取得したすべての VRF からの情報を表示します。

例：

```
Device# show platform hardware qfp active feature utd stats drop all

Would-Drop Statistics:

No diversion interface                                                  0
No egress interface                                                     0
Inspection service down                                                 0
Could not find divert interface                                         0
Could not find divert fib                                               0
UTD FIB did not contain oce_chain                                       0
Invalid IP version                                                      0
```

```
IPS not supported                                                0
Re-inject Error                                                  0
Service Node flagged flow for dropping                        1225
Could not attach feature object                                  0
Could not allocate feature object                                0
Error getting feature object                                     0
Policy: could not create connection                              0
NAT64 Interface Look up Failed                                   0
Decaps: VPATH connection establishment error                     0
Decaps: VPATH could not find flow, no tuple                      0
Decaps: VPATH notification event error                           0
Decaps: Could not delete flow                                    0
Decaps: VPATH connection classification error                    0
Encaps: Error retrieving feature object                          0
Encaps: Flow not classified                                      0
Encaps: VPATH connection specification error                     0
Encaps: VPATH First packet meta-data failed                      0
Encaps: VPATH No memory for meta-data                            0
Encaps: VPATH Could not add TLV                                  0
Encaps: VPATH Could not fit TLV into memory                      0
Service Node Divert Failed                                       0
No feature object                                                0
Service Node not healthy                                       123
Could not allocate VRF meta-data                                 0
Could not allocate debug meta-data                               0
Packet was virtually fragmented (VFR)                            0
IPv6 Fragment                                                    0
IPv4 Fragment                                                    0
```

統合脅威防御（UTD）のマルチテナントに関するトラブルシューティング

トラフィックが転送されない

問題 トラフィックは転送されません。

考えられる原因 仮想サービスがアクティブになっていない可能性があります。

解決法 **show virtual-service list** コマンドを使用して、仮想サービスがアクティブになっているかどうかを確認します。次に、コマンドの出力例を示します。

```
Device# show virtual-service list

Virtual Service List:


Name Status Package Name
--------------------------------------------------------------------------------
snort Activated utdsnort.1_0_1_SV2982_XE_16_3.20160701_131509.ova
```

考えられる原因 指定されたインターフェイスでは、統合脅威防御（UTD）が有効になっていない可能性があります。

解決法 **show platform software utd global** コマンドを使用して、インターフェイスで UTD が有効になっているかどうかを確認します。

```
Device# show platform software utd global

UTD Global state
Engine               : Standard
Global Inspection    : Disabled
Operational Mode     : Intrusion Prevention
Fail Policy          : Fail-open
Container techonlogy : LXC
Redirect interface   : VirtualPortGroup1
UTD interfaces
GigabitEthernet0/0/0
```

考えられる原因 サービスノードが正常に動作していない可能性があります。

解決法 **show platform hardware qfp active feature utd config** コマンドを使用して、サービスノードの状態が緑色かどうかを確認します。

```
Device# show platform hardware qfp active feature utd config

Global configuration
NAT64: disabled
SN threads: 12
CFT inst_id 0 feat id 0 fo id 0 chunk id 4
Context Id: 0, Name: Base Security Ctx
Ctx Flags: (0x60000)
Engine: Standard
SN Redirect Mode : Fail-open, Divert
Threat-inspection: Enabled, Mode: IDS
Domain Filtering : Not Enabled
URL Filtering : Not Enabled
SN Health: Green
```

解決法 また、マルチテナントの場合は、**show platform hardware qfp active feature utd config vrf name** *vrf-name*コマンドを使用して、特定の VRF 関するサービスノードの正常性が緑色であるかどうかを確認できます。

```
Device# show platform hardware qfp active feature utd config vrf name vrf102
Global configuration
  NAT64: disabled
  Drop pkts: disabled
  Multi-tenancy: enabled
  Data plane initialized: yes
  SN threads: 12
  CFT inst_id 0 feat id 0 fo id 0 chunk id 4
  SN Health: Green
```

考えられる原因 Snort プロセスがアクティブになっていない可能性があります。

解決法 **show virtual-service detail** コマンドを使用して、Snortプロセス が稼働しているかどうかを確認します。

```
Device# show virtual-service detail

Virtual service UTDIPS detail
  State                : Activated
  Owner                : IOSd
  Package information
    Name               : utdsnort.1_0_1_SV2982_XE_16_3.20160701_131509.ova
    Path               : bootflash:/utdsnort.1_0_1_SV2982_XE_16_3.20160701_131509.ova
```

```
    Application
      Name               : UTD-Snort-Feature
      Installed version  : 1.0.1_SV2982_XE_16_3
      Description        : Unified Threat Defense
    Signing
      Key type           : Cisco development key
      Method             : SHA-1
    Licensing
      Name               : Not Available
      Version            : Not Available

  Detailed guest status

------------------------------------------------------------------------
Process               Status            Uptime           # of restarts
------------------------------------------------------------------------
climgr                 UP         0Y 0W 0D  0: 0:35        1
logger                 UP         0Y 0W 0D  0: 0: 4        0
snort_1                UP         0Y 0W 0D  0: 0: 4        0
Network stats:
eth0: RX  packets:43, TX  packets:6
eth1: RX  packets:8, TX  packets:6

Coredump file(s): lost+found

  Activated profile name: None
  Resource reservation
    Disk               : 736 MB
    Memory             : 1024 MB
    CPU                : 25% system CPU

  Attached devices
    Type              Name        Alias
    ---------------------------------------------
    NIC               ieobc_1     ieobc
    NIC               dp_1_0      net2
    NIC               dp_1_1      net3
    NIC               mgmt_1      mgmt
    Disk              _rootfs
    Disk              /opt/var
    Disk              /opt/var/c
    Serial/shell                  serial0
    Serial/aux                    serial1
    Serial/Syslog                 serial2
    Serial/Trace                  serial3
    Watchdog          watchdog-2

  Network interfaces
    MAC address             Attached to interface
    ----------------------------------------------------------
    54:0E:00:0B:0C:02       ieobc_1
    A4:4C:11:9E:13:8D       VirtualPortGroup0
    A4:4C:11:9E:13:8C       VirtualPortGroup1
    A4:4C:11:9E:13:8B       mgmt_1

  Guest interface
  ---
  Interface: eth2
  ip address: 48.0.0.2/24
Interface: eth1
  ip address: 47.0.0.2/24

  ---
```

```
  Guest routes
  ---
  Address/Mask                            Next Hop                              Intf.
-------------------------------------------------------------------------------------
0.0.0.0/0                               48.0.0.1                             eth2
0.0.0.0/0                               47.0.0.1                             eth1

  ---

  Resource admission (without profile) : passed
    Disk space    : 710MB
    Memory        : 1024MB
    CPU           : 25% system CPU
    VCPUs         : Not specified
```

考えられる原因 AppNav トンネルがアクティブになっていない可能性があります。

解決法 **show service-insertion type utd service-node-group** および **show service-insertion type utd service-context** コマンドを使用して、AppNav トンネルがアクティブになっているかどうかを確認します。

解決法 次に、**show service-insertion type utd service-node-group** コマンドの出力例を示します。

```
Device# show service-insertion type utd service-node-group

Service Node Group name : utd_sng_1
Service Context : utd/1
Member Service Node count : 1


Service Node (SN) : 30.30.30.2
Auto discovered : No
SN belongs to SNG : utd_sng_1
Current status of SN : Alive
Time current status was reached : Tue Jul 26 11:57:48 2016

Cluster protocol VPATH version : 1
Cluster protocol incarnation number : 1
Cluster protocol last sent sequence number : 1469514497
Cluster protocol last received sequence number: 1464
Cluster protocol last received ack number : 1469514496
```

解決法 次に、**show service-insertion type utd service-context** コマンドの出力例を示します。

```
Device# show service-insertion type utd service-context

Service Context : utd/1
Cluster protocol VPATH version : 1
Time service context was enabled : Tue Jul 26 11:57:47 2016
Current FSM state : Operational
Time FSM entered current state : Tue Jul 26 11:57:58 2016
Last FSM state : Converging
Time FSM entered last state : Tue Jul 26 11:57:47 2016
Cluster operational state : Operational

Stable AppNav controller View:
30.30.30.1

Stable SN View:
30.30.30.2
```

```
Current AppNav Controller View:
30.30.30.1

Current SN View:
30.30.30.2
```

考えられる原因 トラフィックのステータスのデータプレーンUTD統計情報を確認します。トラフィックが転送されない場合、転送および拒否されたパケットの数はゼロになります。数値がゼロ以外の場合、トラフィック転送が行われており、Snort センサーはデータプレーンにパケットを再送信しています。

解決法 **show platform hardware qfp active feature utd stats** コマンドを使用してトラフィックのステータスを確認します。

```
Device# show platform hardware qfp active feature utd stats

Security Context:    Id:0    Name: Base Security Ctx

Summary Statistics:
Active Connections                                                  29
TCP Connections Created                                         712910
UDP Connections Created                                             80
Pkts entered policy feature                         pkt       3537977
                                                    byt     273232057
Pkts entered divert feature                         pkt       3229148
                                                    byt     249344841
Pkts slow path                                      pkt        712990
                                                    byt      45391747
Pkts Diverted                                       pkt       3224752
                                                    byt     249103697
Pkts Re-injected                                    pkt       3224746
                                                    byt     249103373

….
```

解決法 また、マルチテナントの場合は、**show platform hardware qfp active feature utd stats vrf name** *vrf-name*コマンドを使用して、特定の VRF に関するトラフィックのステータスを確認できます。

```
Device# show platform hardware qfp active feature utd stats vrf name vrf 101

Security Context:    Id:1    Name: 1 : vrf101

Summary Statistics:
Active Connections                                                   2
TCP Connections Created                                          34032
UDP Connections Created                                          11448
ICMP Connections Created                                            80
Pkts dropped                                        pkt            626
                                                    byt         323842
Pkts entered policy feature                         pkt         995312
                                                    byt      813163885
Pkts entered divert feature                         pkt         639349
                                                    byt      420083106
Pkts slow path                                      pkt          45560
                                                    byt        7103132
Pkts Diverted                                       pkt         638841
                                                    byt      419901335
```

```
Pkts Re-injected                                    pkt              630642
                                                    byt           412139098
….
```

署名の更新が機能しない

問題　Cisco ボーダレスソフトウェア配布（BSD：Borderless Software Distribution）サーバからの署名更新が機能していません。

考えられる原因　さまざまな理由により署名の更新に失敗した可能性があります。最後に署名の更新に失敗した理由を確認します。

解決法　**show utd engine standard threat-inspection signature update status** コマンドを使用して、最後に署名の更新に失敗した理由を表示します。

```
Device# show utd eng standard threat-inspection signature update status
Current signature package version: 29.0.c
Current signature package name: default
Previous signature package version: None
----------------------------------------
Last update status: Failed
----------------------------------------
Last successful update time: None
Last successful update method: None
Last successful update server: None
Last successful update speed: None
----------------------------------------
Last failed update time: Thu Jan 11 13:34:36 2018 PST
Last failed update method: Manual
Last failed update server: http://172.27.57.252/UTD-STD-SIGNATURE-2983-1-S.pkg
Last failed update reason: [Errno 113] No route to host
----------------------------------------
Last attempted update time: Thu Jan 11 13:34:36 2018 PST
Last attempted update method: Manual
Last attempted update server: http://172.27.57.252/UTD-STD-SIGNATURE-2983-1-S.pkg
----------------------------------------
Total num of updates successful: 0
Num of attempts successful: 0
Num of attempts failed: 1
Total num of attempts: 1
----------------------------------------
Next update scheduled at: None
----------------------------------------
Current status: Idle
```

考えられる原因　ドメインネームシステム（DNS）が正しく設定されていません。

解決法　**show running-config | i name-server** コマンドを使用して、ネームサーバの詳細を表示します。

```
Device#  show run | i name-server

ip name-server 10.104.49.223
```

考えられる原因　システムエラー：ユーザ名とパスワードの組み合わせの処理に失敗しました。

解決法 署名パッケージのダウンロードに正しい認証情報を使用したことを確認します。

ローカルサーバからの署名の更新が機能しない

問題 ローカルサーバからの署名の更新が機能しない。

考えられる原因 最後の失敗の理由：無効なスキーム — HTTP または HTTPS のみに対応します。

解決法 ローカルダウンロード方式として HTTP またはセキュア HTTP（HTTPS）が指定されていることを確認します。

考えられる原因 最後の失敗の理由：名前またはサービスが不明です。

解決法 ローカルサーバに指定されたホスト名または IP アドレスが正しいことを確認します。

考えられる原因 最後の失敗の理由：認証情報が入力されていません。

解決法 ローカル HTTP または HTTPS サーバの認証情報が入力されていることを確認します。

考えられる原因 最後の失敗の理由：ファイルが見つかりません。

解決法 入力した署名ファイル名または URL が正しいことを確認します。

考えられる原因 最後の失敗の理由：ダウンロードが破損しています。

解決法

- 以前の署名のダウンロード時に署名更新の再試行でエラーが発生していないかどうかを確認します。
- 正しい署名パッケージが使用可能であることを確認します。

IOSd Syslog へのロギングが機能しない

問題 IOSd syslog へのロギングが機能しない。

考えられる原因 syslog へのロギングは、統合脅威防御（UTD）の設定では設定できません。

解決法 UTD 設定を表示し、syslog へのロギングが設定されていることを確認するには、**show utd engine standard config** コマンドを使用します。

```
Device# show utd engine standard config

UTD Engine Standard Configutation:
  Operation Mode : Intrusion Prevention
  Policy         : Security

Signature Update:
  Server    : cisco
  User Name : ccouser
  Password  : YEX^SH\fhdOeEGaOBIQAIcOVLgaVGf
  Occurs-at : weekly ;  Days:0 ; Hour: 23; Minute: 50

Logging:
  Server    :   IOS Syslog; 10.104.49.223
  Level     : debug
```

```
Whitelist Signature IDs:
  28878
```

解決法 UTD エンジンのイベントログを表示するには、次の **show utd engine standard logging events** コマンドを使用します。

```
Device# show utd engine standard logging events

2016/06/13-14:32:09.524475 IST [**] [Instance_ID: 1] [**] Drop [**] [1:30561:1]
BLACKLIST DNS request for known malware domain domai.ddns2.biz -
Win.Trojan.Beebone [**] [Classification: A Network Trojan was Detected]
[Priority: 1] [VRF_ID: 2] {UDP} 11.1.1.10:58016 -> 21.1.1.10:53
2016/06/13-14:32:21.524988 IST [**] [Instance_ID: 1] [**] Drop [**] [1:30561:1]
BLACKLIST DNS request for known malware domain domai.ddns2.biz -
Win.Trojan.Beebone [**] [Classification: A Network Trojan was Detected] [Priority: 1]
[VRF_ID: 2] {UDP} a000:0:0:0:0:0:0:10:59964 -> b000:0:0:0:0:0:0:10:53
```

外部サーバへのロギングが機能しない

問題 外部サーバへのロギングが機能していません。

考えられる原因 外部サーバで Syslog が実行されていない可能性があります。

解決法 syslog サーバが外部サーバで実行されているかどうかを確認します。ステータスを表示するには、外部サーバで次のコマンドを設定します。

```
ps -eaf | grep syslog

 root 2073 1 0 Apr12 ? 00:00:02 syslogd -r -m
```

考えられる原因 統合脅威防御（UTD）の Linux コンテナ（LXC：Linux Container）と外部サーバ間の接続が失われている可能性があります。

解決法 管理インターフェイスから外部 syslog サーバへの接続を確認します。

UTD 条件付きデバッグ

条件付きデバッグは、Unified Threat Defense のマルチテナントに対応しています。条件付きデバッグの設定方法の詳細については、以下を参照してください。

http://www.cisco.com/c/en/us/td/docs/routers/asr1000/troubleshooting/guide/Tbl-shooting-xe-3s-asr-1000-book.html#task_AC969BB06B414DCBBDEF7ADD29EF8131

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。