ソフトウェアエージェント

Secure Workload ソフトウェアエージェントは、ワークロードにインストールする軽量のソフトウェアです。その目的は次のとおりです。

- システムで実行されているネットワークインターフェイスやアクティブなプロセスなどのホスト情報を収集します。
- ネットワークフロー情報をモニターおよび収集します。
- インストールされているホストにファイアウォールルールを設定して、（有効になっている場合）セキュリティポリシーを適用します。

エージェントは、インターフェイスアドレスが変更されると、Secure Workloadインベントリを自動的に更新します。

エンドユーザー（従業員）のコンピューターにエージェントをインストールする必要はありません。

ソフトウェアエージェントの展開

（注） 自動ロールマッピングを使用して LDAP/AD アカウントからダウンロードしたインストーラスクリプトは、ユーザーがログアウトするとすぐに失敗します。インストーラスクリプトにクラスタへの連続アクセスを許可するには、ユーザーに対して [ローカル認証を使用（Use Local Authentication）] を有効にすることを推奨します。「Use Local Authentication」オプション

展開が成功すると、エージェントが実行されているホストに固有の一連のパラメータに基づき、Secure Workload クラスタによってエージェントに固有の ID が割り当てられます。ホスト名と BIOS UUID は設定の一部であるため、次の問題が発生する可能性があります。

1. BIOS UUID とホスト名を保持したまま仮想マシンを複製すると、登録に失敗します。登録の失敗は、VDI インスタントクローニングの実行時などにも発生する可能性があります。これは、Secure Workload クラスタに、同じパラメータセットを使用して登録されたソフトウェアエージェントがすでにあるためです。ユーザーは、OpenAPI を使用して登録済みのエージェントを削除できます。場合によっては、起動時に設定された重複 BIOS UUID が、遅延後に VMware によって変更されます。エージェントの登録は、Cisco Secure Workload サービスが再起動されない限り回復しません。
2. ホスト名を変更してホストを再起動すると、エージェントの新しい ID が生成される可能性があります。冗長または古いエージェントエントリは、一定時間が経過すると非アクティブとしてマークされます。ソフトウェアエージェントのトラブルシューティング（106 ページ）を参照してください。

サポートされているプラットフォームと要件

サポートされているプラットフォームとソフトウェアエージェントの追加要件については、以下を参照してください。

- ご使用のリリースのリリースノートは、https://www.cisco.com/c/en/us/support/security/tetration/products-release-notes-list.html から入手できます。
- Secure Workload Web ポータルのエージェント インストール ウィザード：左側のナビゲーションバーで**[管理（Manage）]** > **[エージェント（Agents）]**を選択し、[インストーラ（Installer）] タブをクリックします。インストール方法、プラットフォーム、およびエージェントタイプ（該当する場合）を選択して、サポートされているプラットフォームのバージョンを確認します。
- https://www.cisco.com/go/secure-workload/requirements/agents から入手可能なサポートマトリックス。このリソースには、複数の追加の依存関係が含まれています。すべての列が表示されていることを確認してください。
- 以下の各プラットフォームおよびエージェントタイプのセクションの追加要件。

Linux エージェント：優れた可視性と適用

要件および前提条件

- 「サポートされているプラットフォームと要件」を参照してください。
- サービスをインストールして実行するには、ルート権限が必要です。
- エージェントおよびログファイルのストレージ要件：IBM Z の場合は 500MB。それ以外の場合は 1GB。
- ホストを監視しているセキュリティアプリケーションでセキュリティの除外を構成することにより、他のセキュリティアプリケーションがエージェントのインストールやエージェントのアクティビティをブロックしないようにします。「セキュリティの除外」を参照してください。
- エージェントがインストールされているホストには特別なユーザー **tet-sensor** が作成されることに注意してください。ホストで PAM/SELinux が構成されている場合は、tet-sensor プロセスの実行やコレクタへの接続など、**tet-sensor** ユーザーに適切な権限を付与する必要があります。代替のインストールディレクトリがあり、SELinux が設定されている場合は、代替のインストールディレクトリで実行が許可されていることを確認してください。
- エージェントが自動インストール（インストーラスクリプト）メソッドを使用してインストールされている場合は、unzip コマンドを使用できる必要があります。

エージェントのインストール

優れた可視性や適用を実現するために Linux エージェントをインストールするには、次の 2 つの方法があります。

- インストーラを使用したエージェントの自動インストール
- 従来のパッケージインストーラを使用したエージェントの手動インストール

インストーラを使用した自動インストール（Linux）

インストーラスクリプトは、Linux プラットフォームで優れた可視性と適用のエージェントを展開する場合に推奨される方法です。

デフォルトでは、インストールされたエージェントは優れた可視性と適用の両方をサポートしています。デフォルトでは適用は無効になっていますが、Secure Workload ユーザーインターフェイスを使用して簡単に有効にすることができます。

エージェントをインストールするには、次の手順を実行します。

ステップ 1 左側のナビゲーションバーで、**[管理（Manage）]** > **[エージェント（Agents）]** をクリックします。

ステップ 2 [インストーラ（Installer）] タブをクリックします。

ステップ 3 [インストーラを使用してエージェントを自動インストール（Auto-Install Agent using an Installer）] ワークフローを選択し、[次へ（Next）] をクリックします。

ステップ 4 エージェントをインストールするテナントを選択します。

（注） Secure Workload SaaS クラスタでは、テナントの選択は必要ありません。

ステップ 5 CMDB ラベルを選択し、関連付けられた値を入力して、それをエージェントインストーラに添付します（オプションの選択）。インストールされたエージェントがこのホストで見つかった新しい IP アドレスをクラスタに報告すると、このホストによって報告された IP に割り当てられているアップロード済みの別の CMDB ラベルとともに、ここで選択されたインストーラ CMDB ラベルが新しい IP に自動的に割り当てられます。アップロードされた CMDB ラベルとインストーラ CMDB ラベルの間で競合が発生した場合：

- 正確な IP アドレスに割り当てられたラベルは、サブネットに割り当てられたラベルよりも優先されます。
- 正確な IP アドレスに割り当てられた既存のラベルは、インストーラの CMDB ラベルよりも優先されます。

ステップ 6 [プラットフォーム（Platform）] セクションで [Linux] を選択します。

ステップ 7 ネットワークに HTTP プロキシが必要な場合は [はい（Yes）] を選択し、有効なプロキシ URL を入力します。それ以外の場合は [いいえ（No）] を選択します。

ステップ 8 [インストーラの有効期限（Installer expiration）] セクションで、利用可能なオプションから 1 つを選択します。

- 有効期限なし：インストーラスクリプトは何回も使用できます。
- 1 回のみ：インストーラスクリプトは 1 回のみ使用できます。
- 時間制限：インストーラスクリプトを使用できる日数を設定できます。
- 展開数：インストーラスクリプトを使用できる回数を設定できます。

ステップ 9 [Download Installer（インストーラのダウンロード）]をクリックし、ファイルをローカルディスクに保存します。インストーラスクリプトがローカルに保存されたら、[次へ（Next）] をクリックします。

ステップ 10 インストーラ シェル スクリプトを展開するためにすべての Linux ホストにコピーします。

ステップ 11 コマンド `chmod u+x tetration_installer_default_sensor_linux.sh` を実行し、スクリプトの実行権限を許可します。

（注） スクリプト名は、エージェントのタイプと範囲によって異なる場合があります。

ステップ 12 エージェントをインストールするには、root 権限でコマンド `./tetration_installer_default_sensor_linux.sh` を実行します。以下のスクリプトの使用方法の詳細で指定されているように、事前チェックを実行することをお勧めします。

（注） エージェントがテナントにすでにインストールされている場合、インストーラスクリプトは機能しません。

図 *1:* ソフトウェア エージェント インストーラ スクリプトのダウンロードページ（オンプレミス）

図 ***2:*** ソフトウェア エージェント インストーラ スクリプトのダウンロードページ（***Saas***）

インストーラスクリプトの使用方法は、次のとおりです。

$ bash tetration_linux_installer.sh [–pre-check] [–skip-pre-check=<option>] [–no-install] [–logfile=<filename>] [–proxy=<proxy_string>] [–no-proxy] [–help] [–version] [–sensor- version=<version_info>] [–ls] [–file=<filename>] [–save=<filename>] [–new] [–reinstall] [–unpriv-user] [–force-upgrade] [–upgrade-local] [–upgrade-by-uuid=<filename>] [– basedir=<basedir>] [–logbasedir=<logbdir>] [–visibility]

–pre-check：事前チェックのみを実行します

–skip-pre-check=<option>：指定されたオプションで事前インストールチェックをスキップします。有効なオプションには「all」、「ipv6」および「enforcement」が含まれます。たとえば、'–skip-pre-check=all' は、すべての事前インストールチェックをスキップします。すべての事前チェックはデフォルトで実行されます

–no-install：システムでのセンサーパッケージのダウンロードおよびインストールは行いません

–logfile <filename>：<filename> で指定されたファイルにログを書き込みます。

–proxy <proxy_string>：HTTPS_PROXY の値を設定します。クラスタとの通信にプロキシが必要な場合は、これを使用します。文字列は http://<proxy>:<port> の形式にする必要があります

–no-proxy：システム全体のプロキシをバイパスします。 –proxy フラグが指定されている場合、このフラグは無視されます

–help：このヘルプを印刷します

–version：現在のスクリプトのバージョンを印刷します。

–sensor-version <version_info>：センサーのバージョン（たとえば「–sensor- version=3.4.1.0」）を選択します。このフラグが提供されていない場合、デフォルトで最新バージョンをダウンロードします

–ls：システムで使用可能なすべてのセンサーバージョンを一覧表示します（3.1 以前のパッケージは表示しません）。パッケージはダウンロードしません

–file <filename>：クラスタからダウンロードする代わりに、センサーをインストールするためのローカル zip ファイルを提供します

–save <filename>：zipファイルをダウンロードして <filename> として保存します

–new：以前にインストールされたセンサーをすべて削除します。新しい登録を成功させるには、以前のセンサー ID をクラスタから削除する必要があります

–reinstall：センサーを再インストールし、クラスタと同じ ID を保持します。このフラグは、–new よりも優先されます

–unpriv-user=<username>：unpriv プロセスで tet-sensor の代わりに <username> を使用します

–force-upgrade：–sensor-version フラグで指定されたバージョンへのセンサーのアップグレードを強制します（例：'–sensor-version=3.4.1.0 –force-upgrade'）– sensor-version フラグが指定されていない場合、デフォルトで最新バージョンを適用します

–upgrade-local：–sensor-version フラグで指定されたバージョンへのローカルセンサーのアップグレードをトリガーします（例：'–sensor-version=3.4.1.0 –upgrade-local'）次の場合、デフォルトで最新バージョンを適用します

–sensor-version フラグは提供されませんでした

–upgrade-by-uuid=<filename>：<filename> に uuid がリストされているトリガーセンサーを –sensor-version フラグで指定されたバージョンにアップグレードします（例：‘–sensor-version=3.4.1.0 –upgrade-by-uuid=/usr/local/tet/sensor_id’）– sensor-version フラグが指定されていない場合、デフォルトで最新バージョンを適用します

–baseir=<base_dir>：/usr/local を使用する代わりに <base_dir> を使用してエージェントをインストールします。フルパスは <base_dir>/tetration となります

–logbasedir=<log_base_dir>：/usr/local/tet/log にログインする代わりに

<log_base_dir> を使用します。フルパスは <log_base_dir>/tetration です

–visibility：優れた可視性エージェントのみをインストールします。 –reinstall は、以前にインストールされたエージェントタイプがエンフォーサである場合、このフラグを上書きします

（注）
- Ubuntu は現在、native.deb パッケージを使用しています。新規インストールおよび再インストールの場合はこのパッケージタイプに切り替わり、以前のバージョンからのアップグレードの場合は rpm パッケージのままになります。
- Ubuntu .deb パッケージは /opt/cisco/tetration にインストールされています。
- .deb パッケージの再配置がサポートされていないため、Ubuntu では –basedir オプションはサポートされていません。

従来のパッケージインストーラを使用した手動インストール（Linux）

このセクションでは、エージェントイメージをダウンロードして Linux ホストにインストールする方法について説明します。

ほとんどの場合、手動でインストールする特別な理由がない限り、より単純な自動インストール方法（前述）を使用する必要があります。

前提条件：

デフォルトのテナントの下にインストールしていないときの SaaS および複数のテナントがあるオンプレミスクラスタの場合：

手動でインストールする前に、user.cfg ファイルで ACTIVATION_KEY と HTTPS_PROXY を設定する必要があります。詳細については、「（手動インストールのみ）ユーザー構成ファイルの更新」を参照してください。

ステップ 1 左側のナビゲーションバーで、**[管理（Manage）]** > **[エージェント（Agents）]** をクリックします。

ステップ 2 [インストーラ（Installer）] タブをクリックします。

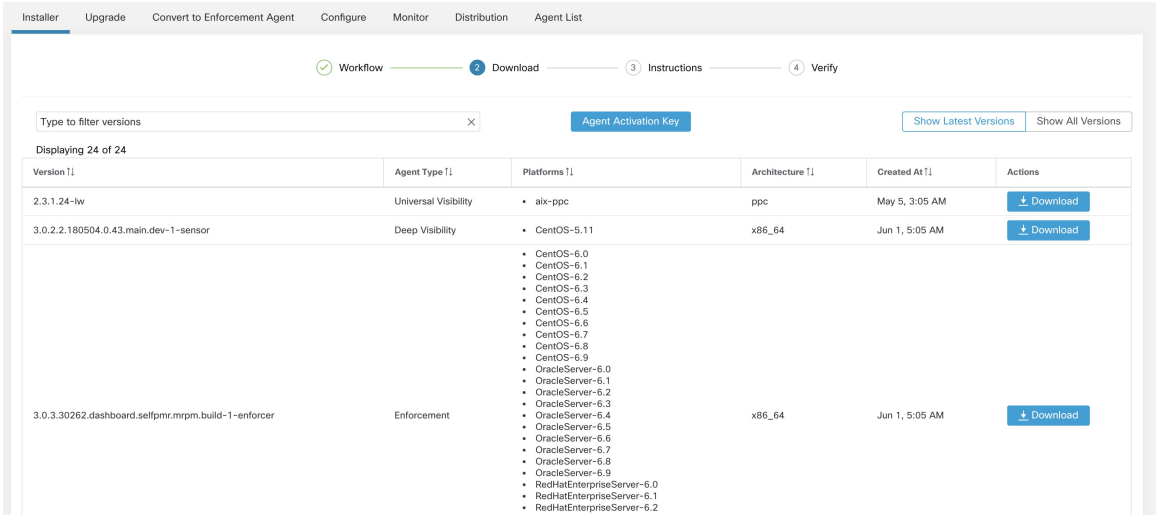
ステップ 3 [従来のパッケージインストーラを使用した手動インストール（Manual Install using classic packaged installers）] ワークフローを選択し、[次へ（Next）] をクリックします。

ステップ 4 適切なバージョン/プラットフォーム/アーキテクチャ/エージェントタイプを見つけて、[ダウンロード（Download）] ボタンをクリックします。

ステップ 5 展開するすべての Linux ホストに rpm パッケージをコピーし、ルート権限で rpm コマンドを実行します。

（注） **エージェントがすでにインストールされている場合は、**再インストールしないでください。エージェントを新しいバージョンにアップグレードする必要がある場合は、「ソフトウェアエージェントのアップグレード」で説明されているアップグレードプロセスに従ってください。

図 *3:* ソフトウェア エージェント バンドルのダウンロードページ



RHEL/CentOS/Oracle プラットフォームの場合：

1. `rpm -ivh <rpm_filename>` コマンドを実行します。

Ubuntu プラットフォームの場合：

1. 最初に `rpm -qpR <rpm_filename>` コマンドを実行して依存関係リストを取得し、すべての依存関係が満たされていることを確認します。
2. 次に、「–nodeps」オプション：`rpm -ivh \\--nodeps <rpm filename>` を指定してインストールします。

エージェントがインストールされていることの確認

ステップ 1 コマンド `sudo rpm -q tet-sensor` を実行します。

ステップ 2 エントリが 1 つあることを確認します。

（注） 指定された出力は、プラットフォームとアーキテクチャによって異なる場合があります。

```
$ sudo rpm -q tet-sensor
tet-sensor-3.1.1.50-1.el6.x86_64
```

Windows エージェント - 優れた可視性および適用

要件および前提条件

- 「サポートされているプラットフォームと要件」を参照してください。
- 管理者権限（インストールとサービス実行の両方）
- Npcap がインストール済みである必要があります。Npcap ドライバがまだインストールされていない場合、推奨される Npcap バージョンが、サービスの開始から 2 分後にエージェントによってサイレントにインストールされます。Npcap のバージョン情報については、「https://www.cisco.com/go/secure-workload/requirements/agents」を参照してください。
- エージェントとログファイルのストレージ要件：1 GB。
- 必要な Windows サービス：Windows ホストのセキュリティが強化されている場合や、Microsoft から出荷されたときのデフォルト構成から逸脱している場合は、エージェントのインストールを正常に行うために必要な一部の Windows サービスが無効になっている可能性があります。「必要な Windows サービス」を参照してください。
- ホストを監視しているセキュリティアプリケーションでセキュリティの除外を設定することにより、他のセキュリティアプリケーションがエージェントのインストールやエージェントのアクティビティをブロックしないようにします。「セキュリティの除外」を参照してください。

エージェントのインストール

優れた可視性または適用の目的で Windows プラットフォームにエージェントをインストールするには、2 つの方法があります。

- インストーラを使用したエージェントの自動インストール
- クラシック パッケージ インストーラを使用したエージェントの手動インストール

ゴールデンイメージを使用したインストールも可能です。「VDI インスタンスまたは VM テンプレートへのエージェントの展開（Windows）」を参照してください。

インストーラを使用したエージェントの自動インストール（Windows）

これは、Windows プラットフォームで優れた可視性または適用エージェントを展開するために推奨される方法です。「インストーラスクリプト」と呼ばれることもあります。

デフォルトでは、インストールされたエージェントは優れた可視性と適用の両方をサポートしています。デフォルトでは適用は無効になっていますが、Secure Workload ユーザーインターフェイスを使用して簡単に有効にすることができます。

ステップ 1 左側のナビゲーションバーで、**[管理（Manage）]** > **[エージェント（Agents）]** をクリックします。

ステップ 2 [インストーラ（Installer）] タブをクリックします。

ステップ 3 [インストーラを使用して自動インストール（Auto-Install using Installers）] ワークフローを選択し、[次へ（Next）] をクリックします。

ステップ 4 エージェントをインストールするテナントを選択します。

（注） Secure Workload SaaS クラスタでは、テナントの選択は必要ありません。

ステップ 5 CMDB ラベルを選択し、関連付けられた値を入力して、それをエージェントインストーラに添付します（オプションの選択）。インストールされたエージェントがこのホストで見つかった新しい IP アドレスをクラスタに報告すると、このホストによって報告された IP に割り当てられているアップロード済みの別の CMDB ラベルとともに、ここで選択されたインストーラ CMDB ラベルが新しい IP に自動的に割り当てられます。アップロードされた CMDB ラベルとインストーラ CMDB ラベルの間で競合が発生した場合：

- 正確な IP アドレスに割り当てられたラベルは、サブネットに割り当てられたラベルよりも優先されます。
- 正確な IP アドレスに割り当てられた既存のラベルは、インストーラの CMDB ラベルよりも優先されます。

ステップ 6 プラットフォームとして [Windows] を選択します。

ステップ 7 ネットワークに HTTP プロキシが必要な場合は [はい（Yes）] を選択し、有効なプロキシ URL を入力します。それ以外の場合は [いいえ（No）] を選択します。

ステップ 8 [インストーラの有効期限（Installer expiration）] セクションで、利用可能なオプションから 1 つを選択します。

- 有効期限なし：インストーラスクリプトは何回も使用できます。
- 1 回のみ：インストーラスクリプトは 1 回のみ使用できます。
- 時間制限：インストーラスクリプトを使用できる日数を設定できます。
- 展開数：インストーラスクリプトを使用できる回数を設定できます。

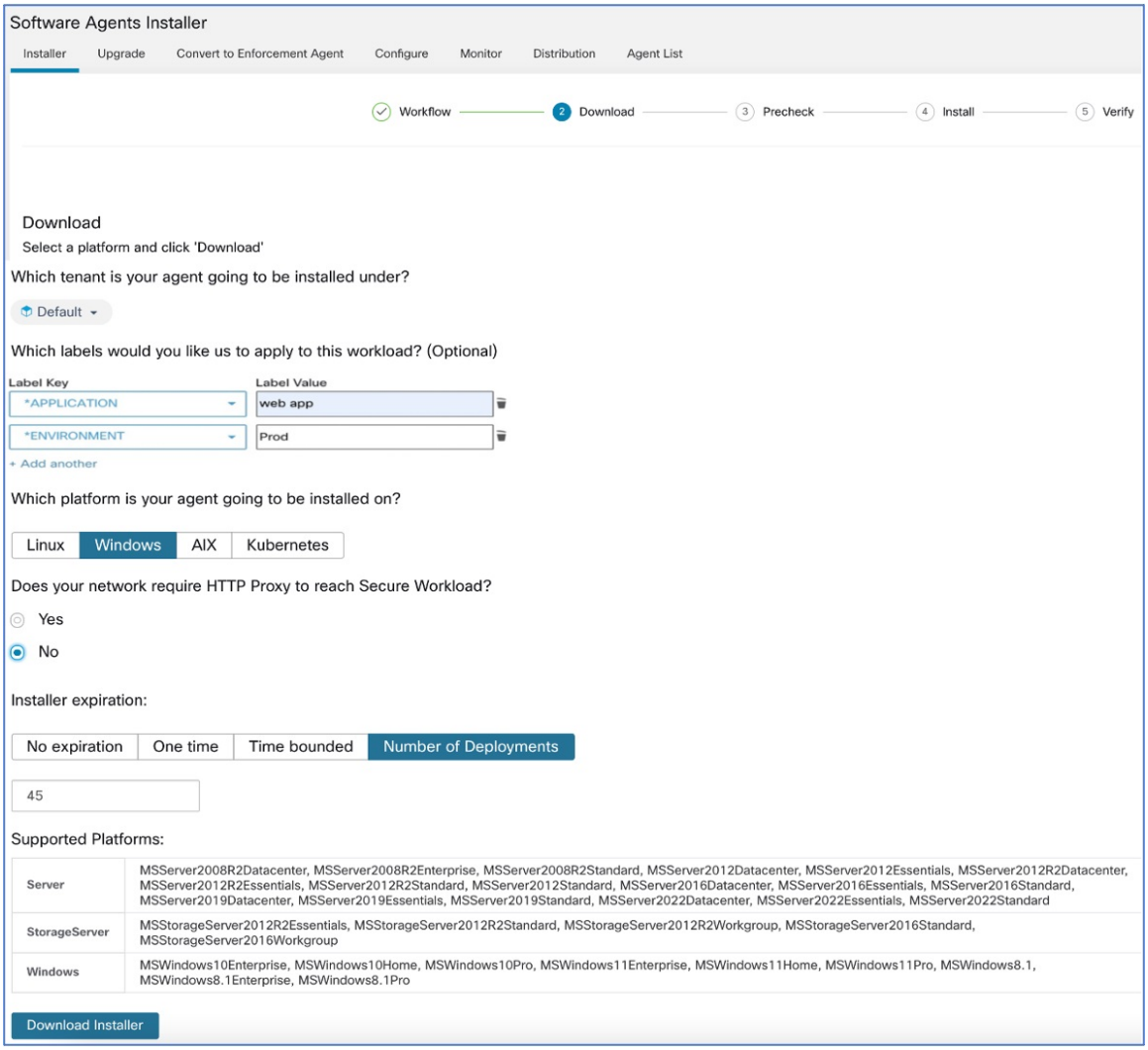
ステップ 9 [インストーラのダウンロード（Download Installer）] をクリックし、ファイルをローカルディスクに保存します。インストーラスクリプトがローカルに保存されたら、[次へ（Next）] をクリックします。

ステップ 10 インストーラの PowerShell スクリプトを展開用のすべての Windows ホストにコピーし、管理者権限でスクリプトを実行します。

以下のスクリプトの使用方法の詳細で指定されているように、事前チェックを実行することをお勧めします。

（注） システムの設定によっては、最初にコマンド `Unblock-File` の実行が必要な場合があります。さらに、エージェントがすでにインストールされている場合、スクリプトは続行されません。

図 *4:* ソフトウェア エージェント インストーラ スクリプトのダウンロードページ（オンプレミス）



インストーラスクリプトの使用方法は、次のとおりです。

powershell -File tetration_windows_installer.ps1 [-preCheck] [-skipPreCheck <Option>] [-noInstall] [-logFile <FileName>] [-proxy <ProxyString>] [-noProxy] [-help] [-version] [-sensorVersion <VersionInfo>] [-ls] [-file <FileName>] [-save <FileName>] [-new] [-reinstall] [-npcap] [-forceUpgrade] [-upgradeLocal] [-upgradeByUUID <FileName>] [-visibility] [- goldenImage] [-installfolder <install Path>]

–pre-check：事前チェックのみを実行します

–skip-pre-check= <Option>：指定されたオプションで事前インストールチェックをスキップします。有効なオプションには「all」、「ipv6」および「enforcement」が含まれます。（たとえば、‘-skipPreCheck all’は、すべての事前インストールチェックをスキップします。）すべての事前チェックはデフォルトで実行されます

–no-install：システムでのセンサーパッケージのダウンロードおよびインストールは行いません

-logFile <FileName>：<FileName> で指定されたファイルにログを書き込みます

-proxy <ProxyString>：HTTPS_PROXY の値を設定します。クラスタとの通信にプロキシが必要な場合は、これを使用します。文字列は http:

//<proxy>:<port> の形式にする必要があります

-noProxy：システム全体のプロキシをバイパスします。-proxy フラグが指定されている場合、このフラグは無視されます

-help：このヘルプを印刷します

-version：現在のスクリプトのバージョンを印刷します

-sensorVersion <VersionInfo>：センサーのバージョン（例：'-sensorVersion 3.4.1.0.win64'）を選択します。このフラグが提供されていない場合、デフォルトで最新バージョンをダウンロードします

-ls：システムで使用可能なすべてのセンサーバージョンを一覧表示します（3.1 以前のパッケージは表示しません）。パッケージはダウンロードされません

-file <filename>：クラスタからダウンロードする代わりに、センサーをインストールするためのローカル zip ファイルを提供します

-save <filename>：zipファイルをダウンロードして <filename> として保存します

-new：以前にインストールされたセンサーをすべて削除します。新しい登録を成功させるには、以前のセンサー ID をクラスタから削除する必要があります

-reinstall：センサーを再インストールし、クラスタと同じ ID を保持します。このフラグは、-new よりも優先されます

-npcap：既存の npcap を上書きします

-forceUpgrade：-sensorVersion フラグで指定されたバージョンへセンサーのアップグレードを強制します（例：'-sensorVersion 3.4.1.0.win64 -forceUpgrade'）

-sensorVersion フラグが指定されていない場合、デフォルトで最新バージョンを適用します

-upgradeLocal：-sensorVersion フラグで指定されたバージョンへのローカルセンサーのアップグレードをトリガーします（例：'-sensorVersion 3.4.1.0.win64 -upgradeLocal'）-sensorVersion フラグが指定されていない場合、デフォルトで最新バージョンを適用します。

-upgradeByUUID <FileName>：<FileName> に uuid がリストされているトリガーセンサーを -sensorVersion フラグで指定されたバージョンにアップグレードします（例：'-sensorVersion 3.4.1.0.win64 -upgradeByUUID "C:\Program Files\Cisco Tetration\sensor_id"'）-sensorVersion フラグが指定されていない場合、デフォルトで最新バージョンを適用します。

-visibility：優れた可視性エージェントのみをインストールします。-reinstall は、以前にインストールされたエージェントタイプがエンフォーサである場合、このフラグを上書きします

-goldenImage：VDI 環境または VM テンプレートのゴールデンイメージにエージェントをインストールするときに、このフラグを使用します。このパラメータにより、ゴールデンイメージのインストール後はエージェントサービス（tetsensor および tetenforcer）が自動的に開始されなくなります。異なるホスト名を持つゴールデンイメージから作成されたインスタンスでは、これらのサービスは予想されるとおりに自動で開始されます。

-installFolder：このフラグを使用して、-installFolder フラグで指定されたカスタムフォルダにエージェントをインストールします（例：‘-installFolder “c:\custom sensor path”’）デフォルトのパスは「C:\Program Files\Cisco Tetration」です。

従来のパッケージインストーラを使用した手動インストール（Windows）

このセクションでは、エージェントイメージをダウンロードしてWindowsホストにインストールする方法について説明します。

（注） ほとんどの場合、手動でインストールする特別な理由がない限り、より単純な自動インストール方法（前述）を使用する必要があります。

（注） 既存の実行中のエージェントに古いバージョンのエージェント MSI を手動で展開しないでください。

パッケージ内のサイト関連ファイル：

- **ca.cert**（必須）：センサー通信用の CA 証明書。
- **Enforcer.cfg**（適用センサーをインストールする場合にのみ必須）：適用エンドポイントの構成が含まれています。
- **sensor_config**（必須）：優れた可視性センサーの構成。
- **sensor_type**：センサーのタイプ（適用または優れた可視性）。
- **site.cfg**（必須）：グローバル サイト エンドポイント構成。
- **user.cfg**（TaaS の場合は必須）：センサー アクティベーション キーとプロキシ構成。

前提条件：

デフォルトのテナントの下にインストールしていないときの SaaS および複数のテナントがあるオンプレミスクラスタの場合：

手動でインストールする前に、user.cfg ファイルで ACTIVATION_KEY と HTTPS_PROXY を設定する必要があります。詳細については、（手動インストールのみ）ユーザー構成ファイルの更新を参照してください。（手動インストールのみ）ユーザー構成ファイルの更新 （31 ページ）

ステップ 1 左側のナビゲーションバーで、**[管理（Manage）]** > **[エージェント（Agents）]** をクリックします。

ステップ 2 [インストーラ（Installer）] タブをクリックします。

ステップ 3 [従来のパッケージインストーラを使用した手動インストール（Manual Install using classic packaged installers）] ワークフローを選択し、[次へ（Next）] をクリックします。

ステップ 4 適切なバージョン/プラットフォーム/アーキテクチャ/エージェントタイプを見つけて、[ダウンロード（Download）] をクリックします。

ステップ 5 ZIP パッケージを展開用のすべての Windows ホストにコピーし、管理者権限で以下の手順に従います。

ステップ 6 **tet-win-sensor<version>.win64-<clustername>.zip** ファイルを解凍して、非圧縮フォルダに移動します。

ステップ 7 コマンド `msiexec.exe /i TetrationAgentInstaller.msi` を実行してインストールします。いくつかのオプションが利用可能です。

MSI インストーラで使用可能なオプション：

- **agenttype=<AgentType>**：AgentType は、適用が必要かどうかに応じて、「sensor」または「sensor」のいずれかを指定します。デフォルトでは、インストーラは同じフォルダ内の **sensor_type** ファイルの内容をチェックします（渡されたパラメータを上書きします）。エージェントが **/quiet** モードでインストールされている場合、これは必須です。
- **overwritenpcap=yes**：デフォルトでは、Npcap がすでに存在する場合、エージェントは Npcap のアップグレードを試みません。このパラメータを渡すと、既存の Npcap のアップグレードが試行されます。このオプションが使用されている場合、後続のエージェントの自動アップグレードでも、Npcap がサポートされている新しいバージョンにアップグレードされます。
- **nostart=yes**：エージェントサービス（tetsensor および tetenforcer）が自動的に開始されないようにするために、VDI 環境または VM テンプレートのゴールデンイメージにエージェントをインストールする場合、このパラメータを渡す必要があります。（異なるホスト名を持つゴールデンイメージから作成された VDI/VM インスタンスでは、これらのサービスは期待どおりに自動的に開始されます。）
- **installfolder=<FullPathCustomFolder>**：上記のコマンドの最後にこのパラメータを使用して、センサーをカスタムフォルダにインストールします。
- **serviceuser=<Service UserName>**：上記のコマンドの最後にこのパラメータを使用して、サービスユーザーを構成します。デフォルトのサービスユーザーは「LocalSystem」です。

  ローカルユーザーの場合、**serviceuser=.\<Service UserName>**

  ドメインユーザーの場合、**serviceuser=<domain_name>\<samaccount name>** です。サービスユーザーには、**ローカル管理者権限**が必要です。
- **servicepassword=<Service UserPassword>**：上記のコマンドの最後にこのパラメーターを使用して、サービスユーザーのパスワードを構成します。パスワードはプレーンテキスト形式である必要があります。
- **proxy="<proxy_address>"**：このパラメータを使用して、Secure Workload クラスタに到達するように HTTPS プロキシを設定します。
- **activationkey=<activation Key>**：デフォルトテナント配下にエージェントがインストールされていない場合は、このパラメータでテナントを指定します。

（注）
- 手動インストール中に activationkey および proxy オプションを使用する場合、user.cfg を手動で構成する必要はありません。
- Npcap がまだインストールされていない場合は、Cisco Secure Workload サービスの tetsensor によって Npcap が自動的にインストールされます。
- エージェントがすでにインストールされている場合は、再インストールしないでください。エージェントを新しいバージョンにアップグレードする必要がある場合は、「ソフトウェアエージェントのアップグレード」で説明されているアップグレードプロセスに従ってください。

エージェントを新しいバージョンにアップグレードする必要がある場合は、「ソフトウェアエージェントのアップグレード」で説明されているアップグレードプロセスに従ってください。

エージェントがインストールされていることを確認する

ステップ 1 フォルダ `C:\\Program Files\\Cisco Tetration`（またはカスタムフォルダ）が存在することを確認します。

ステップ 2 TetSensor サービス（優れた可視性エージェント向け）が存在し、実行されていることを確認します。**管理者**権限で `cmd.exe` コマンドを実行します。

`sc query tetsensor` コマンドを実行します。

状態が **Running** であることを確認します。

`sc query tetsensor` コマンドを実行します。

DISPLAY-NAME が **Cisco Secure Workload Deep Visibility** であるかを確認します。

または

services.msc コマンドを実行します。

Cisco Secure Workload Deep Visibility という名前を見つけます。

状態が **Running** であることを確認します。

ステップ 3 TetEnforcer サービス（適用エージェント向け）が存在し、実行されていることを確認します。

管理者権限で `cmd.exe` コマンドを実行します。

`sc query tetenforcer` コマンドを実行します。

状態が **Running** であることを確認します。

`sc qc tetenforcer` コマンドを実行します。

DISPLAY-NAME が **Cisco Secure Workload Enforcement** であるかを確認します。

または

`services.msc` コマンドを実行します。

Cisco Secure Workload Enforcement という名前を見つけます。

状態が **Running** であることを確認します。

設定されたサービス ユーザー コンテキストでエージェントが実行されていることを確認します。

1. TetSensor（詳細可視性のため）と TetEnforcer（適用のため）の両サービスが設定されたサービスユーザー コンテキストで実行されていることを確認します。TetSensor と TetEnforcer は、同じサービス ユーザー コンテキストで実行されます。

   管理者権限で `cmd.exe` コマンドを実行します。

   `sc query tetsensor` コマンドを実行します。

   SERVICE_START_NAME **<configured service user>** を確認します。

   `sc qc tetenforcer` コマンドを実行します。

   SERVICE_START_NAME **<configured service user>** を確認します。

   または

   `services.msc` コマンドを実行します。

   Cisco Secure Workload Deep Visibility という名前を検索します。

   <configured service user> の **Log On As** を確認します。

   Cisco Secure Workload Enforcement という名前を検索します。

   <configured service user> の **Log On As** を確認します。

   または

   `tasklist /v | find /i "tet"` コマンドを実行します。

   実行中のプロセスのユーザーコンテキストを確認します（5 列目）。

VDI インスタンスまたは VM テンプレートへのエージェントの展開（Windows）

デフォルトでは、エージェントのインストール後にエージェントサービスが自動的に開始されます。ゴールデンイメージにインストールする場合は、インストーラフラグを使用して、エージェントサービスが開始されないようにする必要があります。インスタンスがゴールデンイメージから複製されると、エージェントサービスは予想どおりに自動的に開始されます。

同様に、Npcapは通常、エージェントのインストール後に自動的にインストールされます（エージェントがまだ存在していない場合）。Npcap はゴールデンイメージに自動的にインストールされませんが、必要に応じて、ゴールデンイメージから複製された VM インスタンスに自動的にインストールされます。詳細については、「Windows エージェント インストーラと Npcap」を参照してください。

VDI 環境または VM テンプレートのゴールデンイメージにエージェントをインストールする

ステップ 1 MSI インストーラまたは PowerShell インストーラスクリプトを使用して、VDI 環境または VM テンプレートのゴールデンイメージにエージェントをインストールします。

nostart=yes を指定した MSI インストーラーを使用

- 詳細については、「従来のパッケージインストーラを使用した手動インストール（Windows）」を参照してください。
- msiexec.exe /<MSI installer> nostart=”yes” /quiet /norestart /l*v <installer_log_file>

または

-goldenImage フラグを指定した PowerShell インストーラを使用

- 詳細については、「エージェントのインストール」を参照してください。

ステップ 2 フォルダ `C:\\Program Files\\Cisco Tetration`（またはカスタムフォルダ）が存在することを確認します。

ステップ 3 TetSensor サービス（優れた可視性エージェント向け）が存在し、停止していることを確認します。

管理者権限で `cmd.exe` コマンドを実行します。

`sc query tetsensor` コマンドを実行します。

状態が **Stopped** であることを確認します。

ステップ 4 TetEnforcer サービス（適用エージェント向け）が存在し、停止していることを確認します。

sc query tetenforcer コマンドを実行します。

状態が Stopped であることを確認します。

ステップ 5 これで VM テンプレートが設定されました。

ステップ 6 VM テンプレートをシャットダウンします。

新規 VDI インスタンス VM の作成

ステップ 1 VM テンプレートを複製して、新規 VDI インスタンス VM を作成します。

ステップ 2 VDI インスタンス VM を再起動します。

ステップ 3 VDI インスタンス VM を再起動した後、TetSensor（詳細可視性のため）と TetEnforcer（適用のため）の両サービスが設定されたサービスコンテキストで実行されていることを確認します。「エージェントがインストールされていることを確認する」を参照してください。

ステップ 4 VDI インスタンス VM で、NPCAP ドライバがインストールされ、実行されていることを確認します。

管理者権限でコマンド `cmd.exe` を実行します。

コマンド `sc query npcap` を実行します。

[実行中（Running）] の状態を確認します。

ステップ 5 VDI インスタンス VM で、有効な sensor_id を使用してエージェントが登録されていることを確認します。

- インストールフォルダの sensor_id ファイルを確認します。
- sensor_id が「uuid」で始まる場合、それは有効な sensor_id ではありません。
- エージェントの登録に失敗したにもかかわらず、Secure Workload Web インターフェイスにエージェントが登録済みと表示されている場合：
- OpenAPI を使用してエージェントを削除してください。「ソフトウェアエージェントの展開」の下の注記を参照してください。

（注）
- ゴールデンイメージまたは VM テンプレートのホスト名は変更しないでください。
- エージェントのインストール後にゴールデンイメージまたは VM テンプレートを再起動すると、再起動後に Secure Workload サービスの実行が開始されます。
- VDI インスタンス VM がネットワークフローのレポートに失敗する場合は、「ネットワークフローを報告しない VDI インスタンス VM 」を参照してください。

.

Windows エージェント インストーラと Npcap

1. サポートされている Npcap バージョンについては、https://www.cisco.com/go/secure-workload/requirements/agents のサポートマトリックスを参照してください。
2. インストール：

   Npcap がインストールされていない場合、エージェントはサービスの開始から 10 秒後にサポートされているバージョンをインストールします。ユーザーが Npcap をインストールしていても、サポートされているバージョンより古い場合、Npcap はアップグレードされません。Npcap を自分でアップグレード/アンインストールするか、オプション **overwritenpcap=yes** を指定してエージェントインストーラを実行するか、または **-npcap** を指定してインストーラスクリプトを実行して、サポートされている Npcap バージョンを取得してください。Npcap ドライバがアプリケーションで使用されている場合、エージェントは後で Npcap のアップグレードを試みます。
3. アップグレード：

   Npcap が Windows エージェントによってインストールされ、そのバージョンがサポートされているバージョンより古い場合、Npcap はサービスの開始から 10 秒後にサポートされているバージョンにアップグレードされます。Npcap ドライバがアプリケーションで使用されている場合、エージェントは後で Npcap のアップグレードを試みます。Npcap が Windows エージェントによってインストールされていない場合、Npcap はアップグレードされません。
4. アンインストール：

Npcap が Windows エージェントによってインストールされている場合、Npcap は Windows エージェントによってアンインストールされます。Npcap がユーザーによってインストールされていても、**overwritenpcap=yes** でエージェントインストーラによってアップグレードされた場合、アンインストールされません。Npcap ドライバがアプリケーションで使用されている場合、エージェントは Npcap をアンインストールしません。

AIX エージェント - 詳細可視性と適用

（注） プロセスツリー、パッケージ（CVE）、およびフォレンジックイベントレポート機能は、AIX ではまだ使用できません。さらに、これらの機能の一部は、OS の制限により、サポートされているプラットフォームの特定のマイナーリリースでは利用できない場合があります。

要件および前提条件

「サポートされているプラットフォームと要件」を参照してください。

優れた可視性のための追加要件

- サービスをインストールして実行するには、ルート権限が必要です。
- エージェントおよびログファイルのストレージ要件：500MB。
- ホストを監視しているすべてのセキュリティアプリケーションでセキュリティの除外を構成することにより、他のセキュリティアプリケーションがエージェントのインストールまたはエージェントのアクティビティをブロックしないようにします。「セキュリティの除外」を参照してください。
- AIX は、20 個のネットデバイスのフローキャプチャのみをサポートします（バージョンが AIX 7.1 TL3 SP4 以前の場合は 6 個）。優れた可視性エージェントは、最大 16 のネットワークデバイスからキャプチャを行い、他の 4 つのキャプチャセッションを一般的なシステム用途（tcpdump など）に排他的に使用できるようにします。
- 優れた可視性エージェントは、この動作を確実にするために次のことを行います。
  - エージェントは、エージェントディレクトリ（/opt/cisco/tetration/chroot/dev/bpf0 - /opt/cisco/tetration/chroot/dev/bpf15）の下に 16 個の bpf デバイスノードを作成します。
  - bpf を使用する tcpdump およびその他のシステムツールは、未使用のノード（!EBUSY）が見つかるまで、システムデバイスノード（/dev/bpf0-/dev/bpf19）をスキャンします。
  - エージェントが作成した bpf ノードとシステム bpf ノードは同じメジャー/マイナーを共有し、各メジャー/マイナーは 1 つのインスタンス（tcpdump またはエージェント）によってのみ開かれます。
  - エージェントはシステムデバイスノードにアクセスせず、tcpdump のようにそれらを作成しません（tcpdump-D は /dev/bpf0. . . /dev/bpf19 が存在しない場合これらを作成します）。

- システムで iptrace を実行すると、特定のシナリオで tcpdump および優れた可視性エージェントからのフローキャプチャが防止されます。これは既知の設計上の不具合です。IBM に確認してください。
  - エージェントをインストールする前にこのシナリオが存在するかどうかを確認するには、tcpdump を実行します。**tcpdump: BIOCSETIF: en0: File exists** のようなエラーメッセージが表示される場合、iptrace はフローキャプチャをブロックしています。iptrace を停止すると、問題が解決します。
- AIX では、すべての優れた可視性機能がサポートされているわけではありません。例えば、パッケージとプロセスのアカウンティングはサポートされていません。

ポリシーの適用に関する追加要件：

- IP セキュリティフィルタが有効になっている場合（例：smitty ipsec4）、事前チェックでエージェントのインストールが失敗します。エージェントをインストールする前に、**IP セキュリティフィルタを無効にする**ことをお勧めします。
- IP セキュリティが有効になっている場合、Secure Workload エンフォーサエージェントの実行中にエラーとして報告され、エンフォーサエージェントは適用を停止します。適用エージェントの実行中に IP セキュリティフィルタを安全に無効にするには、サポートにお問い合わせください。

エージェントのインストール

優れた可視性と適用の AIX エージェントは、インストールスクリプトを使用しないとインストールできません。

デフォルトでは、インストールされたエージェントは優れた可視性と適用の両方をサポートしています。デフォルトでは適用は無効になっていますが、Secure Workload ユーザーインターフェイスを使用して簡単に有効にすることができます。

そのプロセスは次のとおりです。

ステップ 1 左側のナビゲーションバーで、**[管理（Manage）]** > **[エージェント（Agents）]** をクリックします。

ステップ 2 [インストーラ（Installer）] タブをクリックします。

ステップ 3 [インストーラを使用して自動インストール（Auto-Install using Installers）] ワークフローを選択し、[次へ（Next）] をクリックします。

ステップ 4 エージェントをインストールするテナントを選択します。Secure Workload SaaS クラスタでは、テナントの選択は必要ありません。

ステップ 5 CMDB ラベルを選択し、関連付けられた値を入力して、それをエージェントインストーラに添付します（オプションの選択）。インストールされたエージェントがこのホストで見つかった新しい IP アドレスをクラスタに報告すると、このホストによって報告された IP に割り当てられているアップロード済みの別の CMDB ラベルとともに、ここで選択されたインストーラ CMDB ラベルが新しい IP に自動的に割り当てられます。アップロードされた CMDB ラベルとインストーラ CMDB ラベルの間で競合が発生した場合：

- 正確な IP アドレスに割り当てられたラベルは、サブネットに割り当てられたラベルよりも優先されます。
- 正確な IP アドレスに割り当てられた既存のラベルは、インストーラの CMDB ラベルよりも優先されます。

ステップ 6 プラットフォームとして AIX を選択します。

ステップ 7 ネットワークに HTTP プロキシが必要な場合は [はい（Yes）] を選択し、有効なプロキシ URL を入力します。それ以外の場合は [いいえ（No）] を選択します。

ステップ 8 [インストーラの有効期限（Installer expiration）] セクションで、利用可能なオプションから 1 つを選択します。

- 有効期限なし：インストーラスクリプトは何回も使用できます。
- 1 回のみ：インストーラスクリプトは 1 回のみ使用できます。
- 時間制限：インストーラスクリプトを使用できる日数を設定できます。
- 展開数：インストーラスクリプトを使用できる回数を設定できます。

ステップ 9 [インストーラのダウンロード（Download Installer）] をクリックし、ファイルをローカルディスクに保存します。インストーラスクリプトがローカルに保存されたら、[次へ（Next）] をクリックします。

ステップ 10 インストーラ シェル スクリプトを、展開するためにすべての AIX ホストにコピーします。

ステップ 11 コマンド `chmod u+x tetration_installer_default_sensor_aix.sh` を実行し、スクリプトの実行権限を許可します。

（注） スクリプト名は、エージェントのタイプと範囲によって異なる場合があります。

ステップ 12 エージェントをインストールするには、root 権限でコマンド `./tetration_installer_default_sensor_aix.sh` を実行します。

以下のスクリプトの使用方法の詳細で指定されているように、事前チェックを実行することをお勧めします。

（注） エージェントがすでにインストールされている場合、スクリプトは続行されません。

図 *5*: ソフトウェア エージェント インストーラ スクリプトのダウンロードページ（オンプレミス）

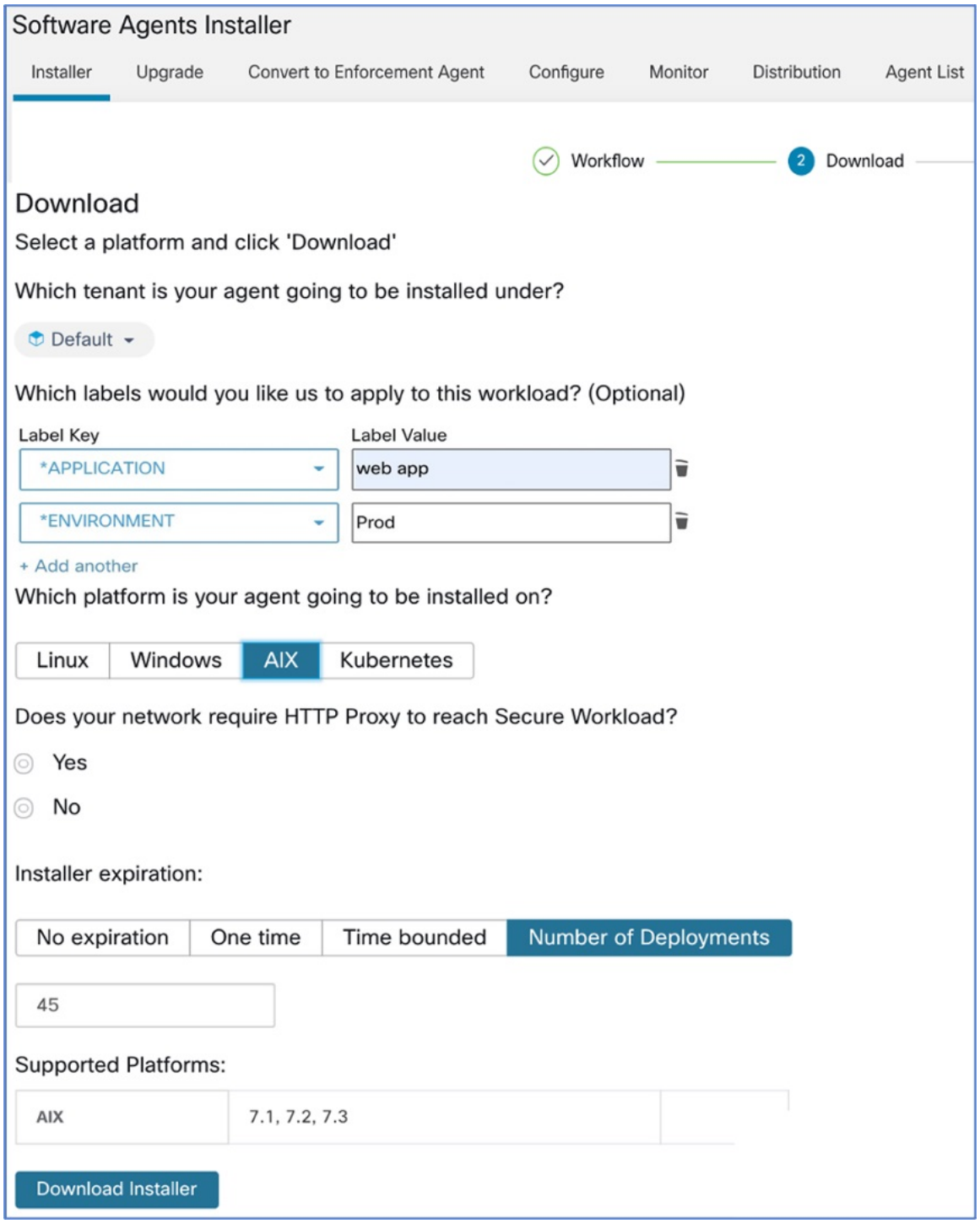
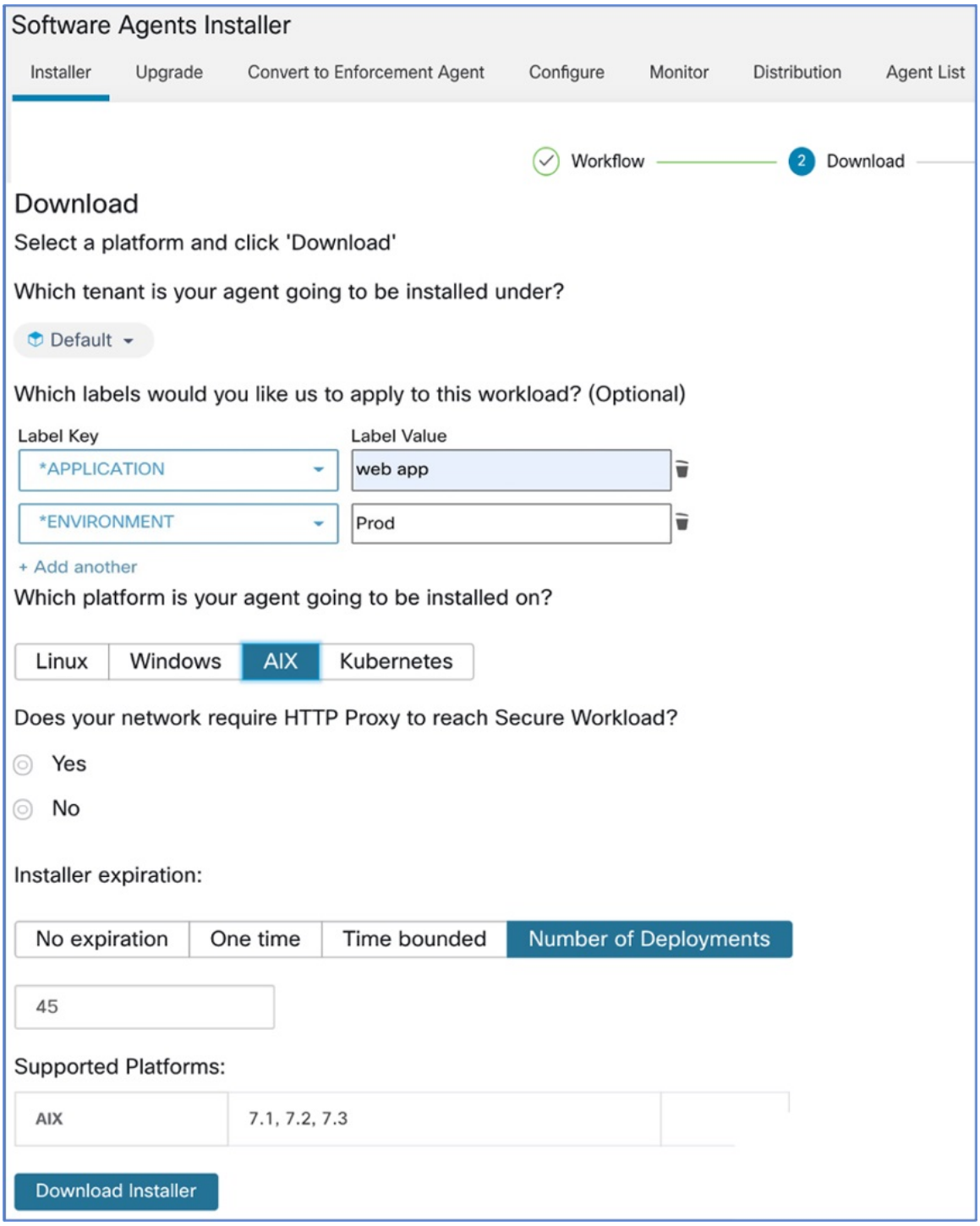


図 *6*: ソフトウェア エージェント インストーラ スクリプトのダウンロードページ（***Saas***）



インストーラスクリプトの使用方法は、次のとおりです。

$ ksh tetration_installer_aix.sh [–pre-check] [–pre-check-user] [–skip-pre-check=<option>] [–no-install] [–logfile=<filename>] [–proxy=<proxy_string>] [–no-proxy] [–help] [–version] [–sensor-version=<version_info>]

[–ls] [–file=<filename>] [–osversion=<osversion>] [– save=<filename>] [–new] [–reinstall] [–unpriv-user] [–libs=<libs.zip|tar.Z>] [–force-upgrade] [–upgrade-local] [–upgrade-by-uuid=<filename>] [–logbasedir=<logbdir>] [–visibility]

–pre-check：事前チェックのみを実行します

–pre-check-user：事前チェック su サポートに nobody ユーザーの代替を提供します

–skip-pre-check=<option>：指定されたオプションで事前インストールチェックをスキップします。有効なオプションには「all」、「ipv6」および「enforcement」が含まれます。たとえば、–skip-pre-check=all' は、すべての事前インストールチェックをスキップします。すべての事前チェックはデフォルトで実行されます

–no-install：システムでのセンサーパッケージのダウンロードおよびインストールは行いません

–logfile <filename>：<filename> で指定されたファイルにログを書き込みます

–proxy=<proxy_string>：HTTPS_PROXY の値を指定します。文字列は http://<proxy>:<port> の形式にする必要があります。

–no-proxy：システム全体のプロキシをバイパスします。–proxy フラグが指定されている場合、このフラグは無視されます

–help：このヘルプを印刷します

–version：現在のスクリプトのバージョンを印刷します

–sensor-version=<version_info>：センサーのバージョン（たとえば「–sensor- version=3.4.1.0」）を選択します。このフラグが提供されていない場合、デフォルトで最新バージョンをダウンロードします

–ls：システムで使用可能なすべてのセンサーバージョンを一覧表示します（3.3 以前のパッケージは表示しません）。パッケージはダウンロードしません

–file <filename>：クラスタからダウンロードする代わりに、センサーをインストールするためのローカル zip ファイルを提供します

–osversion=<osversion>：–save フラグに osversion を指定します

–save=<filename>：zip ファイルをダウンロードして <filename> として保存します。–osversion flag（例：'–save=myimage.aix72.zip – osversion=7.2'）で指定された osversion のパッケージをダウンロードします

–new：以前にインストールされたセンサーをすべて削除します。新しい登録を成功させるには、以前のセンサー ID をクラスタから削除する必要があります

–reinstall：センサーを再インストールし、クラスタと同じ ID を保持します。このフラグは、-new よりも優先されます

–unpriv-user=<username>：unpriv プロセスで tet-snsr の代わりに <username> を使用します

–libs=<libs.zip>：エージェントが使用する指定されたライブラリをインストールします

–force-upgrade：–sensor-version フラグで指定されたバージョンへのセンサーのアップグレードを強制します（例：'–sensor-version=3.4.1.0 –force-upgrade'）– sensor-version フラグが指定されていない場合、デフォルトで最新バージョンを適用します

–upgrade-local：–sensor-version フラグで指定されたバージョンへのローカルセンサーのアップグレードをトリガーします（例：'–sensor-version=3.4.1.0 –upgrade-local'）次の場合、デフォルトで最新バージョンを適用します

–sensor-version フラグは提供されませんでした

–upgrade-by-uuid=<filename>：<filename> に uuid がリストされているトリガーセンサーを –sensor-version フラグで指定されたバージョンにアップグレードします（例：'–sensor-version=3.4.1.0 –upgrade-by-uuid=/usr/local/tet/sensor_id'）– sensor-version フラグが指定されていない場合、デフォルトで最新バージョンを適用します

–logbasedir=<log_base_dir>：/opt/cisco/tetration/log use にログインする代わりに

<log_base_dir> を使用します。フルパスは <log_base_dir>/tetration です

–visibility：優れた可視性エージェントのみをインストールします。 –reinstall は、以前にインストールされたエージェントタイプがエンフォーサである場合、このフラグを上書きします

エージェントがインストールされていることを確認する

手順の概要

1. コマンド `lslpp -c -l tet-sensor.rte` を実行し、以下のように 1 つのエントリがあることを確認します。

手順の詳細

コマンド `lslpp -c -l tet-sensor.rte` を実行し、以下のように 1 つのエントリがあることを確認します。

（注）　具体的な出力データは、バージョンによって異なる場合があります。

$ sudo lslpp -c -l tet-sensor.rte /usr/lib/objrepos:tet-sensor.rte:3.4.1.19::COMMITTED:I:TET tet sensor package:

$ sudo lssrc -s tet-sensor

Subsystem Group PID Status tet-sensor 1234567 active

$ sudo lssrc -s tet-enforcer

Subsystem Group PID Status tet-enforcer 7654321 active

Kubernetes/Openshift エージェント：優れた可視性と適用

要件および前提条件

Kubernetes 1.[16-22]

- RHEL：7.[0-9]（x86_64 アーキテクチャのみ）
- CentOS：7.[0-8]（x86_64 アーキテクチャのみ）
- Oracle Linux：7.[0-8]（x86_64 アーキテクチャのみ）
- Ubuntu：16.04、18.04、20.04（x86_64 アーキテクチャのみ）
- SUSE Linux Enterprise Server：12sp[0-5]（x86_64 アーキテクチャのみ）
- Amazon Linux 2（x86_64 アーキテクチャのみ）

Openshift 4.[5-9]

- Red Hat Enterprise Linux CoreOS：4.[5-9]（x86_64 アーキテクチャのみ）

コンテナ ランタイム

- Docker
- CRI-O
- containerd（>= 1.5.x）

（注） containerd ランタイムでは、config_path が設定されていない場合は、config.toml（デフォルトの場所：/etc/containerd/config.toml）を次のように変更します。

[plugins.”io.containerd.grpc.v1.cri”.registry] config_path = “/etc/containerd/certs.d”

containerd デーモンを再起動します。

追加

- インストールスクリプトには、クラスタノードで特権エージェントポッドを起動するための Kubernetes/Openshift 管理者の資格情報が必要です。
- Secure Workload アプリケーション エンティティは、「tetration」という名前の名前空間に作成されます。
- ノード/ポッドのセキュリティポリシーは、特権モードのポッドを許可する必要があります。

- busybox:1.33 イメージは、事前にインストールされているか、Docker Hub からダウンロード可能である必要があります。
- Kubernetes/Openshift コントロールプレーンノードで実行するには、-toleration フラグを使用して、Secure Workload ポッドの容認を渡すことができます。これは通常、ポッドがコントロールプレーンノードで実行されないようにする NoSchedule 容認です。

ポリシー適用の要件

コンテナ オーケストレーション プラットフォームでポリシーを適用するエージェントは、RHEL 7.[0-9]、CentOS 7.[0-8]、または Ubuntu 16.04/18.04/20.04 ノードでサポートされます。

IPVS ベースの kube-proxy モードは、OpenShift ではサポートされません。

これらのエージェントは、[ルールの保持（Preserve Rules）] オプションを有効にして設定する必要があります。「エージェント設定プロファイルの作成」を参照してください。

適用が適切に機能するためには、インストールされている CNI プラグインが次の条件を満たす必要があります。

- すべてのノードとポッド間にフラットなアドレス空間（IP ネットワーク）を提供すること。クラスタ内通信のためにソースポッド IP をマスカレードするネットワークプラグインはサポートされていません。
- Secure Workload 適用エージェントが使用する Linux iptables ルールまたはマークに干渉しないこと（マークビット 21 および 20 は、NodePort サービスのトラフィックを許可および拒否するために使用されます）

次の CNI プラグインは、上記の要件を満たすことがテストされています。

- 次の Felix 設定を持つ Calico (3.13)：*(ChainInsertMode: Append, Ipta- blesRefreshInterval: 0)* または *(ChainInsertMode: Insert, IptablesFilterAllowAction: Return, IptablesMangleAllowAction: Return, IptablesRefreshInterval: 0)*。その他のオプションは、そのデフォルト値を使用します。

これらのオプションの設定に関する詳細については、Felix 設定リファレンスを参照してください。

エージェントのインストール

この「インストーラースクリプト」による方法では、将来のノードにエージェントが自動的にインストールされます。

ステップ 1 左側のナビゲーションバーで、**[管理（Manage）]** > **[エージェント（Agents）]** をクリックします。

ステップ 2 [インストーラ（Installer）] タブをクリックします。

ステップ 3 [インストーラを使用して自動インストール（Auto-Install using Installers）] を選択し、[次へ（Next）] をクリックします。

ステップ 4 [Kubernetes] を選択します（該当する場合はテナント範囲を選択します）。

ステップ 5 ネットワークに HTTP プロキシが必要な場合は [はい（Yes）] を選択し、有効なプロキシ URL を入力します。それ以外の場合は [いいえ（No）] を選択します。

ステップ 6 [インストーラの有効期限（Installer expiration）] セクションで、利用可能なオプションから 1 つを選択します。

- 有効期限なし：インストーラスクリプトは何回も使用できます。
- 1 回のみ：インストーラスクリプトは 1 回のみ使用できます。
- 時間制限：インストーラスクリプトを使用できる日数を設定できます。
- 展開数：インストーラスクリプトを使用できる回数を設定できます。

ステップ 7 [Download Installer（インストーラのダウンロード）] をクリックし、ファイルをローカルディスクに保存します。インストーラスクリプトがローカルに保存されたら、[次へ（Next）] をクリックします。

ステップ 8 Kubernetes API サーバーにアクセスでき、デフォルトのコンテキスト/クラスタ/ユーザーとしての管理者権限を持つ kubectl 構成ファイルも存在する Linux マシンで、インストーラスクリプトを実行します。

ステップ 9 インストーラはデフォルトの場所（~/.kube/config）からファイルを読み取ろうとしますが、これは `--kubeconfig` コマンドラインオプションで明示的に指定できます。

ステップ 10 インストールスクリプトが成功すると、インストールされた Secure Workload Agent Daemonset とポッドを確認する方法の説明が出力されます。

（注） ダウンロード前にエージェント インストーラ ページで構成された HTTP プロキシは、Secure Workload エージェントが Secure Workload クラスタに接続する方法のみを制御します。この設定は、Kubernetes/Openshift ノードによる Docker イメージの取得方法には影響しません。これらのノードのコンテナランタイムでは、独自のプロキシ設定が使用されるためです。Docker イメージを Secure Workload クラスタからプルできない場合は、コンテナランタイムのイメージプルプロセスのデバッグが必要になり、適切な HTTP プロキシの追加が必要になる場合があります。

図 *7:* ソフトウェア エージェント インストーラ スクリプトのダウンロードページ（オンプレミス）

（手動インストールのみ）ユーザー構成ファイルの更新

以下の手順は、次のすべてを含むインストールにのみ必要です。

- Secure Workload SaaS、または複数のテナントを持つオンプレミスクラスタ（デフォルトのテナントのみを使用するオンプレミスクラスタでは、この手順は不要）
- 手動インストール
- Linux または Windows プラットフォーム

エージェントを Secure Workload クラスタに登録するには、クラスタ アクティベーション キーが必要です。さらに、エージェントがクラスタに到達するために HTTPS プロキシが必要な場合は、プロキシを指定する必要があります。

（注） Windows環境では、手動インストール時にアクティベーションキーとプロキシオプションを使用する場合、user.cfg を手動で構成する必要はありません。

インストールの前に、ユーザー構成ファイルで必要な変数を設定します。

ステップ 1 アクティベーションキーの取得：**[管理（Manage）]** > **[エージェント（Agents）]**に移動し、[インストーラ（Installer）] タブをクリックし、[従来のパッケージインストーラを使用した手動インストール（Manual Install using classic packaged installers）] をクリックしてから、[エージェントアクティベーションキー（Agent Activation Key）] をクリックします。

ステップ 2 Secure Workload エージェントのインストールフォルダにある `user.cfg` ファイルを開いて編集します（例：Linux の場合は `/usr/local/tet`、Windows の場合は `C:\\Program Files\\Cisco Tetration`）。このファイルには、各行に 1 つずつ「key=value」の形式で変数のリストが含まれています。

ステップ 3 アクティベーションキーを **ACTIVATION_KEY** 変数に追加します。例：
`ACTIVATION_KEY=7752163c635ef62e6568e9e852d07bd21bfd60d0`

ステップ 4 エージェントに HTTPS プロキシが必要な場合は、**HTTPS_PROXY** 変数を使用して **http** プロトコルプロキシサーバーとポートを追加します。例：`HTTPS_PROXY=http://proxy.my-company.com:80`

その他のエージェント同様のツール

AnyConnect エージェント

Network Visibility Module (NVM) を備えた Cisco AnyConnect セキュア モビリティ エージェントでサポートされるプラットフォームです。追加の Secure Workload エージェントは必要ありません。AnyConnect コネクタは、これらのエージェントを登録し、フロー観測データ、インベ

ントリデータ、およびラベルをSecure Workload にエクスポートします。詳細については、「AnyConnect コネクタ」を参照してください。

Windows、Mac、Linux プラットフォームについては、『Cisco AnyConnect セキュア モビリティ クライアント データ シート』[英語] を参照してください。

ISE エージェント

Cisco Identity Services Engine（ISE）に登録されたエンドポイント。このエンドポイントに Secure Workload エージェントは必要ありません。ISE コネクタは、ISE アプライアンスの pxGrid サービスを介して ISE からエンドポイントに関するメタデータを収集します。エンドポイントを Secure Workload で ISE エージェントとして登録し、このエンドポイントにインベントリのラベルをプッシュします。詳細については、「ISE コネクタ」を参照してください。

SPAN エージェント

SPAN エージェントは ERSPAN コネクタと連携します。詳細については、「ERSPAN コネクタ」を参照してください。

NetFlow、NetScaler、F5、AWS などのその他のコネクタ

コネクタの詳細については、「コネクタとは」を参照してください。

接続情報

一般に、エージェントがワークロードにインストールされると、Secure Workload クラスタでホストされているバックエンドサービスへの多数のネットワーク接続が開始されます。エージェントのタイプとその機能に応じて、接続数の表示は異なります。

次の表は、さまざまなエージェントタイプによって確立されたさまざまな固定接続を示しています。

表 *1:* エージェントの接続

エージェント タイプ	コンフィギュレーション サーバー	コレクタ	適用バックエンド
可視性（オンプレミス）	CFG-SERVER-IP:443	COLLECTOR-IP:5640	該当なし
可視性（TaaS）	CFG-SERVER-IP:443	COLLECTOR-IP:443	該当なし
適用（オンプレミス）	CFG-SERVER-IP:443	COLLECTOR-IP:5640	ENFORCER-IP:5660
適用（TaaS）	CFG-SERVER-IP:443	COLLECTOR-IP:443	ENFORCER-IP:443
Docker イメージ	CFG-SERVER-IP:443	該当なし	該当なし

説明：

- CFG-SERVER-IP は、コンフィギュレーション サーバーの IP アドレスを表します。
- COLLECTOR-IP は、コレクタの IP アドレスを表します。詳細可視性エージェントと適用エージェントは、利用可能なすべてのコレクタに接続します。
- ENFORCER-IP は、適用エンドポイントの IP アドレスを表します。適用エージェントは、使用可能なエンドポイントのうち 1 つのみに接続します。
- Kubernetes/Openshift エージェントの展開の場合、インストールスクリプトにエージェントソフトウェアが含まれていません。エージェントソフトウェアを含む Docker イメージは、各 Kubernetes/Openshift ノードによって Secure Workload クラスタからプルされます。これらの接続は、コンテナランタイムイメージ取得コンポーネントによって確立され、CFG-SERVER-IP:443 が接続先になります。

（注）

- Secure Workload エージェントは、常にクライアントとして機能し、クラスタ内でホストされているサービスへの接続を開始しますが、サーバーとして接続を開始することはありません。
- 上記の固定接続に加えて、アップグレードがサポートされている特定のエージェントタイプの場合、エージェントは定期的に HTTPS 要求（ポート 443）をクラスタセンサー VIP に対して実行し、使用可能なパッケージを照会します。
- エージェントは NAT サーバーの背後に配置できます。

ワークロードがファイアウォールの背後にある場合や、ホスト ファイアウォール サービスが有効になっている場合は、クラスタへの接続が拒否される可能性があることに注意することが重要です。管理者は、適切なファイアウォールポリシーを作成して、そのような接続を許可する必要があります。

セキュリティの除外

Secure Workload エージェントは、通常の動作中にホストのオペレーティングシステムと継続的に対話します。これにより、ホストにインストールされている他のセキュリティ アプリケーション（ウイルス対策、セキュリティエージェントなど）. が、Secure Workload エージェントに関するアラームを発したり、Secure Workload エージェントのアクションをブロックすることもあります。Secure Workload エージェントを適切にインストールして効果的に機能させるには、ホストを監視しているセキュリティアプリケーションで必要なセキュリティの除外を構成してください。

表 *2 : Secure Workload* エージェントに関するセキュリティ除外のディレクトリ

ホスト OS	ディレクトリ
AIX	/opt/cisco/tetration

ホスト OS	ディレクトリ
Linux	/usr/local/tet または /opt/cisco/tetration or <user chosen inst dir>
Windows	C:\Program Files\Cisco Tetration

表 *3 : Secure Workload* エージェントに関するセキュリティ除外のプロセス

ホスト OS	プロセス（Processes）
AIX	tet-engine、tet-sensor、tet-enforcer
Linux	tet-engine、tet-sensor、tet-enforcer、tet-main、enforcer
Windows	TetSenEngine.exe、TetSen.exe、TetEnfEgine.exe、TetEnfC.exe、TetEnf.exe、TetUpdate.exe、tet-main.exe

表 *4 : Secure Workload* エージェントに関するセキュリティ除外のアクション

ホスト OS	アクション（Actions）
AIX	/dev/bpf*、/dev/ipl、/dev/kmem へのアクセス、curl の呼び出し
Linux	/proc のスキャン、netlink sockets を開く、curl、rpm/dpkg、ip[6]tables-save、ip[6]tables-restore、ipset-restore を呼び出す
Windows	レジストリへのアクセス、ファイアウォールイベントへの登録

表 *5 : Secure Workload* エージェントに関するセキュリティ除外のスクリプト/バイナリ実行

ホスト OS	呼び出されるスクリプト/バイナリ
AIX	ksh: fetch_sensor_id.sh、check_conf_update.sh
Linux	bash：fetch_sensor_id.sh、check_conf_update.sh
Windows	cmd：fetch_sensor_id.cmd、check_conf_update.cmd、dmidecode.exe、npcap-installer.exe、sensortools.exe、signtool.exe

ソフトウェア エージェント サービスの管理

ソフトウェアエージェントは、サポートされているすべてのプラットフォームでサービスとして展開されます。このセクションでは、さまざまな機能とプラットフォームのサービスを管理する方法について説明します。

指定されていない限り、以下のすべてのコマンドを実行するには、ルート権限（Linux/UNIX）または管理者権限（Windows）が必要です。

RHEL/CentOS/OracleLinux-6.x および Ubuntu-14 のサービス管理

サービスの開始

start <service-name> コマンドを実行します。

例

優れた可視性サービスの場合は **start tet-sensor**：適用サービスの場合は **start tet-enforcer**

サービスの停止

コマンド **stop <service-name>** を実行します。

例

stop tet-sensor は優れた可視性サービス用です。**stop tet-enforcer** は適用サービス用です。

サービスの再起動

restart <service-name> コマンドを実行します。

例

優れた可視性サービスの場合は **restart tet-sensor**：適用サービスの場合は **restart tet-enforcer**

サービスステータスの確認

コマンド **status <service-name>** を実行します

例

status tet-sensor for deep visiblity service - **status tet-enforcer** for enforcement service

SLES-11 のサービスの管理

サービスの開始

service <service-name> start コマンドを実行します。

例

優れた可視性サービスの場合は **service tet-sensor start**：適用サービスの場合は **service tet-enforcer start**

サービスの停止

service <service-name> stop コマンドを実行します。

例

優れた可視性サービスの場合は **service tet-sensor stop**、適用サービスの場合は **service tet-enforcer stop** になります。

サービスの再起動

コマンド **service <service-name> stop || true** を実行し、次に **service <service-name> start** を実行します。

サービスステータスの確認

コマンド **status <service-name>** を実行します

例

status tet-sensor は優れた可視性サービス用です。**status tet-enforcer** は適用サービス用です。

RHEL/CentOS/OracleLinux-7.x および 8.x のサービス管理

Ubuntu-16、18、20 および SLES-12 にも同じコマンドを使用できます。

サービスの開始

コマンド **systemctl start <service-name>** を実行します。

例

優れた可視性サービスの場合は **systemctl start tet-sensor**、適用サービスの場合は **systemctl start tet-enforcer**

サービスの停止

コマンド **systemctl stop <service-name>** を実行します

例

優れた可視性サービスの場合は **systemctl stop tet-sensor**、適用サービスの場合は **systemctl stop tet-enforcer** です。

サービスの再起動

コマンド **systemctl restart <service-name>** を実行します。

例

詳細可視性サービスの場合は **systemctl restart tet-sensor** - 適用サービスの場合は **systemctl restart tet-enforcer**

サービスステータスの確認

コマンド **systemctl status <service-name>** を実行します。

例

優れた可視性サービスの場合は **systemctl status tet-sensor**、適用サービスの場合は **systemctl status tet-enforcer**

Windows サーバーまたは Windows VDI のサービス管理

サービスの開始

net start <service-name> コマンドを実行します。

例

優れた可視性サービスの場合は **net start tetsensor**：適用サービスの場合は **net start tetenforcer**

サービスの停止

コマンド**net stop <service-name>** を実行します。

例

優れた可視性サービスの場合は **net stop tetsensor**、適用サービスの場合は **net stop tetenforcer**

サービスの再起動

net stop <service-name> コマンドに続けて、**net start <service-name>** コマンドを実行します。

サービスステータスの確認

コマンド **sc query <service-name>** を実行します。

例

sc query tetsensor は優れた可視性サービス用です。**sc query tetenforcer** は適用サービス用です。

AIX のサービスの管理

サービスの開始

コマンド **startsrc -s <service-name>** を実行します。

例

詳細可視性サービスの場合は **startsrc -s tet-sensor**、適用サービスの場合は **startsrc -s tet-enforcer**

サービスの停止

コマンド **stopsrc -s <service-name>** を実行します。

例

優れた可視性サービスの場合は **stopsrc -s tet-sensor**、適用サービスの場合は **stopsrc -s tet-enforcer**

サービスの再起動

コマンド **stopsrc -s <service-name>** を実行し、次に **startsrc -s <service-name>** を実行します。

サービスステータスの確認

コマンド **lssrc -s <service-name>** を実行します。

例

lssrc -s tet-sensor は優れた可視性サービス用です。**lssrc -s tet-enforcer** は適用サービス用です。

Kubernetes Agent インストールでのサービス管理

サービスの開始/停止

エージェントは個別のサービスとしてではなく、クラスタ全体のデーモンセットとしてインストールされるため、特定のノードでエージェントを停止または開始することはできません。

ノード上のエージェントの再起動

ノードで Secure Workload エージェントのポッドを見つけ、適切な Kubernetes コマンドを実行して強制終了します。ポッドは自動的に再起動します。

ポッドステータスの確認

kubectl get pod -n tetration または **oc get pod -n tetration** （Openshift の場合）を実行すると、Kubernetes クラスタ内にあるすべての Secure Workload エージェントポッドのステータスが一覧で表示されます。

Secure Workload 適用エージェント

このセクションでは、Secure Workload 適用エージェントコンポーネント、メッセージングとインタラクション、UI 構成、およびトラブルシューティングについて説明します。

エージェントによるポリシーの適用

デフォルトでは、エージェントはポリシーを適用しません。準備ができたら、設定目的に基づいて、インストールされたエージェントが選択したホストにポリシーを適用するようにできます。

エージェントがポリシーを適用すると、ファイアウォールが送信元、宛先、ポート、プロトコル、方向などのパラメータに基づいて特定のネットワークトラフィックを許可するかドロップするかを指定する、順序付けられた一連のルールが適用されます。ポリシーの詳細については、「ポリシー」を参照してください。

適用エージェントは、エンドポイントに展開される軽量のプロセスです。適用エージェントは、適用フロントエンド（EFE）を介して、コントローラからセキュアなTCP/SSLチャネル経由でポリシーを受信します。受信したポリシーは、プラットフォームに依存しないスキーマ内にあります。適用エージェントは、プラットフォームに依存しないポリシーをプラットフォーム固有のポリシーに変換し、エンドポイントのファイアウォールをプログラムします。また、ファイアウォールの状態をアクティブに監視し、適用されたポリシーの逸脱を検出すると、キャッシュされたポリシーをファイアウォールに再度適用します。適用エージェントは、ファイアウォール全体を制御したり、ユーザーが設定したルールと連携して動作したりできます。ユーザーが設定したルールを Secure Workload ポリシーと共存させるための設定オプションがあります。適用エージェントは特権ドメインで実行されます。適用エージェントは、Linux マシンでは root として実行され、Windows マシンでは SYSTEM として実行されます。適用エージェント は、CPU やメモリなどのシステムリソースの消費も監視し、UI で有効になっている場合にのみ、エンドホストにポリシーを適用します。ポリシーの詳細については、「ポリシー」を参照してください。

エージェントは、セキュアな TCP/SSL チャネルを介してポリシーを受け取ります。

エージェントは特権ドメインで実行されます。エージェントは、Linux マシンでは root として実行され、Windows マシンでは SYSTEM として実行されます。

プラットフォームによっては、ポリシーの適用が有効になっている場合、エージェントはファイアウォールを完全に制御したり、既存の設定済みルールと連携して動作したりできます。

適用オプションの詳細、およびエージェントを有効にしてポリシーを適用するように設定する方法については、「エージェント設定プロファイルの作成」を参照してください。

エージェントと適用ステータスの監視

エージェントのステータスの確認

- コントローラとの通信

  適用エージェントは、TLS/SSLプロトコルを介した双方向の安全なチャネルを介してEFEと通信します。コントローラからのメッセージは、ポリシーの生成者によって署名され、適用エージェントによって検証されます。

- Secure Workload ネットワーク ポリシー メッセージ

Secure Workload ネットワークポリシーは、ホストに適用される有効なインテントに対する、一連の具体的なルールです。次のセクションで構成されています。

- [ファイアウォールルール（Firewall Rules）]：送信元、宛先、ポート、プロトコル、方向などのパラメータに基づいて、ファイアウォールが特定のネットワークトラフィックを許可またはドロップするかどうかを指定する、順序付けされた一連のルールです。エージェントは、（入力/出力および IPv4/IPv6 の両方について）コントローラが受け取った順序に従ってルールをプログラムします。
- [キャッチオールルール（Catch-all Rules）]：明示的に指定されたルールに一致しないトラフィックをカバーする、各方向の ALLOW または DROP のデフォルトアクションです。

- Secure Workload エージェント設定メッセージ

  コントローラは、適用エージェントの動作を制御するためのさまざまなフラグを含む、エージェント設定メッセージを送信します。これらのフラグについては、次のように説明されています。

  - [適用の有効化（enable enforcement）]：このフラグが設定されている場合、適用エージェントはファイアウォールに Secure Workload ルールを適用する準備ができています。コントローラへの接続を許可するゴールデンルールをプログラムし、後述のルール保持フラグに応じて他のファイアウォール状態をクリアします。最新の既知のポリシーを受信した場合、適用エージェントは有効化されるとすぐにそのポリシーを適用します。適用の有効化フラグが設定されていない場合（デフォルト）、適用エージェントはアイドル状態です。適用が有効化され、その後無効化された場合、適用エージェントによりファイアウォールの状態がクリアされ、キャッチオールのデフォルトアクションが ALLOW に設定されます。
  - [ルールの保持（preserve rules）]：このフラグが設定されている場合、適用エージェントは Secure Workload ルールのみを制御し、これらのルールはファイアウォールでユーザーが設定したルールと共存します。このフラグが設定されていない場合、適用エージェントはファイアウォール全体を制御し、Secure Workload ルールのみがファイアウォールで維持されます。
  - [ブロードキャストの有効化（enable broadcast）]：このフラグが設定されている場合（デフォルト）、適用エージェントはファイアウォールをプログラムし、入力および出力ブロードキャスト トラフィックを許可します。
  - [マルチキャストの有効化（enable multicast）]：このフラグが設定されている場合（デフォルト）、適用エージェントはファイアウォールをプログラムし、入力および出力マルチキャストトラフィックを許可します。
  - [Windows適用モード（windows enforcement mode）]：Windows 適用モードは、WAF（デフォルトの適用モード）または WFP に設定できます。WAF モードでは、Windows の高度なファイアウォールを使用してネットワークポリシーが適用されます。WFP モードでは、Windows フィルタエンジンで WFP フィルタを直接プログラミングすることにより、ネットワークポリシーが適用されます。

- エージェントからコントローラへのレポート

  適用エージェントは、EFE経由で定期的にステータスと統計レポートをコントローラに送信します。ステータスレポートには、プログラムされた最新のポリシーステータス（成功、失敗、ある場合はエラー）が含まれます。統計レポートには、プラットフォームに応じたポリシー統計（許可またはドロップされたパケットおよびバイト数）が含まれます。

UI 設定

エージェント設定プロファイル

エージェント設定プロファイルを設定するには、次の手順を実行します。

ステップ 1 左上隅にある [設定（Settings）] をクリックします。

ステップ 2 [エージェント設定（Agent Config）] をクリックします

ステップ 3 [ソフトウェアエージェント設定（Software Agent Config）] タブで、[プロファイルの作成（Create Profile）] をクリックします。

ステップ 4 [プロファイルの作成（Create Profile）] で、名前を入力し、[適用の有効化（Enforcement Enable）] を選択します。ユーザーがファイアウォールルールの保持を希望する場合は、[ルールの保持の有効化（Preserve Rules Enable）] を選択します。ユーザーがブロードキャストまたはマルチキャストトラフィックの許可を希望する場合は、それぞれ [ブロードキャストを許可（Allow Broadcast）] または [マルチキャストを許可（Allow Multicast）] を選択します。

ステップ 5 [保存（Save）] をクリックして、エージェント設定プロファイルを作成します。新しいプロファイルは、エージェント設定プロファイルの下にリストされます。

図 *8 :* エージェントへの設定プロファイルの適用

エージェント構成インテントを設定するには、次の手順を実行します。

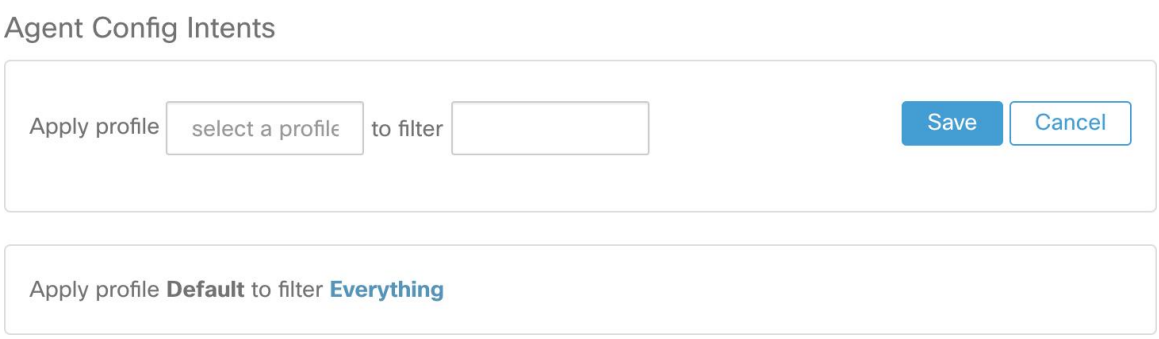
ステップ 1 [ソフトウェアエージェント構成（Software Agent Config）] ページで、[インテントの作成（Create Intent）] をクリックします。

ステップ 2 [プロファイルの適用（Apply Profile）] で、[エージェント構成プロファイル（Agent Config Profiles）] の下にリストされているプロファイルを入力し、フィルタを選択します。

ステップ 3 フィルタがまだ作成されていない場合は、[新しいフィルタの作成（Create new filter）] をクリックして新しいフィルタを作成します。[名前（Name）]、[説明（Description）]、[クエリ（Query）]、および [範囲（Scope）] を入力します。

ステップ 4 [保存（Save）] をクリックすると、[エージェント構成インテント（Agent Config Intents）] の下に新しいエントリが作成されます。

図 *9*: エージェントステータスのモニタリング



適用エージェントの確認

ステップ 1 右上隅にあるハート型のボタンをクリックして、[エージェント（Agents）] を選択します。

ステップ 2 [エージェント（Agents）] ページで、[適用エージェント（Enforcement Agents）] をクリックします。

ステップ 3 [適用エージェント（Enforcement Agents）] ページでは、CPU オーバーヘッド、帯域幅オーバーヘッド、エージェントの状態、ソフトウェア アップデート ステータス、エージェント ソフトウェア バージョンの配布、エージェント OS の配布を確認できます。

図 *10 : [*エージェント（*Agents***）*]* ページ**

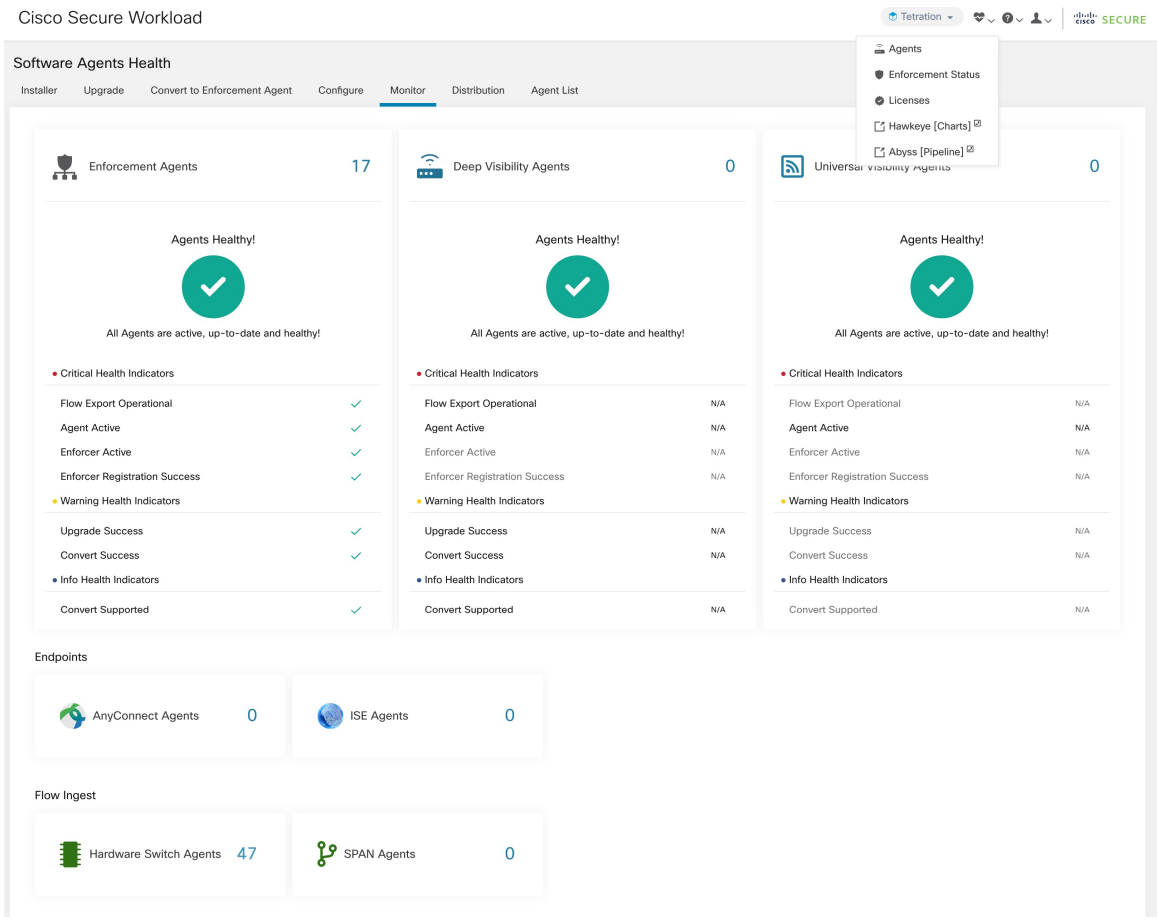
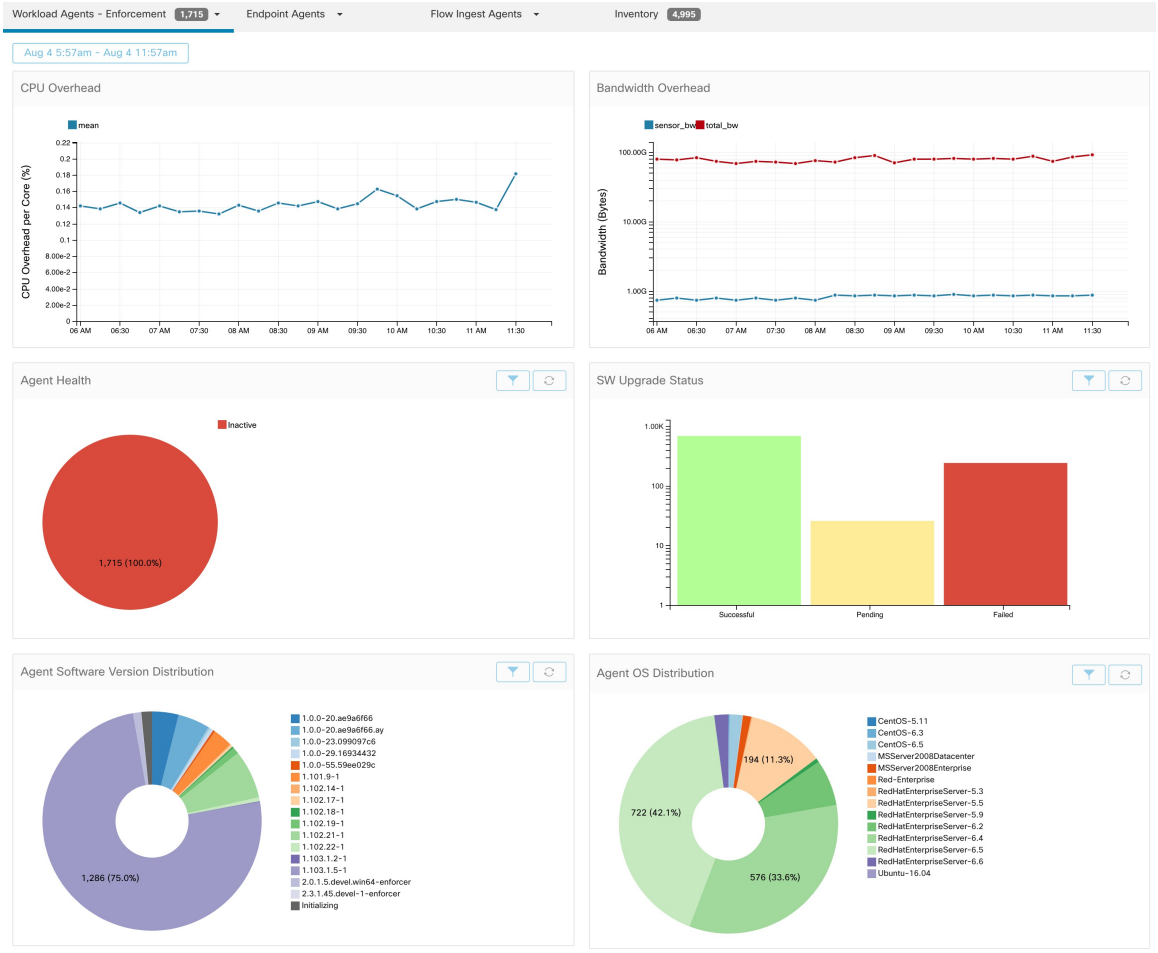


図 *11 : [*適用エージェント（***Enforcement Agents***）*]* ページ



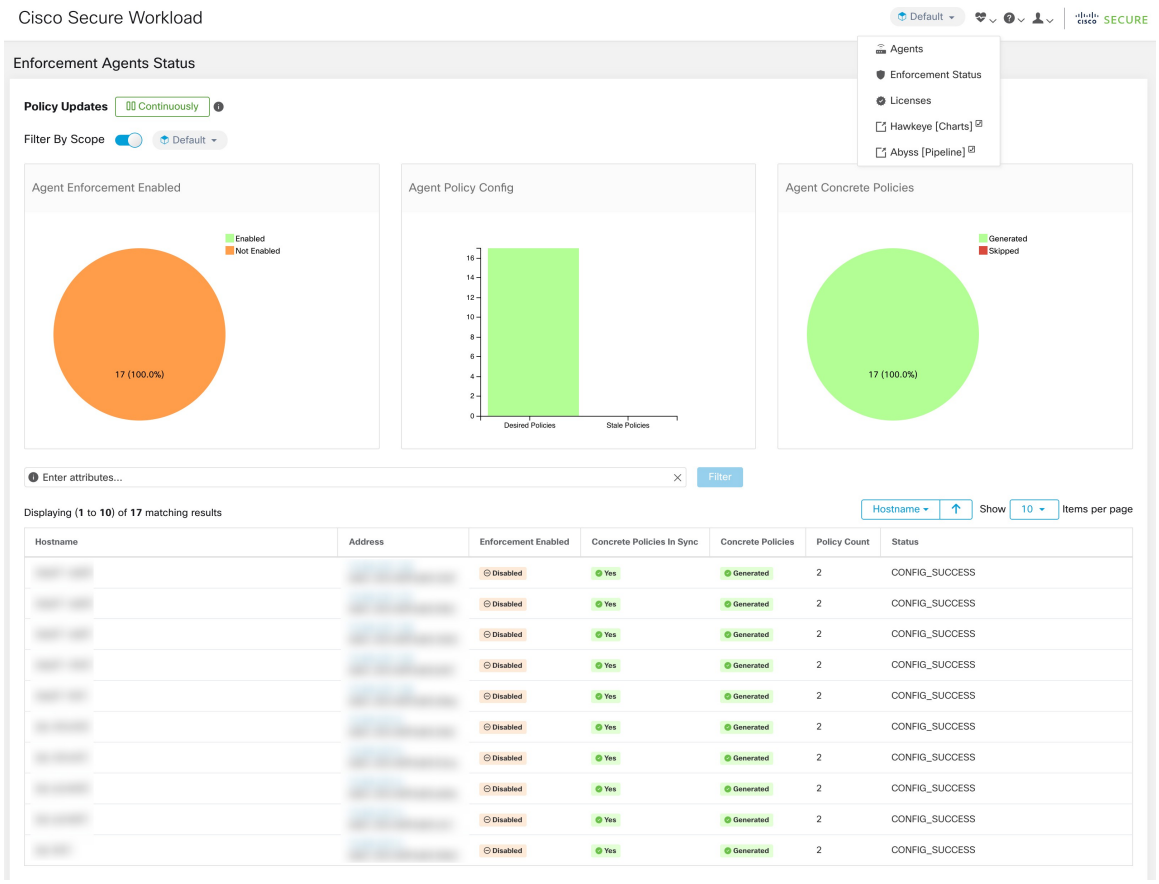
適用ステータスの確認

ステップ 1 右上隅にあるハート型のボタンをクリックして、[適用ステータス（Enforcement Status）] を選択します。

ステップ 2 [適用エージェントのステータス（Enforcement Agent Status）]ページで、適用が有効かどうか、エージェントポリシー設定、および適用が有効になっているエージェントのリストを確認できます。

ステップ 3 リストから適用エージェントのいずれかをクリックして、IP アドレス、範囲、インベントリタイプ、適用グループ、実験グループ、ユーザーラベル、トラフィック量（合計バイト/合計パケット）などのエージェントの詳細を表示します。IPアドレスをクリックすると、以下で言及されている詳細なエージェントステータスが表示されます。

図 *12* : 適用ステータス



ワークロードプロファイルでの詳細なエージェントステータスの表示

ステップ 1 上記の手順に従って、エージェントのステータスを確認します。

ステップ 2 [適用エージェント（Enforcement Agents）] ページで、[エージェントOS分布（Agent OS Distribution）] をクリックします。OS を選択し、ボックスの右上隅にあるフィルタイメージをクリックします。

ステップ 3 [ソフトウェアエージェントリスト（Software Agents Agent List）] ページに、エージェントと選択した OS の分布が一覧表示されます。

ステップ 4 [エージェント（Agent）] をクリックすると、[エージェントの詳細（Agent Details）] セクションが表示されます。IP アドレスをクリックして、[ワークロードプロファイル（Workload Profile）] ページに移動します。

ステップ 5 [ワークロードプロファイル（Workload Profile）]ページでは、ホストプロファイル、エージェントプロファイルや、帯域幅、長期プロセス、パッケージ、プロセススナップショット、設定、インターフェイス、統計、ポリシー、コンテナポリシーなど、他のエージェント固有の詳細情報が表示されます。

ステップ 6 [設定（Config）] タブをクリックして、エンドホストの設定を表示します。

ステップ 7 [ポリシー（Policies）] タブをクリックして、エンドホストに適用されたポリシーを表示します。

図 *13* : ワークロードプロファイル - 設定

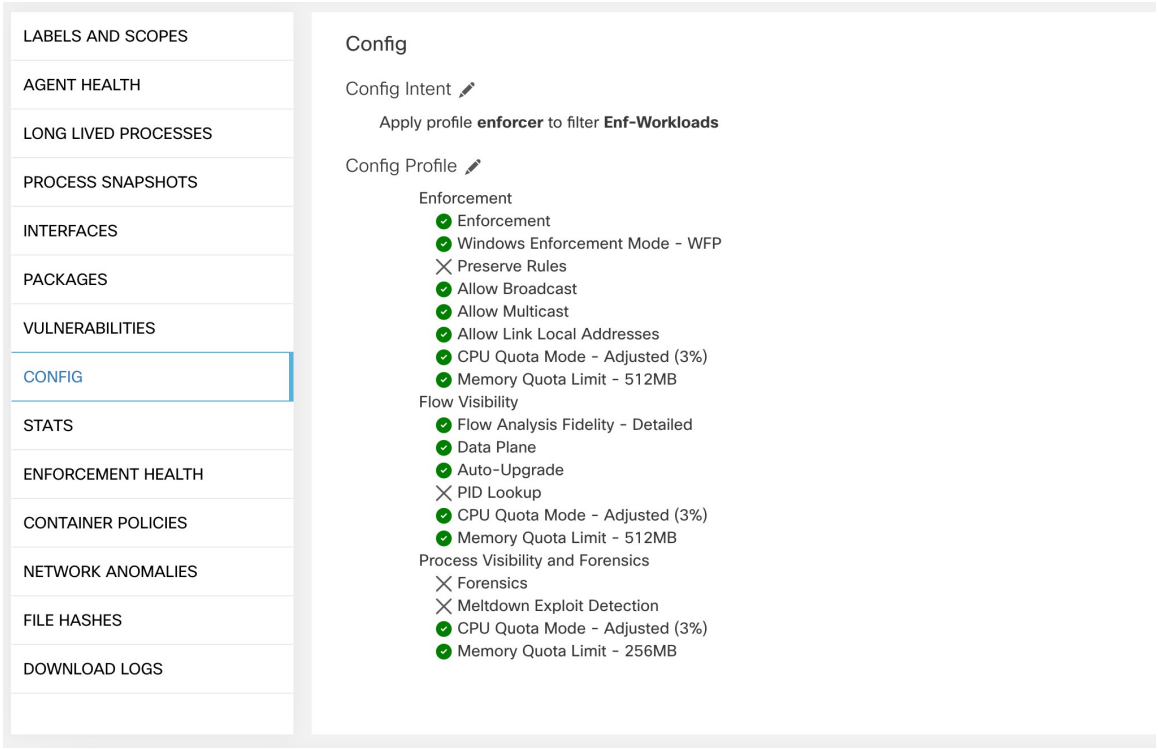
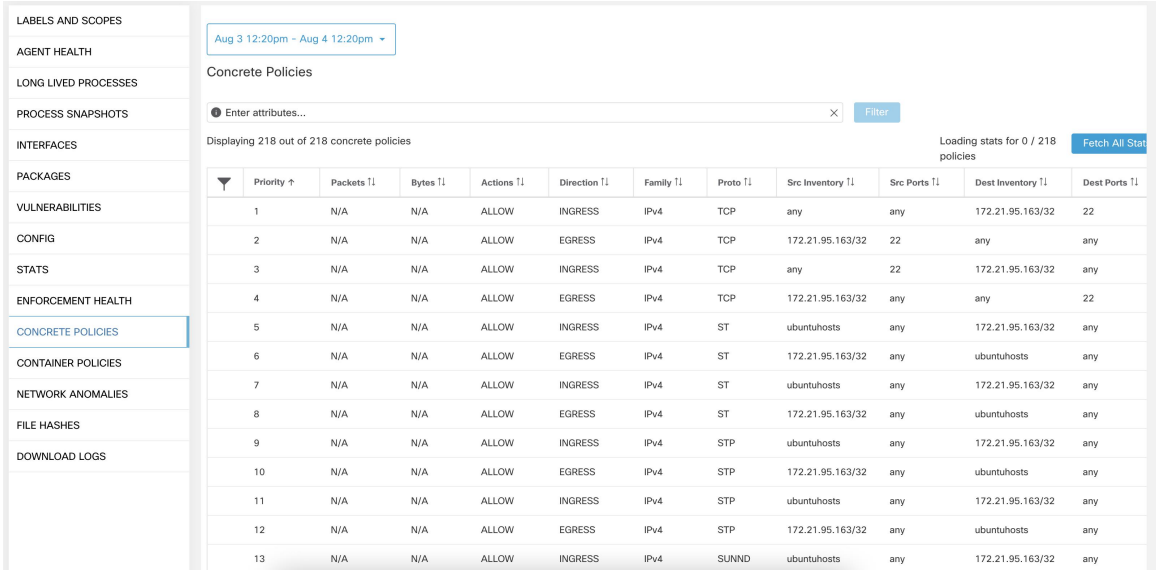


図 *14* : ワークロードプロファイル - ポリシー



適用が有効である場合のホスト IP アドレスの変更

適用が有効になっている場合にホストの IP アドレスを変更すると、ホスト IP がホスト ファイアウォールルールで認識され、Catch All が拒否に設定されている場合に影響が出る可能性があります。このシナリオでは、次の手順でホスト IP アドレスを変更することをお勧めします。

ステップ 1 Secure Workload UI で、適用を無効にして新しいエージェント構成プロファイルを作成します。

ステップ 2 IP アドレスを変更する必要があるホスト、およびその古い IP アドレスと新しい IP アドレスのリストを含むインテントを作成します。

ステップ 3 新しく作成したエージェント構成プロファイルをこのインテントに適用し、インテントを保存します。

ステップ 4 これらの選択されたホストでは、適用を無効にしておく必要があります。

ステップ 5 これらのホストの IP アドレスを変更します。

ステップ 6 Secure Workload UI で、範囲内のフィルタをこれらのホストの新しい IP アドレスで更新します。

ステップ 7 [エージェントワークロードプロファイル（Agent Workload Profile）]ページの[インターフェイス（Interfaces）]タブで IP アドレスの変更を確認します。[ポリシー（Policies）] タブで、ポリシーが新しい IP アドレスで生成されていることを確認します。

ステップ 8 上記で作成したインテントおよびプロファイルを削除します。

ステップ 9 範囲の元のエージェント構成プロファイルで適用が無効になっている場合は、適用を有効にします。

Linux プラットフォームの Secure Workload 適用

Linux プラットフォームでは、Secure Workload 適用エージェントは iptables/ip6tables/ipset を使用してネットワークポリシーを適用します。適用エージェントは、エンドホストで有効になると、デフォルトで iptables を制御およびプログラムします。IPv6 ネットワークスタックが有効になっている場合は、ip6tables を介して IPv6 ファイアウォールを制御します。

Linux プラットフォームでのエージェントの適用

Linux プラットフォームでは、エージェントは iptables/ip6tables/ipset を使用してネットワークポリシーを適用します。エンドホストでエージェントを有効にすると、デフォルトで iptables を制御およびプログラムします。IPv6 ネットワークスタックが有効になっている場合は、ip6tables を介して IPv6 ファイアウォールを制御します。

Linux iptables/ip6tables

Linux カーネルには、IPv4 および IPv6 パケットフィルタルールのテーブルをセットアップ、維持、および検査するために使用される iptables および ip6tables があります。これは、さまざまな事前定義されたテーブルで構成されています。各テーブルには、事前定義されたチェーンが含まれており、ユーザー定義のチェーンを含めることもできます。これらのチェーンには一連のルールが含まれており、これらの各ルールでパケットの一致基準を指定します。事前定義さ

れたテーブルには、raw、mangle、filter、nat が含まれます。事前定義されたチェーンには、INPUT、OUTPUT、FORWARD、PREROUTING、および POSTROUTING が含まれます。

Secure Workload エージェントは、パケットを許可またはドロップするルールを含むフィルタテーブルをプログラムします。フィルタテーブルは、事前定義されたチェーンである INPUT、OUTPUT、および FORWARD で構成されます。これらに加えて、エージェントはカスタム TA チェーンを追加して、コントローラからポリシーを分類および管理します。これらの TA チェーンには、ポリシーから派生した Secure Workload ルールと、エージェントによって生成されたルールが含まれています。エージェントは、プラットフォームに依存しないルールを受信すると、それらを解析して iptable/ip6table/ipset ルールに変換し、これらのルールをフィルタテーブルの TA 定義チェーンに挿入します。ファイアウォールのプログラミング後、適用エージェントはファイアウォールを監視し、ルール/ポリシーの逸脱がないか確認します。逸脱がある場合は、ファイアウォールを再プログラミングします。また、ファイアウォールでプログラムされたポリシーを追跡し、ポリシーのステータスを定期的にコントローラに報告します。

この動作を表す例を次に示します。

プラットフォームに依存しないネットワーク ポリシー メッセージの一般的なポリシーの構成は次のとおりです。

```
source set id: “test-set-1”
destination set id: “test-set-2”
source ports: 20-30
destination ports: 40-50
ip protocol: TCP
action: ALLOW
. . .
set_id: “test-set-1”
     ip_addr: 1.2.0.0
     prefix_length: 16
     address_family: IPv4
set_id: “test-set-2”
     ip_addr: 3.4.0.0
     prefix_length: 16
     address_family: IPv4
```

エージェントは、他の情報とともに、このポリシーを処理し、プラットフォーム固有の ipset および iptabel ルールに変換します。

```
ipset rule:
Name: ta_f7b05c30ffa338fc063081060bf3
Type: hash:net
Header: family inet hashsize 1024 maxelem 65536
Size in memory: 16784
References: 1
Members:
1.2.0.0/16
Name: ta_1b97bc50b3374829e11a3e020859
Type: hash:net
Header: family inet hashsize 1024 maxelem 65536
Size in memory: 16784
References: 1
Members:
3.4.0.0/16
iptables rule:
TA_INPUT -p tcp -m set --match-set ta_f7b05c30ffa338fc063081060bf3 src -m set --match-
‹→set ta_1b97bc50b3374829e11a3e020859 dst -m multiport --sports 20:30 -m multiport --
‹→dports 40:50 -j ACCEPT
```

警告

ipset カーネルモジュール

エージェント構成プロファイルで [適用（Enforcement）] が有効で、[ルールの維持（Preserve Rules）] が無効になっている場合、Linux ホストで実行されている対象のエージェントにより、ipset カーネルモジュールの max_sets 設定が十分大きいことが確認されます。変更が必要な場合、エージェントは ipset カーネルモジュールを新しい max_sets 値でリロードします。代わりに [ルールの維持（Preserve Rules）] が有効になっている場合、エージェントにより現在の ipset モジュールの max_sets 値がチェックされますが、変更はされません。現在設定されている max_sets 値は、cat /sys/module/ip_set/parameters/max_sets で確認できます。

ホストファイアウォールのバックアップ

エージェント構成プロファイルで初めて [適用（Enforcement）] が有効になると、Linux ホストで実行されている対象のエージェントは、ホストのファイアウォールを制御する前に、ipset および ip[6] テーブルの現在のコンテンツを /opt/cisco/tetration/backup に保存します。

適用設定の連続した無効化/有効化の切り替えでは、新しいバックアップは生成されません。ディレクトリは、エージェントのアンインストール時に削除されません。

WFP モードの Windows プラットフォームでの Secure Workload 適用

Windows プラットフォームでは、Secure Workload 適用エージェントは Windows ファイアウォールを使用してネットワークポリシーを適用します。

セキュリティが強化された Windows ファイアウォール

次のようなタイプの設定に基づいてネットワークトラフィックを規制する、Windows のネイティブコンポーネントです。

- インバウンド ネットワーク トラフィックを規制するファイアウォールルール
- アウトバウンド ネットワーク トラフィックを規制するファイアウォール ルール
- ネットワークトラフィックの送信元と宛先の認証ステータスに基づくファイアウォール オーバーライド ルール
- IPSec トラフィックと Windows サービスに適用されるルール。

Secure Workload ネットワークポリシーは、インバウンドおよびアウトバウンドのファイアウォールルールを使用してプログラムされます。

Secure Workload ルールと Windows ファイアウォール

Windows プラットフォームでは、Secure Workload ネットワークポリシーは次のように適用されます。

1. プラットフォームに依存しないファイアウォールルールを、Secure Workload ネットワークポリシーから Windows ファイアウォールルールに変換します。
2. Windows ファイアウォールでルールをプログラムします。
3. Windows ファイアウォールがルールを適用します。
4. Windows ファイアウォールとそのルールセットの状態をモニターします。変更が検出された場合は、逸脱をレポートし、Windows ファイアウォールで Secure Workload ネットワークポリシーをリセットします。

セキュリティ プロファイル（Security Profiles）

Windows ファイアウォールは、ホストが現在接続しているネットワークに基づいてルールをグループ化します。これらはプロファイルと呼ばれ、次のような 3 つのプロファイルがあります。

- ドメインプロファイル
- プライベートプロフィール
- パブリックプロファイル

Secure Workload ルールはすべてのプロファイルにプログラムされていますが、アクティブなプロファイル内のルールのみが継続的に監視されます。

効果的な設定および混合リストのポリシー

Windows ファイアウォールのルールセットは、優先順位に基づいて順序付けられていません。複数のルールが 1 つのパケットに一致する場合、それらのルールのうち最も制限の厳しいものが有効になります。つまり、拒否ルールが許可ルールより優先されます。詳細については、「Microsoft TechNet に関する記事」[英語] を参照してください。

エージェント適用セクションからの混合リスト（許可および拒否の両方）ポリシーの例について考えてみます。

```
1. ALLOW 1.2.3.30 tcp port 80
2. ALLOW 1.2.3.40 udp port 53
3. BLOCK 1.2.3.0/24 ip
4. ALLOW 1.2.0.0/16 ip
5. Catch-all: DROP ingress, ALLOW egress
```

ホスト 1.2.3.30 の TCP ポート 80 宛てのパケットがファイアウォールに到達すると、上記のすべてのルールに一致しますが、最も制限の厳しいルール 3 が適用され、パケットはドロップされます。この動作は、ルールが順番に評価されてルール 1 が適用され、パケットが許可されるという期待に反します。

この動作の違いは、前述の Windows ファイアウォールの設計により、Windows プラットフォームで予期されるものです。この動作は、異なるルールアクションが設定された重複するルールを持つ混合リストポリシーで観察できます。

次に例を示します。

```
1. ALLOW 1.2.3.30 tcp
2. BLOCK 1.2.3.0/24 tcp
```

他のファイアウォールやポリシーからの干渉

Secure Workload ネットワークポリシーを意図したとおりに適用するには、エージェントに Windows ファイアウォールの完全かつ排他的な制御を許可することを推奨します。次の場合、エージェントはポリシーを確実に適用できません。

- サードパーティのファイアウォールが存在する (Windows ファイアウォールは、ホスト上でアクティブなファイアウォール製品である必要があります）。
- ファイアウォールが現在のプロファイルに対して無効になっている。
- 競合するファイアウォール設定がグループポリシーを使用して展開されている。以下の設定で競合が発生する場合があります。
  - ファイアウォールルール
  - 現在のプロファイルに設定されたデフォルトのインバウンドまたはアウトバウンドアクションがポリシーのキャッチオールルールとは異なる場合
  - ファイアウォールが現在のプロファイルで無効になっている場合

ステートフル適用

Windows Advanced Firewall は**ステートフル**ファイアウォールと見なされます。つまり、特定のプロトコル（TCP など）に対して、ファイアウォールは内部状態の追跡を維持して、ファイアウォールに到達する新しいパケットが既知の接続に属しているかどうかを検出します。既知の接続に属するパケットは、ファイアウォールルールを調べることなく許可されます。これにより、着信テーブルと発信テーブルの両方でルールを確立することなく、双方向通信が可能になります。

たとえば、Web サーバーについて、**ポート 443 へのすべての TCP 接続を受け入れる** というルールを考えてみます。

意図は明確です。ポート 443 でサーバーへのすべての TCP 接続を受け入れ、サーバーがクライアントに向けて通信できるようにします。この場合、着信テーブルには、ポート 443 での TCP 接続を許可するルール 1 つのみを挿入します。発信テーブルに挿入する必要のあるルールはありません。これは Windows Advanced Firewall によって暗黙的に行われるためです。

状態追跡は、明示的な**接続**が確立および維持される一部のプロトコルにのみ適用されることに注意してください。他のプロトコルの場合、双方向通信を可能にするために、着信ルールと発信ルールの両方をプログラムする必要があります。

適用が有効な場合、プロトコルが TCP の場合、特定の具象ルールは**ステートフル**としてプログラムされます（エージェントは、コンテキストに基づいて、ルールを着信テーブルまたは発信テーブルのどちらに挿入するかを決定します）。他のプロトコル（**ANY** を含む）の場合、着信ルールと発信ルールの両方がプログラムされます。

警告

ホストファイアウォールのバックアップ

エージェント設定プロファイルで初めて適用が有効になると、Windowsホストで実行中の対象エージェントは、ホストのファイアウォール制御を開始する前に、現在の Windows 高度ファイアウォールのコンテンツを *Program- Data\Cisco\Tetration\backup* にエクスポートします。適用の設定を続けて無効化または有効化しても、新しいバックアップは生成されません。このディレクトリは、エージェントのアンインストール時に削除されません。

WFP モードでの Windows プラットフォームにエージェントを適用

Windows プラットフォームでは、エージェントは WFP フィルタをプログラミングするネットワークポリシーを適用します。Windowsの高度なファイアウォールは、ネットワークポリシーの設定には使用されません。

WFP（Windows フィルタリング プラットフォーム）

WFP（Windows Filtering Platform）は、ネットワークトラフィック処理フィルタを設定するために Microsoft が提供する一連の API です。ネットワークトラフィック処理フィルタは、カーネルレベルの API およびユーザーレベルの API を使用して設定できます。WFP フィルタは、ネットワークレイヤ、トランスポートレイヤ、アプリケーションレイヤの適用（ALE）など、さまざまなレイヤで設定できます。Secure Workload WFP フィルタは、Windows ファイアウォールルールと同様に、ALE レイヤで設定されます。それぞれのレイヤには、重みの高いものから順に並べられた、いくつかのサブレイヤがあります。各サブレイヤ内で、フィルタは重みの高いものから順に並べられます。ネットワークパケットは、すべてのサブレイヤを通過します。各サブレイヤで、ネットワークパケットは、重みに基づいて一致するフィルタを通過し、許可またはブロックのアクションを返します。すべてのサブレイヤを通過した後、アクションに基づいてパケットが処理されます。ブロックアクションは許可をオーバーライドします。

WAF における WFP の利点

- Windows ファイアウォール設定の依存関係を回避
- GPO 制限なし
- 移行とポリシー復元の容易さ
- ユーザーによるポリシー順序の制御を実現
- Windows ファイアウォールの厳格なブロック優先のポリシー順序を回避
- ポリシー更新時の CPU オーバーヘッドの削減
- 効率的な 1:1 ポリシールールフィルタの作成
- 高速なシングルステップ更新

エージェントによる WFP のサポート

適用が WFP を使用するように設定されている場合、Secure Workload フィルタは Windows ファイアウォールルールをオーバーライドします。

WFP モードで、エージェントはさまざまな WFP オブジェクトを設定します。

- プロバイダー：フィルタ管理に使用されます。パケットフィルタリングには影響しません。GUID と名前が指定されます。
- サブレイヤー：サブレイヤーには、名前、GUID、および重みが指定されます。Secure Workload サブレイヤーは、Windows Advanced Firewall サブレイヤーよりも大きな重み付けで設定されます。
- フィルタ：フィルタには、名前、GUID、ID、重み、レイヤー ID、サブレイヤーキー、アクション（PERMIT/BLOCK）、およびフィルタ条件があります。WFP フィルタは、ゴールデンルール、セルフルール、ポリシールール用に設定されています。エージェントは、ポートスキャン防止フィルタも設定します。Secure Workload フィルタは、フラグ FWPM_FILTER_FLAG_CLEAR_ACTION_RIGHT を使用して設定されます。このフラグのため、Secure Workload フィルタのアクションを Microsoft ファイアウォールルールによって上書きすることはできません。Secure Workload ネットワークポリシールールごとに、方向（インバウンド/アウトバウンド）とプロトコルに基づいて、1 つ以上の WFP フィルタが設定されます。

TCP インバウンドポリシーの場合：

```
id: 14 , TCP Allow 10.195.210.184 Dir=In localport=3389
```

設定された WFP フィルタ

```
Filter Name:                  Secure Workload Rule 14
-----------------------------------------------------
EffectiveWeight:                        18446744073709551589
LayerKey:                               FWPM_LAYER_ALE_AUTH_LISTEN_V4
Action:                                 Permit
Local Port:                             3389
Filter Name:                            Secure Workload Rule 14
-----------------------------------------------------
EffectiveWeight:                        18446744073709551589
LayerKey:                               FWPM_LAYER_ALE_AUTH_RECV_ACCEPT_V4
Action:                                 Permit
RemoteIP:                               10.195.210.184-10.195.210.184
```

Secure Workload エージェントは、インバウンド CATCH-ALL ポリシーの **Cisco Secure Workload デフォルトインバウンド**フィルタを設定します。Secure Workload エージェントは、アウトバウンド CATCH-ALL ポリシーの **Cisco Secure Workload デフォルトアウトバウンド**フィルタを設定します。

エージェント WFP のサポートと Windows ファイアウォール

- エージェントは、WAF ルールまたは WAF プロファイルを**監視しません**。
- エージェントはファイアウォールの状態を**監視しません**。
- エージェントでは、ファイアウォールの状態を有効にする**必要はありません**。

• エージェントは GPO ポリシーと競合しません。

効果的な設定および混合リストのポリシー

WFP モードでのエージェント適用は、混合リストまたはグレーリストのポリシーをサポートします。

エージェント適用セクションからの混合リスト（許可および拒否の両方）ポリシーの例を考えてみます。

```
1. ALLOW 1.2.3.30 tcp port 80-            wt1000
2. BLOCK 1.2.3.0/24 ip-                   wt998
3. ALLOW 1.2.0.0/16 ip-                   wt997
4. Catch-all: DROP ingress, ALLOWegress - wt996
```

ホスト 1.2.3.30 tcp ポート 80 に向かうパケットがファイアウォールに到達すると、ルール 1 に一致します。しかし、ホスト 1.2.3.10 に向かうパケットは、フィルタ 2 のためにブロックされます。ホスト 1.2.2.10 に向かうパケットは、フィルタ 3 によって許可されます。

ステートフル適用

Cisco Secure Workload の WFP フィルタは、ALE レイヤで構成されます。ネットワークトラフィックは、ソケットの connect()、listen()、および accept() 操作に対してフィルタリングされます。L4 接続に関連するネットワークパケットは、接続が確立されるとフィルタリングされなくなります。

構成された WFP フィルタの可視性

構成された Secure Workload WFP フィルタは、c:\program files\tetration\tetenf.exe を使用して表示できます。次のオプションがサポートされます。

- 「管理者」権限を使用して「cmd.exe」を実行します
- c:\program files\tetration\tetenf.exe -l -f <-verbose> <-output=outfile.txt> を実行する

または

- 「管理者」権限を使用して「cmd.exe」を実行します
- netsh wfp show filters を実行する
- 構成された Secure Workload フィルタについて filters.xml を確認する

WFP モードでステルスモードフィルタを無効にする

ステルスモードフィルタ（ポートスキャンフィルタ）を無効にする手順は次のとおりです。

ステップ 1 \conf\enforcer.cfg を編集します

ステップ 2 disable_wfp_stealth_mode: 1 を追加します。

ステップ 3 ファイルを保存します。

ステップ 4 管理者権限で TetEnforcer サービスを再起動します。

a) コマンド `sc stop tetenforcer` を実行して、TetEnforcer サービスを停止します。

b) コマンド `sc start tetenforcer` を実行して、TetEnforcer サービスを開始します。

ステップ 5 次の手順で確認します。

a) 「管理者」権限を使用してコマンド `cmd.exe` を実行します。

b) コマンド `c:\program files\tetration\tetenf.exe -l -f <-verbose> <-output=outfile.txt>` を実行します。

```
"Tetration Internal Rule block portscan" filters are not configured.
```

設定済みの WFP フィルタの削除

設定された Secure Workload WFP フィルタは、c:\program files\tetration\tetenf.exe を使用して表示できます。フィルタを誤って削除しないようにするには、ユーザーが [トークン（token）] を指定する必要があります。削除コマンドを実行する際、yyyyは現在の年、mmは現在の月を数値形式で表します。たとえば、今日の日付が 2021 年 1 月 21 日の場合、トークンは [-token=202101] になります。

次のオプションがサポートされます。

- 「管理者」権限を使用して「cmd.exe」を実行します。
- 設定されたすべての Secure Workload フィルタを削除するには、c:\program files\tetration\tetenf.exe -d -f -all - token=<yyyymm> を実行します。
- 設定されたすべての Secure Workload WFP オブジェクトを削除するには、c:\program files\tetration\tetenf.exe -d -all -token=<yyyymm> を実行します。
- Secure Workload WFP フィルタを名前で削除するには、c:\program files\tetration\tetenf.exe -d -name=<WFP filter name> -token=<yyyymm> を実行します。

既知の制限事項

- 適用モードが WFP に設定されている場合、エージェント設定プロファイルの [ルールの保持（Preserve Rules）] 設定は無効になります。

Windows OS ベースのフィルタリング属性

Windows ベースのワークロードにポリシーを適用する際の精度をさらに高めるために、次の方法でネットワークトラフィックをフィルタリングできます。

- アプリケーション
- サービス名（Service Name）

- ユーザー名（ユーザーグループありまたはなし）

これは、WAF および WFP モードでサポートされています。Windows OS ベースのフィルタは、生成されたネットワークポリシーでコンシューマフィルタとプロバイダーフィルタに分類されます。コンシューマフィルタは、コンシューマワークロードで開始されたネットワークトラフィックをフィルタリングします。プロバイダーフィルタは、プロバイダーワークロード宛てのネットワークトラフィックをフィルタリングします。

Windows OS ベースのフィルタの構成

該当するセグメンテーションポリシーで Windows OS ベースのフィルタを設定します。

始める前に

この手順は、既存のポリシーを変更していることが前提となります。Windows OS ベースのフィルタを追加するポリシーを作成していない場合は、最初にそのポリシーを作成します。

ステップ 1 [防御（Defend）] > [セグメンテーション（Segmentation）] の順に選択します。

ステップ 2 Windows OS ベースのフィルタを設定するポリシーを含む範囲をクリックします。

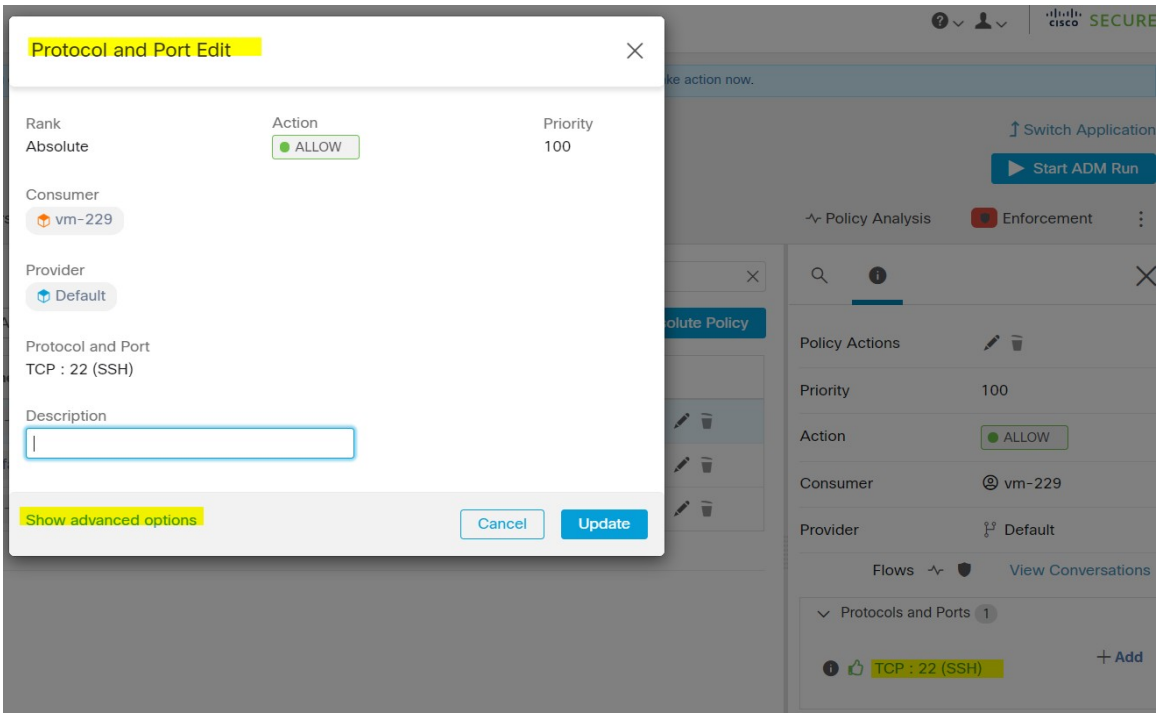
ステップ 3 ポリシーを編集するワークスペースをクリックします。

ステップ 4 [ポリシーの管理（Manage Policies）] をクリックします。

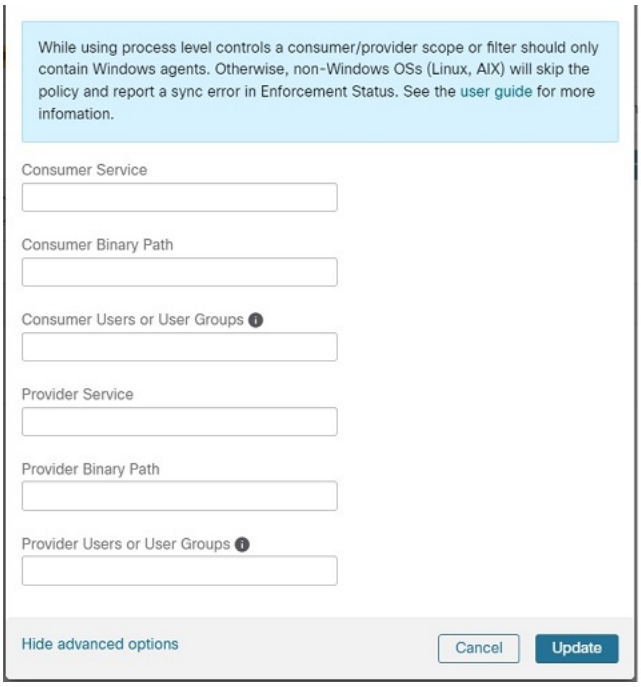
ステップ 5 編集するポリシーのテーブル行で、[プロトコルとポート（Protocols and Ports）] 列の既存の値をクリックします。

ステップ 6 右側のペインで、[プロトコルとポート（Protocols and Ports）] の下にある既存の値をクリックします。

この例では、[TCP : 22 (SSH)] をクリックします。



ステップ 7 [詳細オプションの表示（Show advanced options）] をクリックします。



ステップ 8 アプリケーション名、サービス名、またはユーザー名に基づいてコンシューマフィルタを設定します。

- アプリケーション名はフルパス名にする必要があります。
- サービス名は短いサービス名にする必要があります。

- ユーザー名は、ローカルユーザー名（tetter など）またはドメインユーザー名（sensor-dev@sensor-dev.com、sensor-dev\sensor-dev など）にできます。
- ユーザーグループは、ローカルユーザーグループ（管理者など）またはドメインユーザーグループ（domain users\\sensor-dev など）にできます。
- 複数のユーザー名および（または）ユーザーグループ名は「,」で区切って指定できます（例：sensor-dev\\@sensor-dev.com,domain users\\sensor-dev）。
- サービス名とユーザー名は同時に設定できません。

ステップ 9 アプリケーション名、サービス名、またはユーザー名に基づいてプロバイダーフィルタを設定します。

前のステップのコンシューマフィルタと同じガイドラインに従います。

ステップ 10 必要に応じてバイナリへのパスを入力します。

たとえば、**c:\test\putty.exe** と入力します。

ステップ 11 **[更新（Update）]**をクリックします。

ステップ 12 ポリシーを適用するには、[適用（Enforcement）]>[ポリシーの適用（Enforce Policies）]>[次へ（Next）]>[次へ（Next）]>[同意して適用（Accept and Enforce）] の順にクリックします。

Windows OS ベースのフィルタリング属性を使用したポリシーの確認とトラブルシューティング

Windows OS ベースのフィルタリング属性を使用する場合は、次のトピックを使用して、ポリシーが期待どおり動作することをワークロードで確認します。

Cisco TAC は、必要に応じてこの情報を使用して、このようなポリシーのトラブルシューティングを行うことができます。

アプリケーション名に基づくポリシー

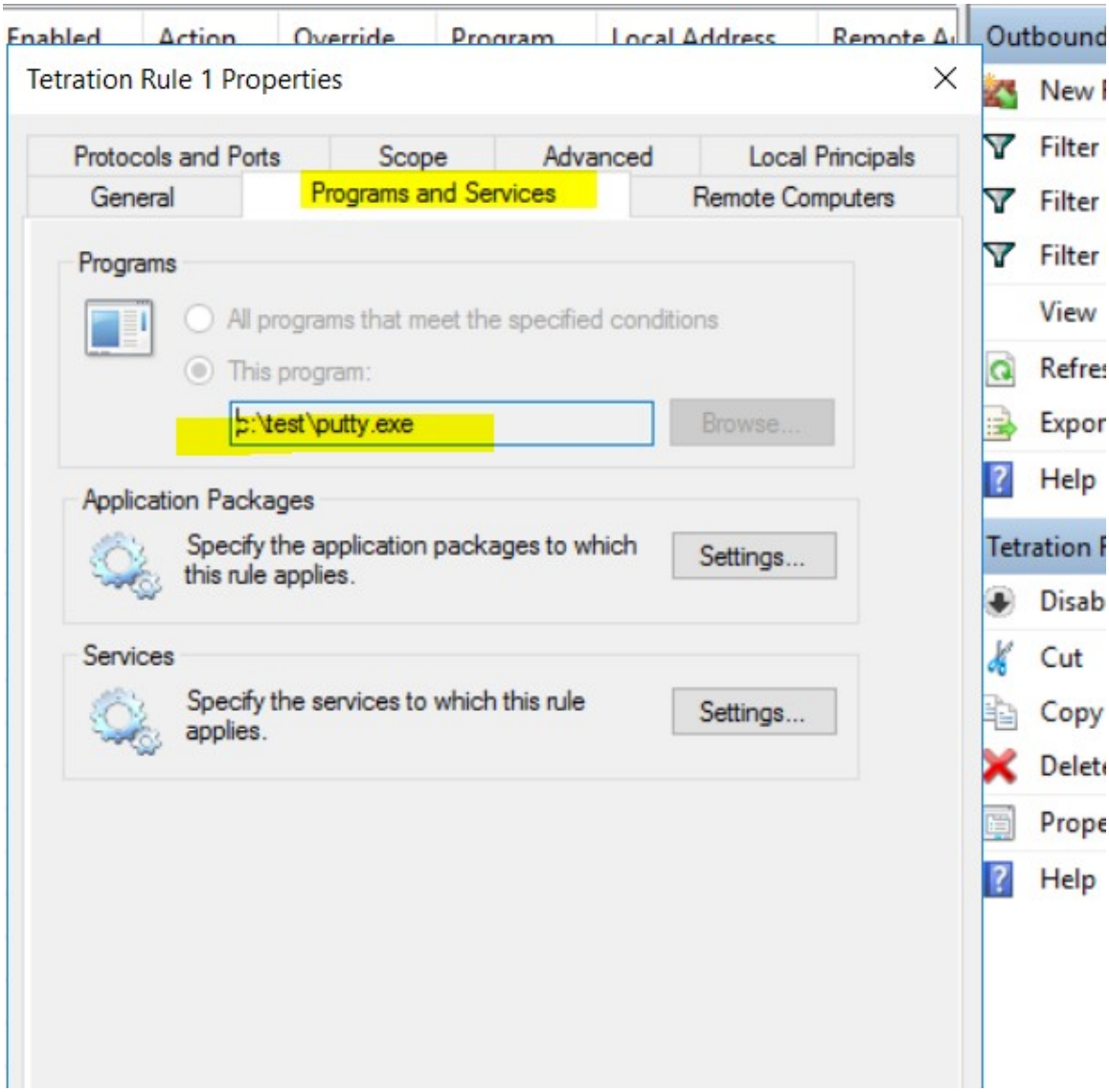
次の情報を使用して、アプリケーション名に基づく Windows OS ワークロードのポリシーを確認およびトラブルシューティングします。

次のセクションでは、**c:\test\putty.exe** として入力されたアプリケーションバイナリのワークロードにポリシーを表示させる方法について説明します。

アプリケーション名に基づくポリシーの例

```
dst_ports {
start_port: 22
end_port: 22
consumer_filters {
application_name: “c:\test\putty.exe”
}
}}
ip_protocol: TCP
address_family: IPv4
inspection_point: EGRESS
```

生成されたファイアウォールルール



netsh を使用して生成されたフィルタ

高度なポリシーにフィルタが追加されていることをネイティブの Windows ツールで確認するには、次の手順を実行します。

- 「管理者」権限を使用して「cmd.exe」を実行します
- 「netsh wfp show filters」を実行します
- 出力ファイル filters.xml が、現在のディレクトリに生成されます。
- 出力ファイル（filters.xml）の FWPM_CONDITION_ALE_APP_ID でアプリケーション名を確認します。

```
<fieldKey>FWPM_CONDITION_ALE_APP_ID</fieldKey>
                      <matchType>FWP_MATCH_EQUAL</matchType>
                      <conditionValue>
                             <type>FWP_BYTE_BLOB_TYPE</type>
                             <byteBlob>
                                    <data>
‹→5c00640065007600690063006500 5c006800610072006400640069007300 6b0076006f006
‹→</data>
                                 <asString>\device\harddiskvolume2\temp\putty.exe</
‹→asString>
                 </byteBlob>
        </conditionValue>
```

tetenf.exe -l -f を使用して生成された WFP フィルタ

```
Filter Name:                    Secure Workload Rule 1
---------------------------------------------------------
EffectiveWeight:                18446744073709551592
LayerKey:                       FWPM_LAYER_ALE_AUTH_CONNECT_V4
Action:                         Permit
RemoteIP:                       10.195.210.15-10.195.210.15
Remote Port:                    22
Protocol:                       6
AppID:                          \device\harddiskvolume2\test\putty.exe
```

アプリケーション名が無効な場合

- WAF モードでは、無効なアプリケーション名に対してファイアウォールルールが作成されます。
- WFP モードでは、無効なアプリケーション名に対して WFP フィルタは作成されませんが、NPCは拒否されません。エージェントは警告メッセージをログに記録し、残りのポリシールールを構成します。

サービス名に基づくポリシー

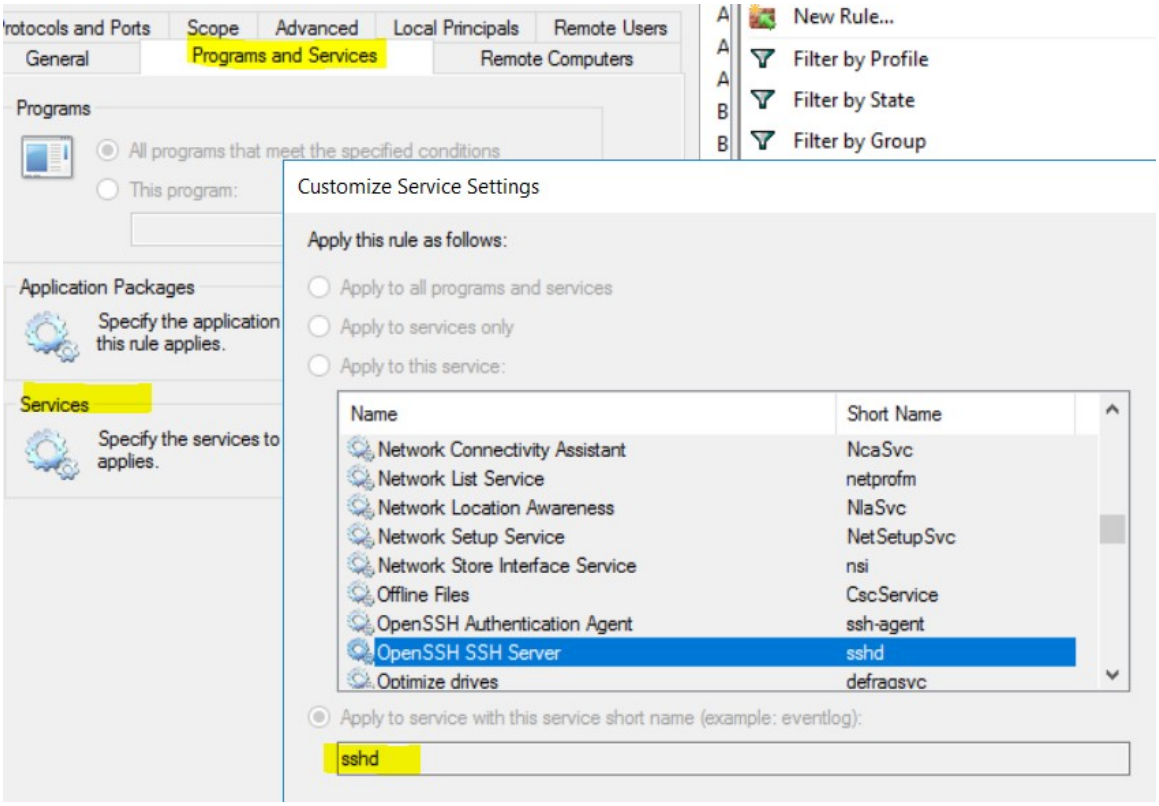
次の情報を使用して、サービス名に基づく Windows OS ワークロードのポリシーを確認およびトラブルシューティングします。

次のセクションでは、ワークロードにポリシーを表示させる方法について説明します。

サービス名に基づくサンプルポリシー

```
    dst_ports {
             start_port: 22
             end_port: 22
             provider_filters {
                    service_name: “sshd”
             }
         }}
         ip_protocol: TCP
         address_family: IPv4
         inspection_point: INGRESS
```

生成されたファイアウォールルール



netsh を使用して生成されたフィルタ

高度なポリシーにフィルタが追加されていることをネイティブの Windows ツールで確認するには、次の手順を実行します。

- 「管理者」権限を使用して「cmd.exe」を実行します
- 「netsh wfp show filters」を実行します
- 出力ファイル filters.xml が現在のディレクトリに生成されます
- 出力ファイル（filters.xml）でユーザー名の FWPM_CONDITION_ALE_USER_ID を確認します。

```
<item>
                    <fieldKey>FWPM_CONDITION_ALE_USER_ID</fieldKey>
                    <matchType>FWP_MATCH_EQUAL</matchType>
                    <conditionValue>
                              <type>FWP_SECURITY_DESCRIPTOR_TYPE</type>

<sd>O:SYG:SYD:(A;;CCRC;;;S-1-5-80-3847866527-469524349-687026318-
→516638107)</sd>
                    </conditionValue>
</item>
```

tetenf.exe -l -f を使用して生成された WFP フィルタ

```
Filter Name:       Secure Workload Rule 3
-----------------------------------------------------
EffectiveWeight:            18446744073709551590
LayerKey:                   FWPM_LAYER_ALE_AUTH_RECV_ACCEPT_V4
Action:                     Permit
Local Port:                 22
Protocol:                   6
User or Service:            NT SERVICE\sshd
```

サービス名が不正な場合

- WAF モードで、存在しないサービス名に対してファイアウォールルールが作成されます
- WFP モードでは、存在しないサービス名に対して WFP フィルタは作成されません
- サービス SID タイプは「無制限」または「制限付き」である必要があります。サービスの種類が「なし」の場合、ファイアウォールルールと WFP フィルタを追加できますが、効果はありません。

  SID タイプを確認するには、次のコマンドを実行します。

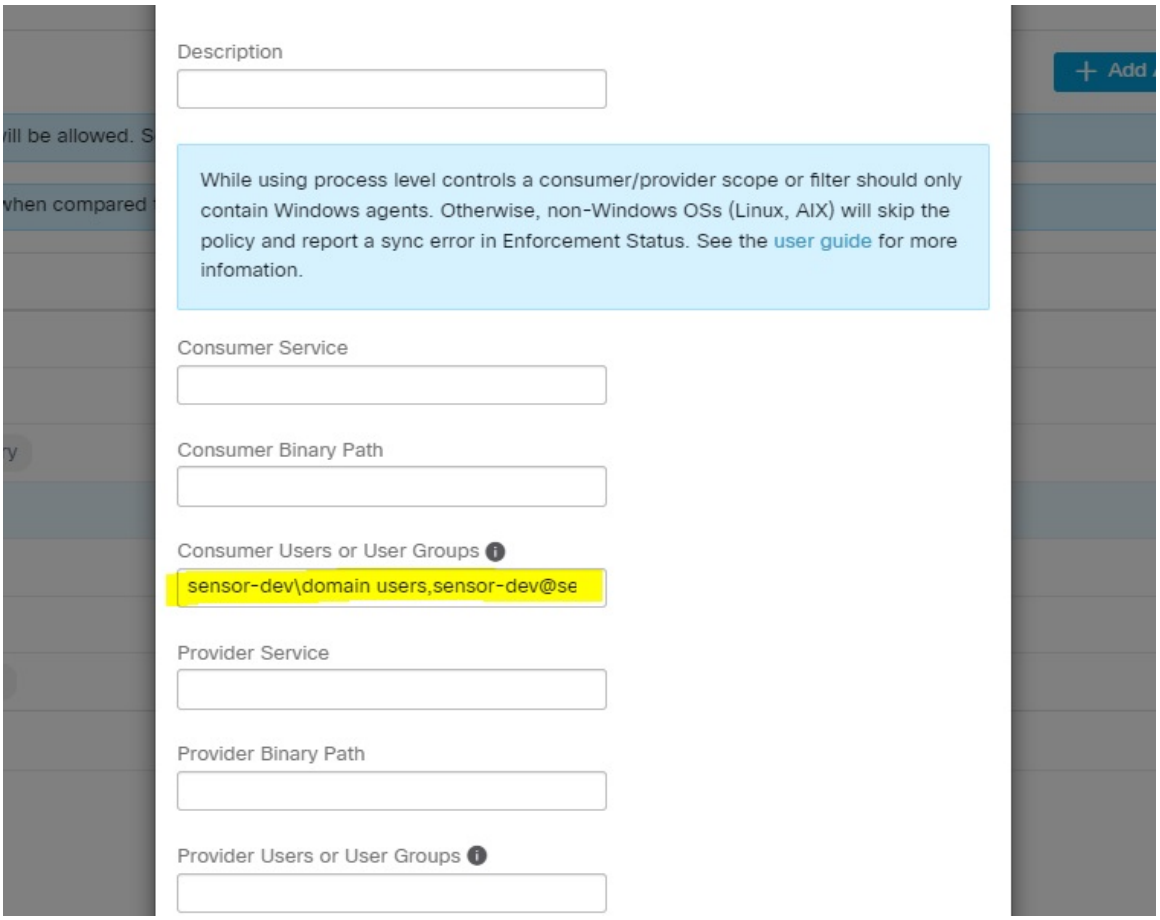
  ```
  sc qsidtype <service name>
  ```

ユーザーグループまたはユーザー名に基づくポリシー

次の情報を使用して、ユーザーグループ名の有無にかかわらず、ユーザー名に基づく Windows OS ワークロードのポリシーを確認およびトラブルシューティングします。

このトピックのセクションでは、ワークロードにポリシーを表示する方法について説明します。

このトピックの例は、次の情報を使用して設定されたポリシーに基づいています。



ユーザー名に基づくサンプルポリシー

```
dst_ports {
          start_port: 30000
          end_port: 30000
          provider_filters {
               user_name: “sensor-dev\sensor-dev”
           }
}}
ip_protocol: TCP
address_family: IPv4
inspection_point: EGRESS
```

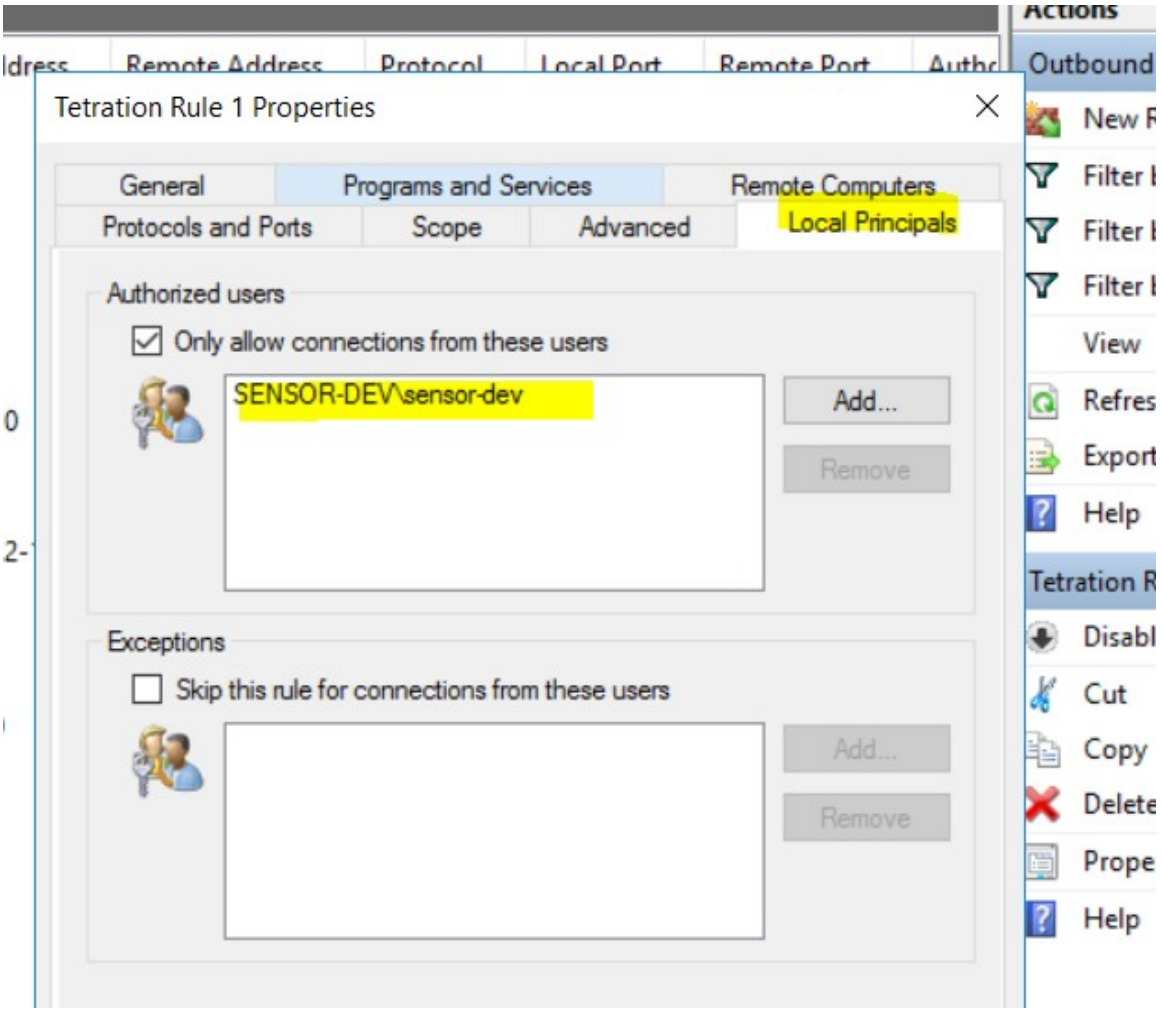
ユーザーグループとユーザー名に基づくサンプルポリシー

```
dst_ports {
start_port: 30000
end_port: 30000
provider_filters {
user_name: “sensor-dev\domain users,sensor-dev\sensor-dev”
}
}}
ip_protocol: TCP
address_family: IPv4
inspection_point: EGRESS
```

生成されたファイアウォールルール

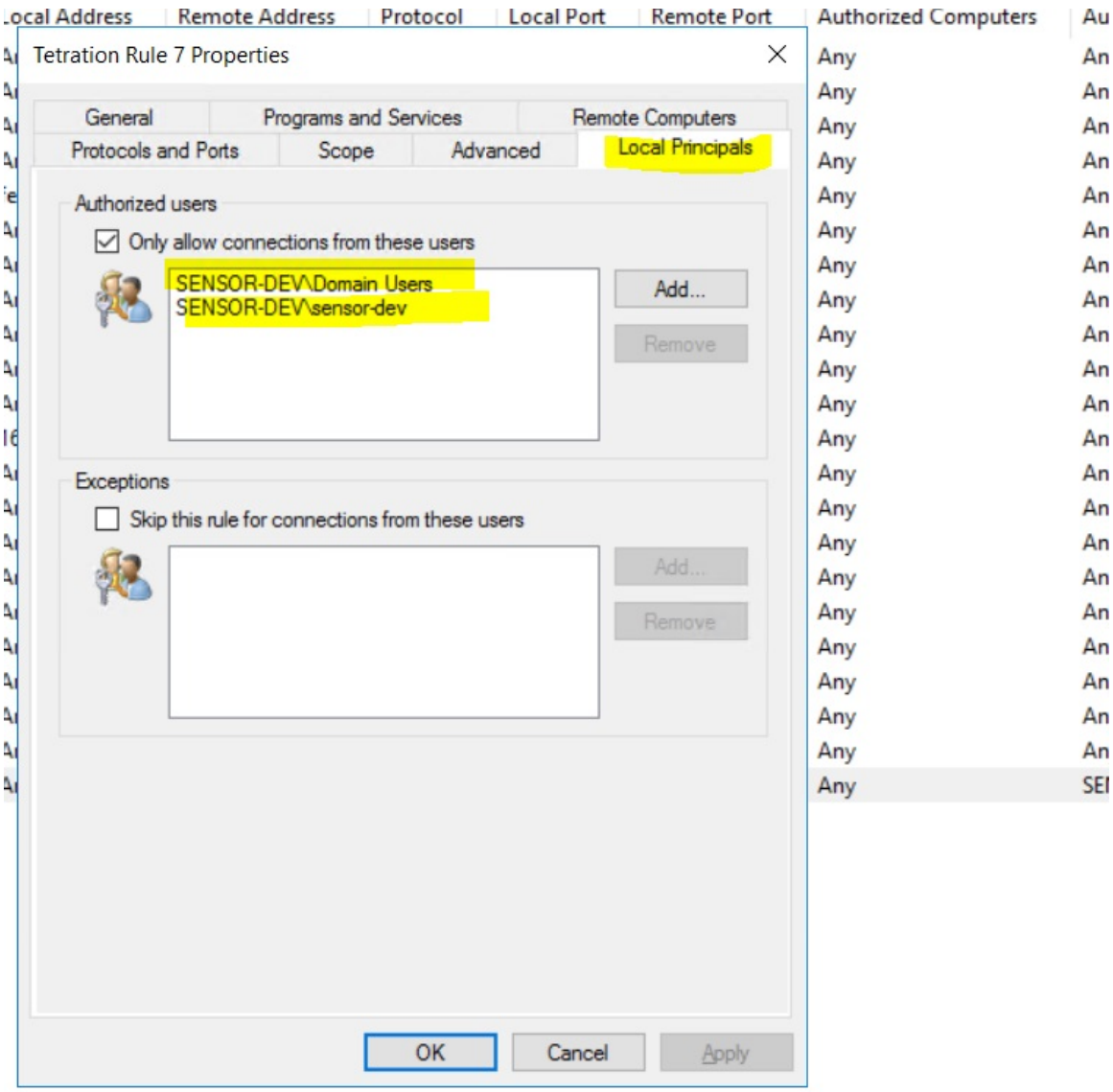
ユーザー名に基づくファイアウォールルール

例：ユーザー名 sensor-dev\\sensor-dev に基づくファイアウォールルール



ユーザーグループとユーザー名に基づくファイアウォールルール

例：ユーザー名 sensor-dev\\sensor-dev およびユーザーグループ domain users\\sensor-dev に基づくファイアウォールルール



netsh を使用して生成されたフィルタ

高度なポリシーにフィルタが追加されていることをネイティブの Windows ツールで確認するには、次の手順を実行します。

- 「管理者」権限を使用して「cmd.exe」を実行します
- 「netsh wfp show filters」を実行します
- 出力ファイル filters.xml が、現在のディレクトリに生成されます。
- 出力ファイル（filters.xml）でユーザー名の FWPM_CONDITION_ALE_USER_ID を確認します。

```
    <item>
                <fieldKey>FWPM_CONDITION_ALE_USER_ID</fieldKey>
                <matchType>FWP_MATCH_EQUAL</matchType>
                <conditionValue>
                       <type>FWP_SECURITY_DESCRIPTOR_TYPE</type>
                      <sd>O:LSD:(A;;CC;;;S-1-5-21-4172447896-825920244-2358685150)</sd>

                </conditionValue>
    </item>
```

tetenf.exe -l -f を使用して生成された WFP フィルタ

ユーザー名に基づくフィルタ

例：ユーザー名 SENSOR-DEV\sensor-dev に基づく WFP ルール

```
Filter Name:                    Secure Workload Rule 1
-----------------------------------------------------
EffectiveWeight:                18446744073709551590
LayerKey:                       FWPM_LAYER_ALE_AUTH_CONNECT_V4
Action:                         Permit
RemoteIP:                       10.195.210.15-10.195.210.15
Remote Port:                    30000
Protocol:                       6
User or Service:                SENSOR-DEV\sensor-dev
```

ユーザーグループとユーザー名に基づくフィルタ

例：ユーザー名 SENSOR-DEV\\sensor-dev およびユーザーグループ名 SENSOR-DEV\\Domain Users に基づく WFP ルール

```
Filter Name:          Secure Workload Rule 1
-----------------------------------------------------
EffectiveWeight:              18446744073709551590
LayerKey:                     FWPM_LAYER_ALE_AUTH_CONNECT_V4
Action:                       Permit
RemoteIP:                     10.195.210.15-10.195.210.15
Remote Port:                  30000
Protocol:                     6
User or Service:              SENSOR-DEV\Domain Users, SENSOR-DEV\sensor-dev
```

ネットワークポリシールールにサービス名とユーザー名を設定することはできません。

ユーザー名またはユーザーグループが不正な場合

- ユーザー名またはユーザーグループが無効な場合、ネットワークポリシーはWindowsエージェントによって拒否されます。

Windows OS ベースの推奨ポリシー設定

可能な場合は、常にポリシーでポートとプロトコルを指定します。任意のポート、任意のプロトコルを許可することはお勧めしません。

たとえば、ポートとプロトコルの制限を含めて生成されたポリシーは次のようになります。

```
dst_ports {
          start_port: 22
          end_port: 22
          consumer_filters {
```

```
            application_name: "c:\test\putty.exe"
        }
}}
ip_protocol: TCP
```

対照的に、任意のプロトコルと任意のポートを使用して iperf.exe によって開始されたネットワーク接続を許可する場合、生成されるポリシーは次のようになります。

```
match_set {
dst_ports { end_port: 65535 consumer_filters {
application_name: "c:\test\iperf.exe"
}
} address_family: IPv4 inspection_point: EGRESS match_comment:
"PolicyId=61008290755f027a92291b9d:61005f90497d4f47cedacb86:"
}
```

上記のフィルタの場合、Secure Workload はプロバイダーのネットワークトラフィックを許可するポリシールールを次のように作成します。

```
match_set {
dst_ports { end_port: 65535
} address_family: IPv4 inspection_point: INGRESS match_comment:
"PolicyId=61008290755f027a92291b9d:61005f90497d4f47cedacb86:"
}
```

このネットワークルールは、プロバイダーのすべてのポートを開きます。任意のプロトコルを使用して OS ベースのフィルタを作成しないことをお勧めします。

既知の制限事項

- Windows 2008 R2 は、Windows OS ベースのフィルタリングポリシーをサポートしていません。
- ネットワークポリシーは単一のユーザー名で設定できますが、MS Firewall UI は複数のユーザーをサポートします。

警告

- Windows OS ベースのポリシーを使用している場合、コンシューマやプロバイダーの範囲またはフィルタには Windows エージェントのみを含める必要があります。そうしないと、Windows 以外の OS（Linux、AIX）ではポリシーがスキップされ、適用ステータスで同期エラーが報告されます。
- フィルタリング基準が緩い Windows OS フィルタを作成しないでください。不要なネットワークポートが開く可能性があります。
- ネットワークフローのプロセスコンテキスト、ユーザーコンテキスト、またはサービスコンテキストに関する知識が乏しいか、知識がまったくないために、ポリシーに Windows OS ベースのフィルタがある場合、ポリシー分析に矛盾が生じます。

AIX プラットフォームでのエージェントの適用

AIX プラットフォームでは、エージェントは IPFilter ユーティリティを使用してネットワークポリシーを適用します。エンドホストでエージェントを有効にすると、デフォルトで IPv4 フィルタテーブル を制御およびプログラムします。IPv6 適用はサポートされていません。

IPFilter

AIX の IPFilter パッケージは、ファイアウォールサービスを提供するために使用されます。AIX では、カーネル拡張パックとして利用できます。また、カーネル拡張モジュール（/usr/lib/drivers/ipf）としてロードされます。このモジュールには、ipfilter ルールをプログラムするために使用される ipf、ippool、ipfstat、ipmon、ipfs、および ipnat ユーティリティが含まれており、各ルールでパケットの一致基準を指定します。詳細については、AIX の IPFilter man ページを参照してください。

適用が有効な場合、エージェントは IPFilter を使用して、IPv4 パケットを許可またはドロップするルールを含む IPv4 フィルタテーブルをプログラムします。エージェントは、それらのルールをグループ化して、コントローラからポリシーを分類および管理します。ルールには、ポリシーから派生した Secure Workload ルールと、エージェントによって生成されたルールが含まれます。

エージェントは、プラットフォームに依存しないルールを受け取ると、それらのルールを解析して ipfilter/ippool ルールに変換し、フィルタテーブルに挿入します。ファイアウォールのプログラミング後、適用エージェントはファイアウォールを監視し、ルールやポリシーの逸脱がないか確認します。逸脱がある場合は、ファイアウォールを再プログラミングします。また、ファイアウォールでプログラムされたポリシーを追跡し、ポリシーのステータスを定期的にコントローラに報告します。

プラットフォームに依存しないネットワーク ポリシー メッセージの一般的なポリシーの構成は次のとおりです。

```
source set id: “test-set-1”
destination set id: “test-set-2”
source ports: 20-30
destination ports: 40-50
ip protocol: UDP
action: ALLOW
. . .
set_id: “test-set-1”
        ip_addr: 1.2.0.0
        prefix_length: 16
        address_family: IPv4
set_id: “test-set-2”
        ip_addr: 5.6.0.0
        prefix_length: 16
        address_family: IPv4
```

エージェントは、他の情報とともに、このポリシーを処理し、プラットフォーム固有の ippool および ipfilter ルールに変換します。

```
table role = ipf type = tree number = 51400
{ 1.2.0.0/16; };
table role = ipf type = tree number = 75966
{ 5.6.0.0/16; };
```

```
pass in quick proto udp from pool/51400 port 20:30 to pool/75966 port 40:50 group TA_
‹→INPUT
```

警告

ホストファイアウォールのバックアップ

エージェント設定プロファイルで初めて適用が有効になると、AIXホストで実行されている対象のエージェントは、ホストのファイアウォールを制御する前に、ippoolとipfilterの現在のコンテンツを */opt/cisco/tetration/backup* に保存します。適用設定の連続した無効化/有効化の切り替えでは、新しいバックアップは生成されません。ディレクトリは、エージェントのアンインストール時に削除されません。

既知の制限事項

IPv6 適用はサポートされていません。

Allow ポリシーにより、既存の UDP 接続のトラフィックが中断される可能性があります。

ソフトウェアエージェントの設定

ソフトウェアエージェント設定の要件と前提条件

必要な Secure Workload ユーザーロール：

- サイト管理者
- カスタマー サポート

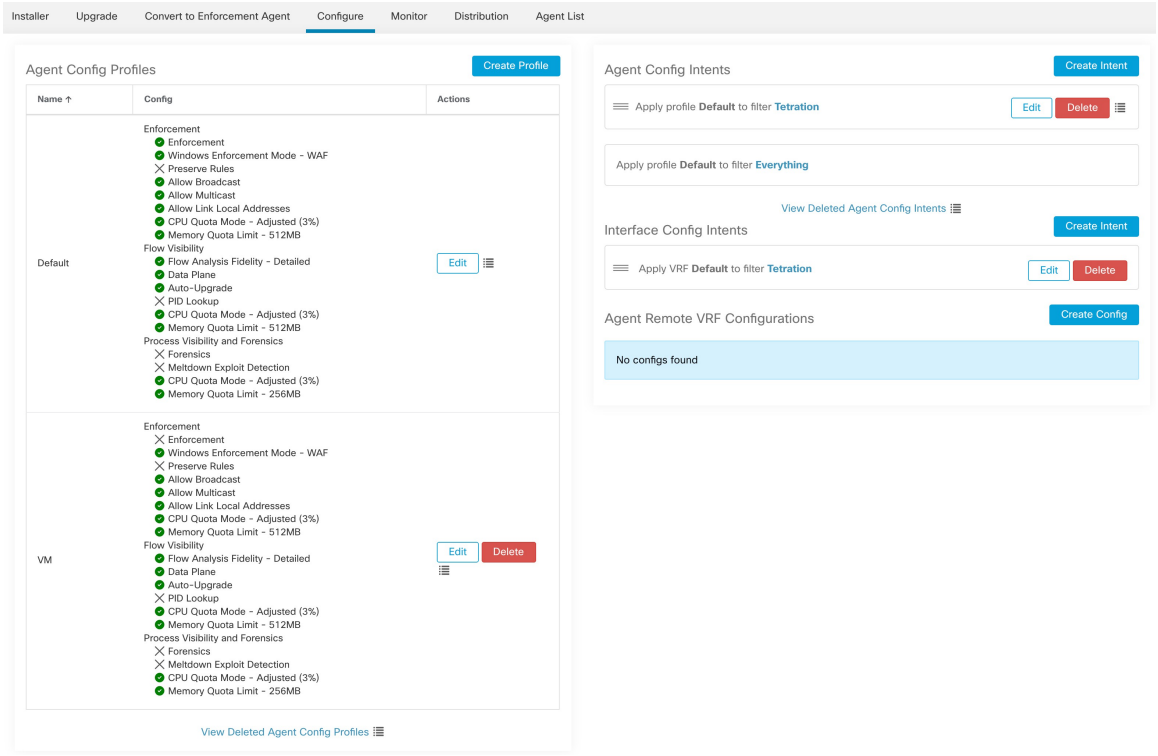
さらに、お客様または別の許可されたユーザーが、各ワークロードでエージェントサービスを実行するためのホストに対する権限を持っていることを確認してください。「ソフトウェアエージェント サービスの管理」を参照してください。

サポートされているプラットフォーム、要件、およびエージェントのインストール手順については、「ソフトウェアエージェントの展開」を参照してください。

ソフトウェアエージェントの構成

ソフトウェアエージェントは、**エージェント構成プロファイル**を**インベントリフィルタ**または**範囲**に関連付ける**エージェント構成インテント**を作成することによって構成されます。最初に一致したインテントが各エージェントに適用されます。Secure Workload展開には、特定の構成プロファイルに関連付けられていないすべてのセンサーに適用される、デフォルトのエージェント構成が常に存在します。

図 *15* : ソフトウェアエージェント構成ページ



エージェント設定プロファイルの作成

ステップ 1 左側のナビゲーションバーで、**[管理（Manage）]** > **[エージェント（Agents）]** をクリックします。

ステップ 2 [Configure] タブをクリックします。

ステップ 3 [プロファイルの作成（Create Profile）] ボタンをクリックします。

ステップ 4 プロファイルの名前を入力し（必須）、プロファイルを使用できる範囲を選択します。

ステップ 5 次の表にリストされているフィールドに適切な値を入力します。

表 *6:* 適用設定

フィールド	説明
施行	[有効（Enable）]：エージェントでのポリシー適用を有効にします。 [無効（Disable）]（デフォルト）：エージェントでのポリシー適用を有効にしません。 （注） 有効化された適用を無効にしてから再度有効にすると、ファイアウォールの状態がクリアされ、キャッチオールのデフォルトアクションがALLOWに設定されます。
[ルールの保持（Preserve Rules）]	[有効（Enable）]：エージェントの既存のファイアウォールルールを保持します。 [無効（Disable）]（デフォルト）：Secure Workloadから適用ポリシールールを適用する前に、既存のファイアウォールルールをクリアします。 動作はプラットフォームによって異なります。各プラットフォームの詳細を確認するには、このドキュメントで「ルールの保持」を検索してください。
[ブロードキャストの許可（Allow Broadcast）]	[有効（Enable）]（デフォルト）：ファイアウォールにルールを追加して、ワークロードの入力および出力ブロードキャスト トラフィックを許可します。 [無効（Disable）]：ルールを追加しません。エージェントのデフォルトポリシーがdenyである場合、ブロードキャスト トラフィックはドロップされます。
[マルチキャストの許可（Allow Multi-casting）]	[有効（Enable）]（デフォルト）：ファイアウォールにルールを追加して、ワークロードの入力および出力マルチキャストトラフィックを許可します。 [無効（Disable）]：ルールを追加しません。エージェントのデフォルトポリシーがdenyである場合、マルチキャストトラフィックはドロップされます。
[リンクローカルの許可（Allow Link Local）]	[有効（Enable）]（デフォルト）：ファイアウォールにルールを追加して、ワークロードのリンクローカルアドレスのトラフィックを許可します。 [無効（Disable）]：ルールを追加しません。エージェントのデフォルトポリシーがdenyである場合、マルチキャストトラフィックはドロップされます。

フィールド	説明
適用プロセスの CPU クォータモード	[調整済み（Adjusted）]（デフォルト）：CPU 制限は、システム上の CPU の数に応じて調整されます。たとえば、CPU 制限が 3% に設定されていて、システムに 10 個の CPU がある場合、このモードを選択すると、エージェントは合計 30%（上位で測定）の使用が許可されます。 [上位（Top）]：CPU 制限値は、平均上位ビューと一致します。たとえば、CPU 制限が 3% に設定されていて、システムに 10 個の CPU があるとします。この場合でも CPU 使用率は 3% のままとなります。これはかなり制限の厳しいモードですので、必要な場合にのみ使用してください。 [無効（Disable）]：CPU 制限機能が無効になります。エージェントは、OS で許可されている CPU リソースを使用します。 詳細については、agent_cpu_sla.pdf を参照してください。
[CPUクォータ制限（%）（CPU Quota Limit (%)）]	エージェントが使用できるシステム処理能力の実際の制限をパーセントで指定します。
[メモリクォータ制限（MB）（Memory Quota Limit (MB)）]	プロセスが使用できるメモリ制限を MB 単位で指定します。プロセスがこの制限に達すると、再起動します。
[Windows適用モード（Windows Enforcement Mode）]	Windows ワークロードでは、エージェントは以下を使用してネットワークポリシーを適用します。 • [WFP]：Windows Filtering Platform（Windows Filter Engine で WFP フィルタを直接プログラミングします）。 • [WAF]（デフォルト）：Windows Advanced Firewall。 このガイドの「WFP モードの Windows プラットフォームでの Secure Workload 適用」および「WAF モードの Windows プラットフォームでの Secure Workload 適用」の情報も参照してください。

表 *7:* フローの可視性設定

フィールド	説明
[データプレーン（Data Plane）]	[有効（Enable）]（*）：エージェントがクラスタにレポートを送信できるようにします。 [無効（Disable）]：エージェントのレポートを無効にします。
自動アップグレード	[有効（Enable）]（*）：新しいパッケージが利用可能になったときに、エージェントを自動的にアップグレードします。 [無効（Disable）]：エージェントを自動的にアップグレードしません。
[PIDルックアップ（PID Lookup）]	[有効（Enable）]：エージェントでのPIDルックアップを有効にします。有効にすると、エージェントは、ネットワークフローをワークロードで実行中のプロセスに関連付けるようベストエフォートで試みます。この操作はコストがかかる可能性があるため、エージェントは各エクスポートサイクルで実行される操作の数を抑制して、CPUオーバーヘッドを制御します。設定が有効になっていても、一部のフローがどのプロセスにも関連付けられていない場合があります。 [無効（Disable）]（*）：エージェントでPIDルックアップを有効にしません。

フィールド	説明
[CPUクォータモード（CPU Quota Mode）]	[調整済み（Adjusted）]（*）：CPU 制限は、システム上のCPUの数に応じて調整されます。たとえば、CPU 制限が 3% に設定されていて、システムに 10 個の CPU がある場合、このモードを選択すると、エージェントは合計30%（上位で測定）の使用が許可されます。 [上位（Top）]：CPU制限値は、平均上位ビューと一致します。たとえば、CPU 制限が 3% に設定されていて、システムに 10 個の CPU があるとします。この場合でもCPU使用率は3%のままとなります。これはかなり制限の厳しいモードですので、必要な場合にのみ使用してください。 [無効（Disable）]：CPU制限機能が無効になります。エージェントは、OS で許可されている CPU リソースを使用します。 詳細については、agent_cpu_sla.pdf を参照してください。
[CPUクォータ制限（%）（CPU Quota Limit (%)）]	エージェントが使用できるシステム処理能力の実際の制限をパーセントで指定します。
[メモリクォータ制限（MB）（Memory Quota Limit (MB)）]	プロセスが使用できるメモリ制限を MB 単位で指定します。プロセスがこの制限に達すると、再起動します。
[クリーンアップ期間（日）（Cleanup period (days)）]	[有効（Enable）]：エージェントの自動クリーンアップを有効にします。非アクティブなエージェントを削除するまでの日数を入力します。 [無効（Disable）]：エージェントの自動クリーンアップを有効にしません。
[流動解析の忠実度（Flow Analysis Fidelity）]	[カンバセーション（Conversations）]：すべてのセンサーでカンバセーションモードを有効にします。 [詳細（Detailed）]（*）：すべてのセンサーで詳細モードを有効にします。

図 *16 :* フローの可視性

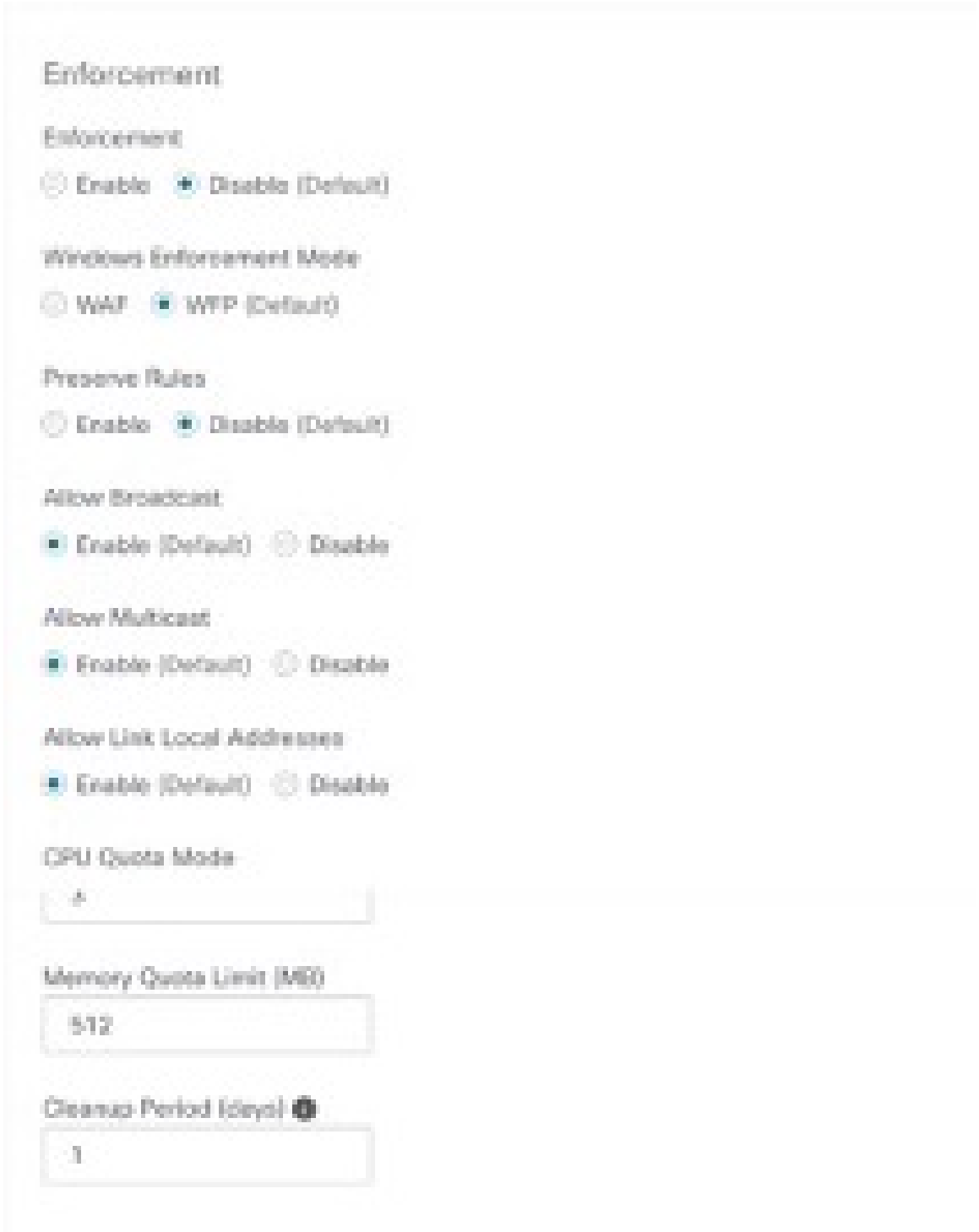

Enforcement
Enforcement
Enable
Disable (Default)
Windows Enforcement Mode
WAF
WFP (Default)
Preserve Rules
Enable
Disable (Default)
Allow Broadcast
Enable (Default)
Disable
Allow Multicast
Enable (Default)
Disable
Allow Link Local Addresses
Enable (Default)
Disable
CPU Quota Mode
Memory Quota Limit (MB)
512
Cleanup Period (days)
1

表 *8:* プロセスの可視性とフォレンジック設定

フィールド	説明
フォレンジック	[有効（Enable）]：エージェントでフォレンジックを有効にします。この機能は、次のCPU制限で指定された追加のCPUサイクルを消費する可能性があることに注意してください。たとえば、CPU 制限が 3% で、この機能が有効になっている場合、エージェントは合計で最大 6% を使用できると想定します。 [無効（Disable）]：エージェントでフォレンジックを無効にします。
[Meltdown エクスプロイト検出（Meltdown Exploit Detection）]	[有効（Enable）]：エージェントでの Meltdown エクスプロイト検出を有効にします。この機能を使用するには、フォレンジックを有効にする必要があります。詳細については、「互換性」の「サイドチャネル」を参照してください。 [無効（Disable）]：エージェントでの Meltdown エクスプロイト検出を無効にします。
[CPUクォータモード（CPU Quota Mode）]	[調整済み（Adjusted）]（*）：CPU 制限は、システム上のCPUの数に応じて調整されます。たとえば、CPU 制限が 3% に設定されていて、システムに 10 個の CPU がある場合、このモードを選択すると、エージェントは合計30%（**上位**で測定）の使用が許可されます。 [上位（Top）]：CPU制限値は、平均上位ビューと一致します。たとえば、CPU 制限が 3% に設定されていて、システムに 10 個の CPU があるとします。この場合でも CPU 使用率は 3% のままとなります。これはかなり制限の厳しいモードですので、必要な場合にのみ使用してください。 [無効（Disable）]：CPU制限機能が無効になります。エージェントは、OS で許可されている CPU リソースを使用します。 詳細については、agent_cpu_sla.pdf を参照してください。
[CPUクォータ制限（%）（CPU Quota Limit (%)）]	エージェントが使用できるシステム処理能力の実際の制限をパーセントで指定します。
[メモリクォータ制限（MB）（Memory Quota Limit (MB)）]	プロセスが使用できるメモリ制限を MB 単位で指定します。プロセスがこの制限に達すると、再起動します。

ステップ 6 [保存（Save）] をクリックします。

エージェント構成インテントの作成

ステップ 1 左側のナビゲーションバーで、**[管理（Manage）]** > **[エージェント（Agents）]** をクリックします。

ステップ 2 [Configure] タブをクリックします。

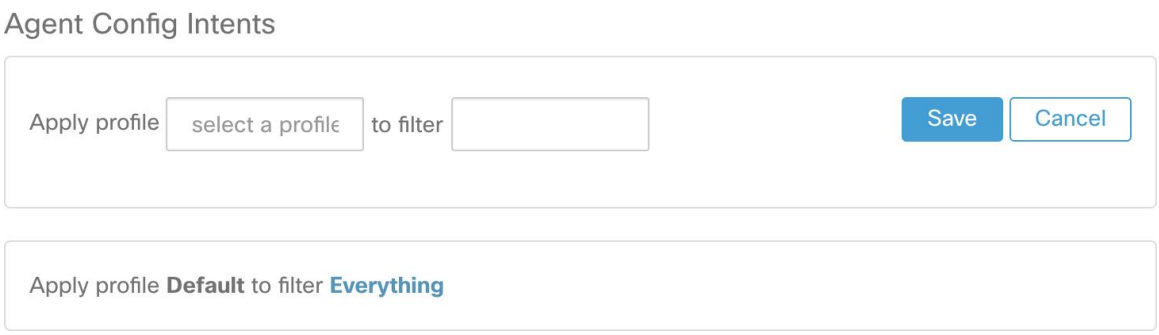
ステップ 3 [エージェント構成インテント（Agent Config Intent）] 見出しの横にある [インテントの作成（Create Intent）] ボタンをクリックします。

ステップ 4 次の表にリストされているフィールドに適切な値を入力します。

フィールド	説明
プロファイル	既存のプロファイルの名前を入力し、ドロップダウンメニューからそのプロファイル名を選択します（必須）。
フィルタ	既存のフィルタまたは範囲の名前を入力するか、ドロップダウンメニューから [新しいフィルタの作成（Create new filter）] を選択します（必須）。 フィルタの作成の詳細については、「フィルタ」を参照してください。

ステップ 5 **[保存（Save）]** をクリックします。

図 *17*: エージェント構成インテント



エージェントのリモート VRF 設定の作成

これは、Secure Workload ソフトウェアエージェントに VRF を割り当てる際に推奨される方法です。Secure Workload アプライアンスは、この設定を使用して、Secure Workload アプライアンスへの接続時にエージェントに表示されるソース IP アドレスとソースポートに基づいて、VRF をソフトウェアセンサーに割り当てます。

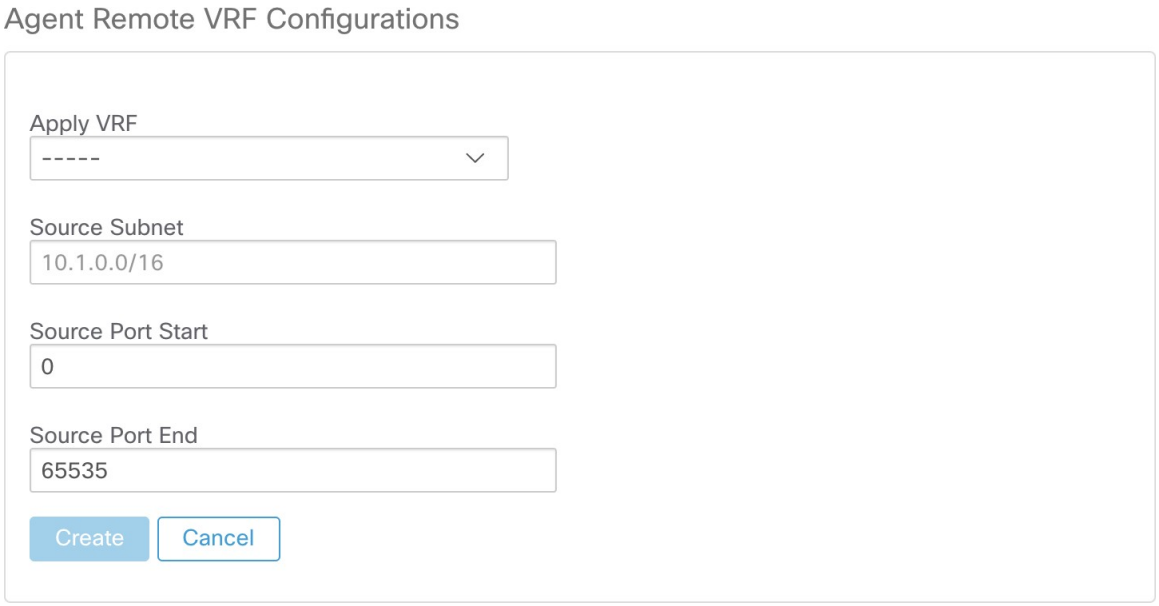
ステップ 1 左側のナビゲーションバーで、**[管理（Manage）]** > **[エージェント（Agents）]** をクリックします。

ステップ 2 [Configure] タブをクリックします。

ステップ 3 [エージェントのリモートVRF設定（Agent Remote VRF Configurations）] 見出しの横にある [設定の作成（Create Config）] ボタンをクリックします。

ステップ 4 フィールドに適切な値を入力し、[保存（Save）] をクリックします。

図 *18 :* リモート *VRF* 設定



インターフェイス構成インテントの作成

VRF をエージェントに割り当てる推奨される方法は、リモート VRF 構成設定を使用する方法です。まれなケースですが、エージェントホストに複数のインターフェイスがあり、異なる VRF を割り当てる必要がある場合、ユーザーはインターフェイス構成インテントを使用してインターフェイスに VRF を割り当てることができます。

ステップ 1 左側のナビゲーションバーで、**[管理（Manage）]** > **[エージェント（Agents）]** をクリックします。

ステップ 2 [Configure] タブをクリックします。

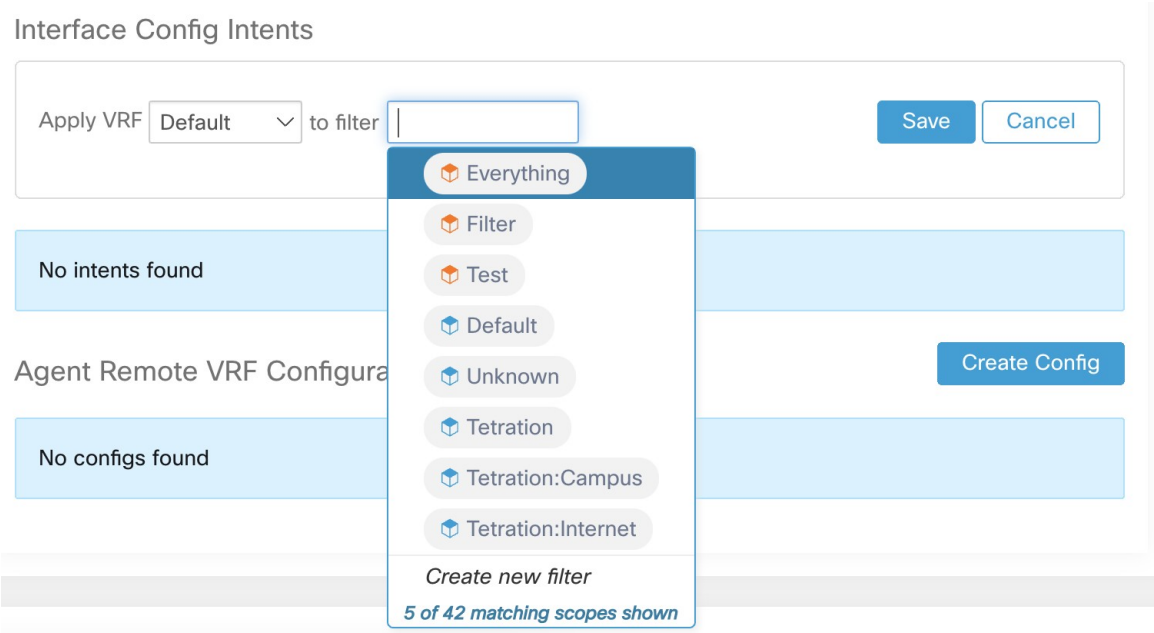
ステップ 3 [インターフェイス構成インテント（Interface Config Intent）] 見出しの横にある [インテントの作成（Create Intent）] ボタンをクリックします。

ステップ 4 次の表にリストされているフィールドに適切な値を入力します。

フィールド	説明
VRF	（必須）ドロップダウンメニューから VRF を選択します。
フィルタ	（必須）既存のフィルタまたは範囲の名前を入力するか、ドロップダウンメニューから [新しいフィルタの作成（Create new filter）] を選択します。 フィルタの作成の詳細については、「フィルタ」を参照してください。

ステップ 5 **[保存（Save）]** をクリックします。

図 *19*：インターフェイス構成インテント



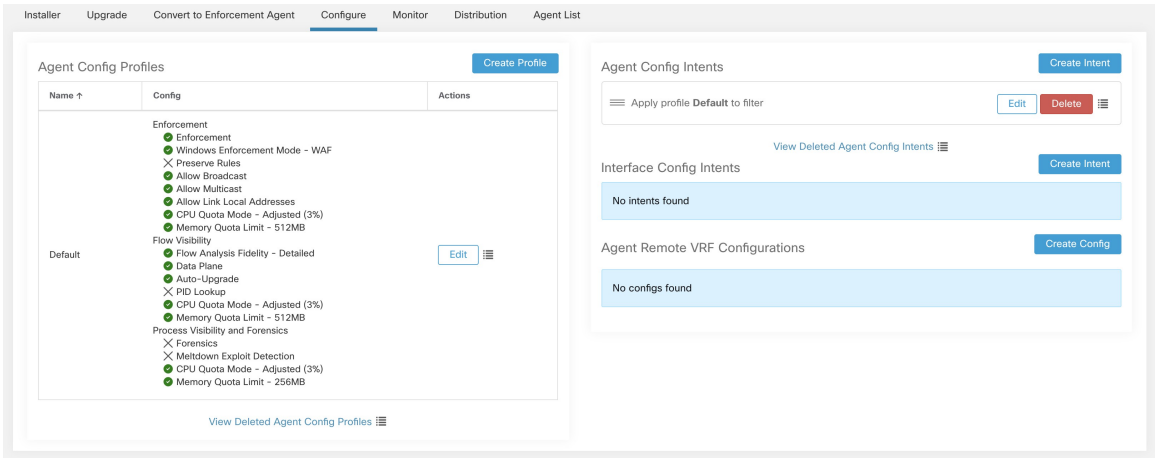
（注） Catch All インターフェイス構成インテントが適用されないという既知の問題があります。これは、より優先順位の高いインターフェイス構成インテントをユーザーが削除する場合にのみ当てはまります。そのような場合、エージェントはデフォルトの Catch All インテントにフォールバックしません。

ユーザーのロールとエージェント構成へのアクセス

1. ルート範囲の所有者は、「構成プロファイル」の作成と「構成インテント」の仕様にのみアクセスできます。

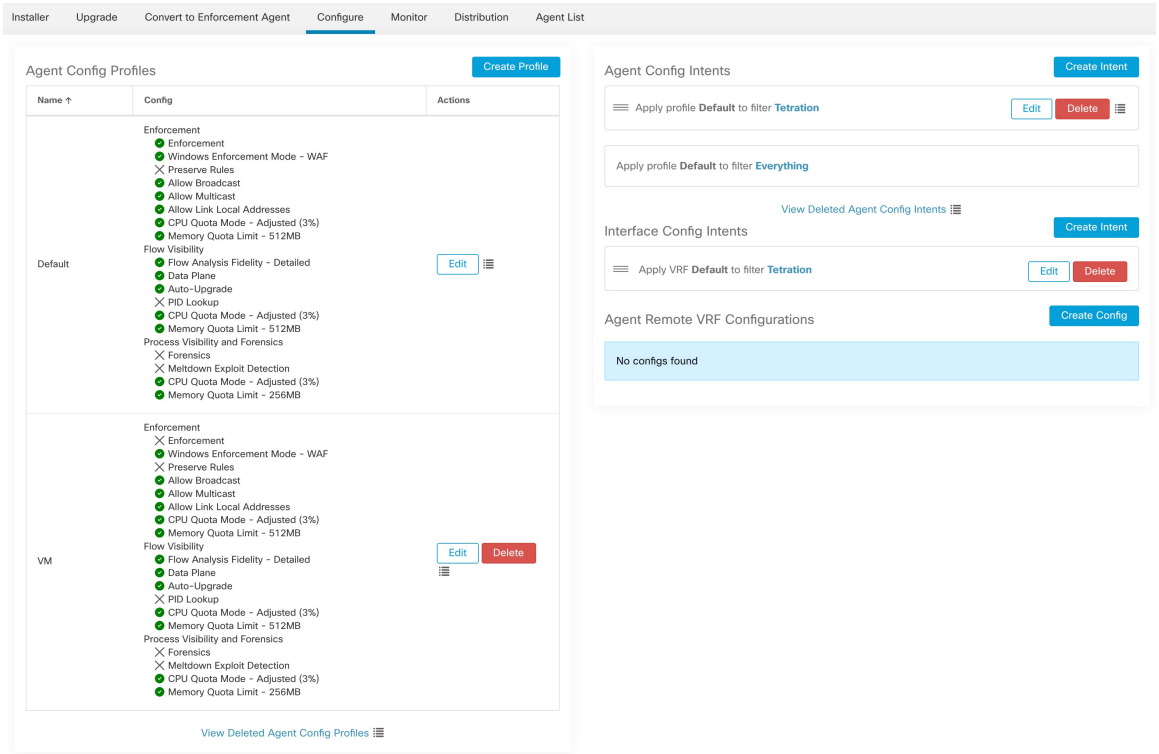
2. ルート範囲の所有者は、所有する範囲にのみ関連付けられた構成プロファイルを作成し、所有するフィルタや範囲に該当するエージェントにのみ構成プロファイルを適用できます。

図 *20* : 範囲所有者ユーザーの *[*ソフトウェアエージェント構成（*Software Agent Config***）*]* タブ**



3. サイト管理者ユーザーは、インターフェイス構成インテントとリモートVRF構成の指定を含む、[エージェント構成（Agent Config）]ページのすべてのコンポーネントにアクセスできます。

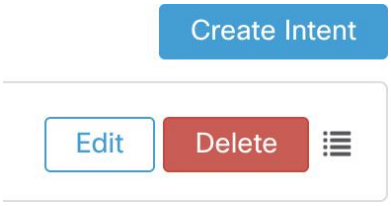
図 *21* : サイト管理者ユーザーの *[*ソフトウェアエージェント構成（*Software Agent Config***）*]* タブ**



ログの変更

ルート範囲の SCOPE_OWNER の資格を持つ**サイト管理者**およびユーザーは、以下に示すとおり、項目の横のアイコンをクリックすることで、各プロファイルおよびインテントの変更ログを表示できます。

図 *22 :* ログの変更



これらのユーザーは、対応する各テーブルの下にある [削除されたプロファイル/インテントを表示（View Deleted Profile/Intent）] リンクをクリックして、削除されたプロファイルとインテントのリストを表示することもできます。

変更ログの詳細については、../change_log を参照してください。ルート範囲の所有者は、その範囲に属するエンティティの変更ログエントリの表示に制限されます。

ソフトウェアエージェントのアップグレード

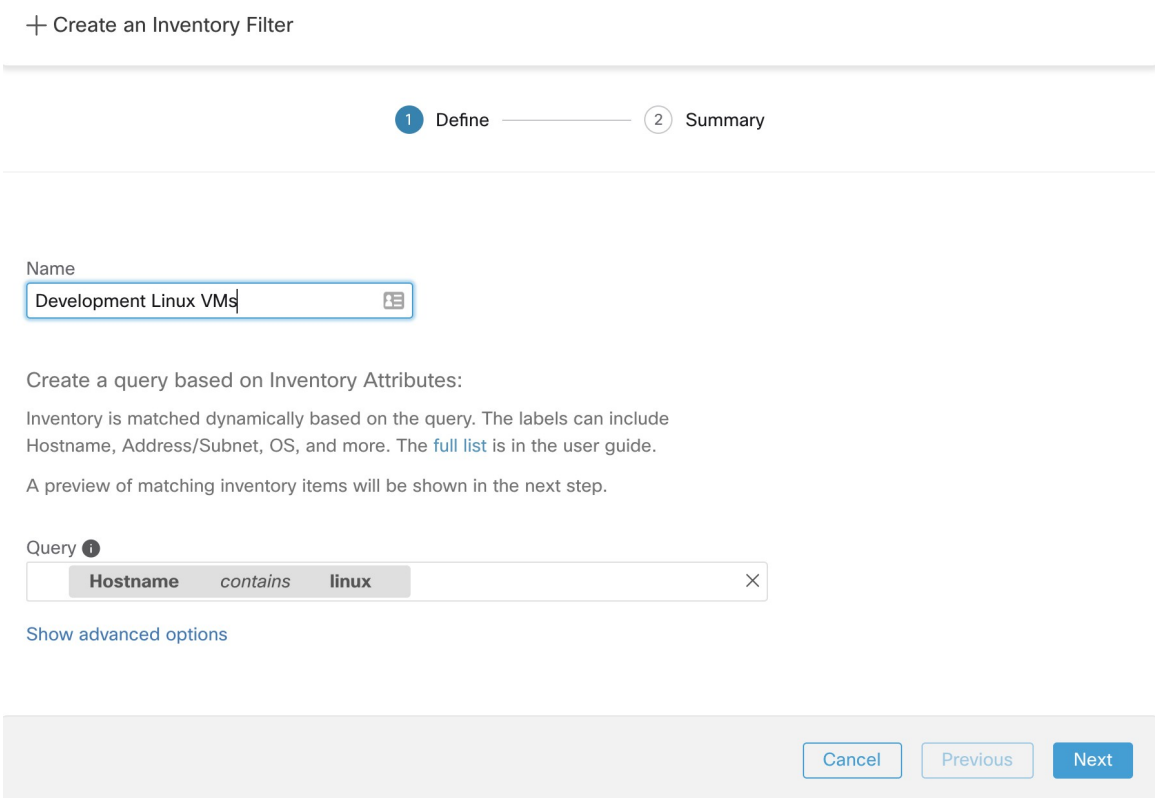
UI からのエージェントのアップグレード

エージェントは、「ソフトウェアエージェントの設定」で詳しく説明されているエージェント設定インテントワークフローを使用してアップグレードできます。エージェント設定プロファイルの設定時に、[有効（Enabled）] または [無効（Disabled）] にできる [自動アップグレード（Auto Upgrade）] オプションがあります。オプションが [有効（Enabled）] になっている場合、インベントリフィルタ条件に一致するエージェントは、利用可能な最新バージョンのソフトウェアに自動的にアップグレードされます。

次のセクションでは、ソフトウェアエージェント設定インテントワークフローを使用して、ソフトウェアエージェントのアップグレード動作を指定する方法について説明します。

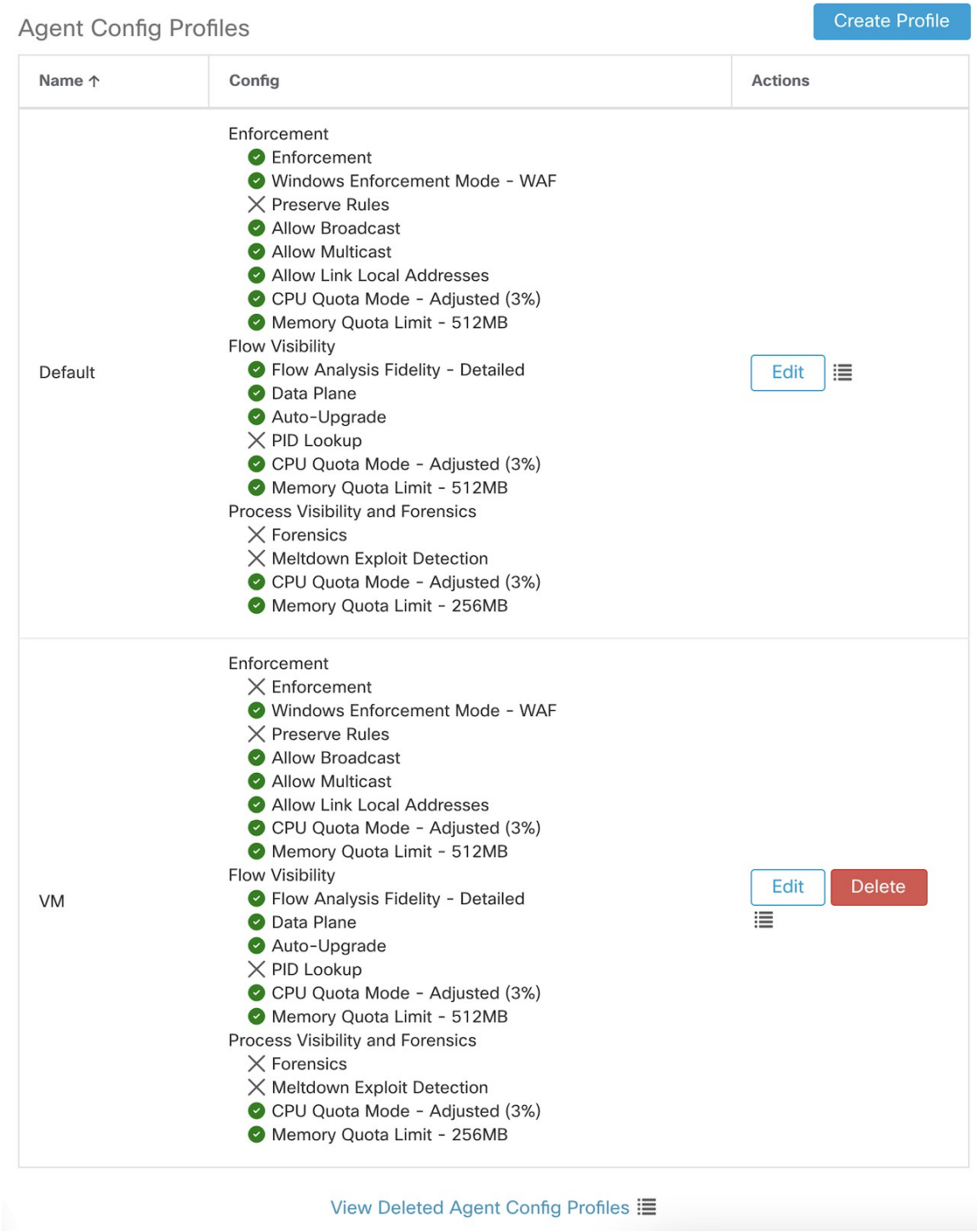
1. [インベントリフィルタ（Inventory Filters）] ページでインベントリフィルタを作成します。詳細については「フィルタ」をご覧ください。

図 *23* : インベントリフィルタ



2. 上記のインベントリフィルタによって選択されたエージェントにユーザーが適用するエージェント設定プロファイルを作成します。エージェント設定プロファイルには、選択したエージェントを自動アップグレードするかどうかを制御する [自動アップグレード（Auto Upgrade）] オプションがあることに注意してください。

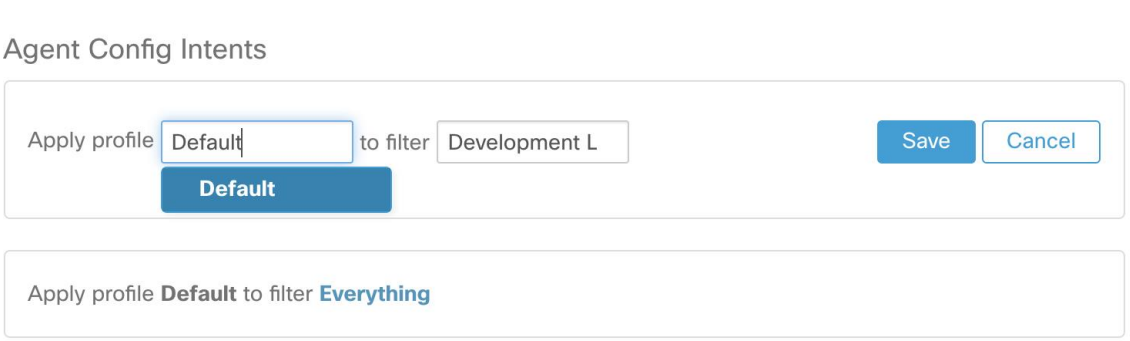
図 *24* : エージェント設定



3. 最後に、選択した設定プロファイルを、選択したエージェントのセットに適用するエージェント設定インテントを作成する必要があります（インベントリフィルタを使用）。自動アップグレードオプションが有効になっている場合、選択したすべてのエージェントが

自動アップグレードされます。通常、エージェントプロファイルがエージェントに適用された後、エージェントがアップグレードされるまでに最大 30 分かかることがあります。

図 *25*：エージェント設定インテント



（注） デフォルトのエージェントプロファイルの自動アップグレード設定は、ERSPAN に適用されます。

次のセクションでは、センサー設定インテントワークフローを使用せずに、エージェントを手動でアップグレードする方法について説明します。

ステップ 1 左側のナビゲーションバーで、**[管理（Manage）]** > **[エージェント（Agents）]** をクリックします。

ステップ 2 [アップグレード（Upgrade）] タブをクリックします。

ステップ 3 詳細可視性エージェントと適用エージェントのみが表示され、エージェントごとに、アップグレード可能な新しいバージョンのみがリストに表示されます。デフォルトでは、最新バージョンが選択されます。

ステップ 4 フィルタボックスに検索クエリを入力して、エージェントリストをフィルタリングします。たとえば、「`Platform = CentOS-7.6`」と入力します。

ステップ 5 このバージョンにアップグレードするエージェントを選択し、[アップグレード（Upgrade）] ボタンをクリックします。

（注） 通常の状況では、エージェントにアップグレードを処理させることが強く推奨されます。これがサポートされている唯一のアップグレード方法です。ユーザーが新しいバージョンを手動でダウンロードして実行中のエージェントに直接適用する仕方でアップグレードを制御する場合は、安全上の注意事項に従いつつ実行する必要があります。

Kubernetes/Openshift エージェントの動作のアップグレード

daemonset インストーラスクリプトを使用して Kubernetes/Openshift ノードにインストールされたエージェントは、セルフアップグレードが可能です。アップグレードプロセスは、自動アップグレードオプションによって、または Kubernetes/Openshift クラスタ内の任意のノードに対

してアップグレードを手動でトリガーすることによって制御されます。この環境でのアップグレードのメカニズムでは、daemonset 仕様で Docker イメージをアップグレードします。つまり、次の段落で説明するように、1 つのエージェントのアップグレードが、daemonset によってカバーされるすべてのエージェントに影響します。

Daemonset Pod 仕様が変更されると、Kubernetes/Openshift はグレースフルシャットダウンをトリガーし、新しい Docker イメージを取得して、Kubernetes/Openshift クラスタ内のすべてのノードで Secure Workload エージェントポッドを開始します。これにより、アップグレードを許可するポリシーがクラスタ内のノードのサブセットにのみ適用される場合でも、エージェントは他のノードでアップグレードされます。

すべてのノードで自動アップグレードが無効になっている場合は、新しいインストーラスクリプトをダウンロードしてインストールを再実行することにより、手動アップグレードが可能です。インストールスクリプトは、新しいインストールと既存のインストールのアップグレードの場合を自動検出し、インストール済みであることを検出すると、daemonset ポッドを手動でアップグレードするように機能します。

ソフトウェアエージェントの削除

優れた可視性/適用 Linux エージェントの削除

RPM ベースのインストール：

1. コマンド「**rpm -e tet-sensor**」を実行します。
2. [ソフトウェアエージェント（Software Agent）] ページの UI からエージェントを削除します。

Ubuntu.deb ベースのインストール：

Ubuntu エージェントの新規インストールでは、ネイティブの .deb 形式が使用されるようになりました。

1. コマンド「**dpkg –purge tet-sensor**」を実行します。
2. [ソフトウェアエージェント（Software Agent）] ページの UI からエージェントを削除します。

（注）
- エージェントの操作中に、一部のカーネルモジュールがカーネルによって自動的にロードされる可能性があります。たとえば、Linux で適用が有効になっている場合、Netfilter モジュールがロードされる可能性があります。エージェントには、カーネルによってロードされたモジュールのリストがありません。したがって、エージェントのアンインストール中に、カーネルモジュールをアンロードできない可能性があります。
- 適用エージェントがシステムファイアウォールにポリシーを適用した場合、エージェントをアンインストールすると、適用されたポリシーがクリアされ、システムファイアウォールが開きます。

優れた可視性/適用 Widnows エージェントの削除

Secure Workload エージェントをアンインストールするには、次の 2 つのオプションがあります。

ステップ 1 [コントロールパネル（Control Panel）] → [プログラム（Programs）] → [プログラムと機能（Programs And Features）] に移動し、**Cisco Secure Workload Agent**（Cisco Tetration Agent）をアンインストールします。

ステップ 2 または、「**C:\Program Files\Cisco Tetration**」内のショートカット **Uninstall.lnk** を実行します。

ステップ 3 [ソフトウェアエージェント（Software Agent）] ページの UI からエージェントを削除します。

ステップ 4 適用エージェントがシステムファイアウォールにポリシーを適用した場合、エージェントをアンインストールすると、適用されたポリシーがクリアされ、システムファイアウォールが開きます。

（注）
- エージェントのインストール中に Npcap がインストールされている場合、それもアンインストールされます。
- デフォルトのログファイルでは、アンインストール中に構成ファイルと証明書は削除されません。それらを削除したい場合は、同じフォルダでショートカット **UninstallAll.lnk** を実行します。

優れた可視性/適用 AIX エージェントの削除

ステップ 1 コマンド「**installp -u tet-sensor**」を実行します。

ステップ 2 [ソフトウェアエージェント（Software Agent）] ページの UI からエージェントを削除します。

（注）
- 優れた可視性エージェントは、System Resource Controller によって tet-sensor として制御されます。そのため、起動、停止、再起動、削除が可能です。このサービスは、inittab を tet-sen-engine として使用して永続化されます。
- 適用エージェントは、System Resource Controller によって tet-enforcer として制御されます。そのため、起動、停止、再起動、削除が可能です。このサービスは、inittab を tet-enf-engine として使用して永続化されます。
- エージェントの操作中に、一部のカーネルモジュールがカーネルによって自動的にロードされる可能性があります。たとえば、AIX で適用が有効になっている場合、ipfilter モジュールがロードされます。エージェントには、カーネルによってロードされたモジュールのリストがありません。したがって、エージェントのアンインストール中に、カーネルモジュールをアンロードできない可能性があります。
- 適用エージェントがシステムファイアウォールにポリシーを適用した場合、エージェントをアンインストールすると、適用されたポリシーがクリアされ、システムファイアウォールが開きます。

ユニバーサル Linux エージェントの削除

ステップ 1 アンインストールスクリプト **'/usr/local/tet-light/uninstall.sh'** を実行します。
ステップ 2 [ソフトウェアエージェント（Software Agent）] ページの UI からエージェントを削除します。

ユニバーサル Windows エージェントの削除

ステップ 1 アンインストールスクリプト「**C:\Program Files\Cisco Tetration\Lightweight Sensor\uninstall.cmd**」を実行します。
ステップ 2 [ソフトウェアエージェント（Software Agent）] ページの UI からエージェントを削除します。

適用 Kubernetes/Openshift エージェントの削除

ステップ 1 元のインストーラスクリプトを見つけるか、Secure Workload UI から新しいスクリプトをダウンロードします。
ステップ 2 アンインストールオプション **install.sh –uninstall** を実行します。インストール時と同じ考慮事項が適用されます。

- Linux x86_64 アーキテクチャでのみサポートされます。
- ~/.kube/config に含まれている管理者ユーザーログイン情報を使用するか、または –kubeconfig オプションを使用して kubectl 管理者ログイン情報ファイルを指定します。

ステップ 3 [ソフトウェアエージェント（Software Agent）]ページの UI からすべての Kubernetes ノードを削除します。

ワークロードエージェントにより収集されエクスポートされるデータ

このセクションでは、ソフトウェアエージェントの主要コンポーネント、バックエンドサービスへの登録方法、分析目的で収集されてクラスタにエクスポートされるデータについて説明します。

登録

エージェントがシステムに正常にインストールされたら、有効な一意の識別子を取得するために、エージェントをバックエンドサービスに登録する必要があります。次の情報が登録要求で送信されます。

- ホスト名
- BIOS-UUID
- プラットフォーム情報（CentOS-6.5 など）
- 自己生成クライアント証明書（openssl コマンドで生成）
- エージェントタイプ（可視性または適用）

エージェントがサーバーから有効な ID を取得できなかった場合、取得するまで再試行を続けます。エージェントの登録は非常に重要です。登録されていない場合、その後に続く他のサービス（コレクタなど）との通信がすべて拒否されます。

エージェントのアップグレード

エージェントは定期的（約 30 分） にバックエンドサービスにメッセージを送信して、現在のバージョンを報告します。バックエンドサービスは、エージェントの ID とその現在のバージョンを使用して、新しいソフトウェアパッケージがエージェントで使用できるかどうかを判断します。次の情報が送信されます。

- エージェントの ID （登録成功後に取得）
- 現在のエージェントのバージョン

構成サーバー

エージェントは、構成された構成サーバーに次の情報をエクスポートします。

- Hostname
- エージェントの ID（登録成功後に取得）
- インターフェイスのリスト。各インターフェイスに含まれる内容：
  1. インターフェイスの名前
  2. IP ファミリ（IPv4 または IPv6）
  3. IP アドレス
  4. ネットマスク
  5. MAC アドレス
  6. インターフェイスのインデックス

インターフェイスプロパティが変更されると（既存のインターフェイスのIPアドレスの変更、新しいインターフェイスの起動など）、このリストが更新され、構成サーバーに報告されます。

ネットワークフロー

ネットワークフロー情報は、システムを流れるすべてのパケットを要約したものです。フロー情報をキャプチャするには、詳細と会話の 2 つのモードがあります。デフォルトでは、詳細モードのキャプチャが使用されます。キャプチャされたフローは、1 秒ごとにコレクタにエクスポートされます（これは構成で変更できます）。エクスポートされた情報には、次の内容が含まれています。

- フロー識別子：ネットワークフローを一意に識別します。これには、IPプロトコル、送信元と宛先の IP、レイヤー 4 ポートなどの一般情報が含まれます。
- IP 情報：TTL、IP フラグ、パケット ID、IP オプション、フラグメンテーションフラグなど、IP ヘッダーに表示される情報が含まれます。
- TCP 情報：シーケンス番号、ACK 番号、TCP オプション、Rcvd ウィンドウサイズなど、TCP ヘッダーに表示される情報が含まれます。
- フロー情報：フローの統計情報（合計パケット数、合計バイト数、TCP フラグ統計情報、パケット長統計情報、ソケット統計情報など）、フローが観察されたインターフェイスインデックス、フローの開始時間と終了時間
- K8s 環境では、エージェントはポッドからのネットワークフローもキャプチャします。これらをホストで見られるフローと関連付けて、関連するフローとして報告します。

会話モードでは、エージェントは 15 秒～ 5 分ごとにアクティブフローを報告します。フローのエクスポート時間はプロトコルによって異なり、新しく完了したフローは、フローが観察されてから 25 秒以内に報告されます。

（注） n K8s 環境では、ポッド/ホストフローの関連付けは会話モードでは実行されません。

どちらのモードでも、エージェントは次のフローをエクスポートしないことに注意してください。

- ARP/RARP 会話
- エージェントからコレクタへのフロー

マシン情報

マシン情報には、ホストで実行されているすべてのプロセスが記述されています。さらに、プロセスに関連付けられたネットワーク情報と、プロセスの起動に使用されるコマンドが含まれています。マシン情報は毎分エクスポートされ、次の情報が含まれます。

- [プロセス ID（Process ID）]
- [ユーザーID（User ID）]：プロセスの所有者
- [親プロセス ID（Parent Process ID）]
- プロセスの起動に使用されるコマンド文字列
- [ソケット情報（Socket information）]：プロトコル（UDP または TCP など）、アドレスタイプ（IPv4 または IPv6）、送信元および宛先 IP、送信元および宛先ポート、TCP 状態、プロセスの開始時刻と終了時刻、バイナリを処理するパス
- [フォレンジック情報（Forensic information）]：詳細については、「互換性」のセクションを参照してください。

エージェント統計情報

エージェントは、システムの統計や独自の統計を含む、次のようなさまざまな統計を追跡します。

- エージェントの開始時間と稼働時間
- ユーザーモードとカーネルモードでのエージェントの実行時間
- ドロップされた受信データパケットの数
- 成功および失敗した SSL 接続の数
- 総フローパケット数とバイト数
- コレクタにエクスポートされたフローとパケットの合計

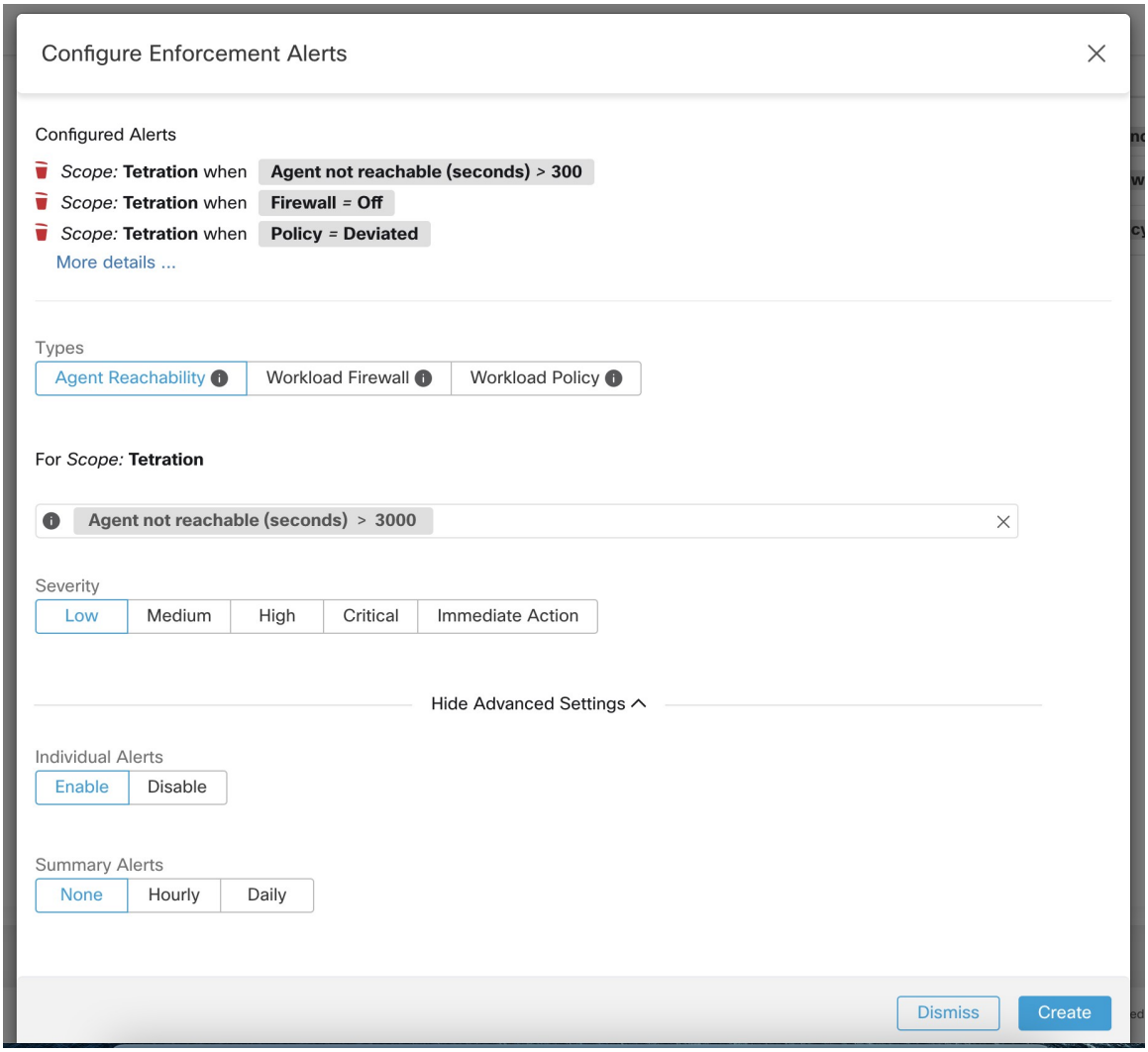
- エージェントのメモリおよび CPU の使用状況

適用アラート

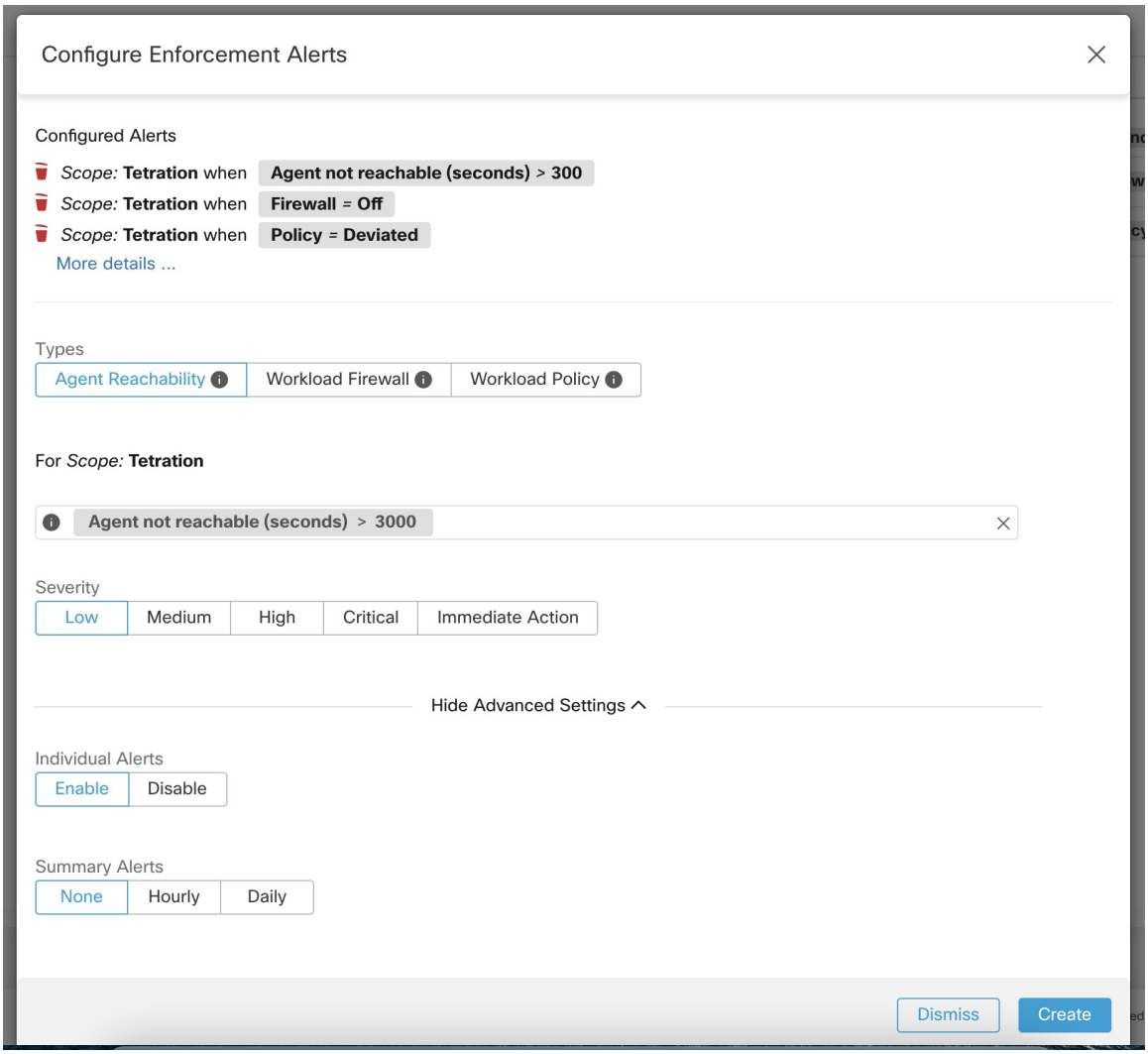
（注） 適用アラートはアラート設定モデルを使用して設定できます。

適用アラートはアラート設定モデルを使用して設定できます。モデルに関する一般的な情報については、「アラート設定モーダル」を参照してください。

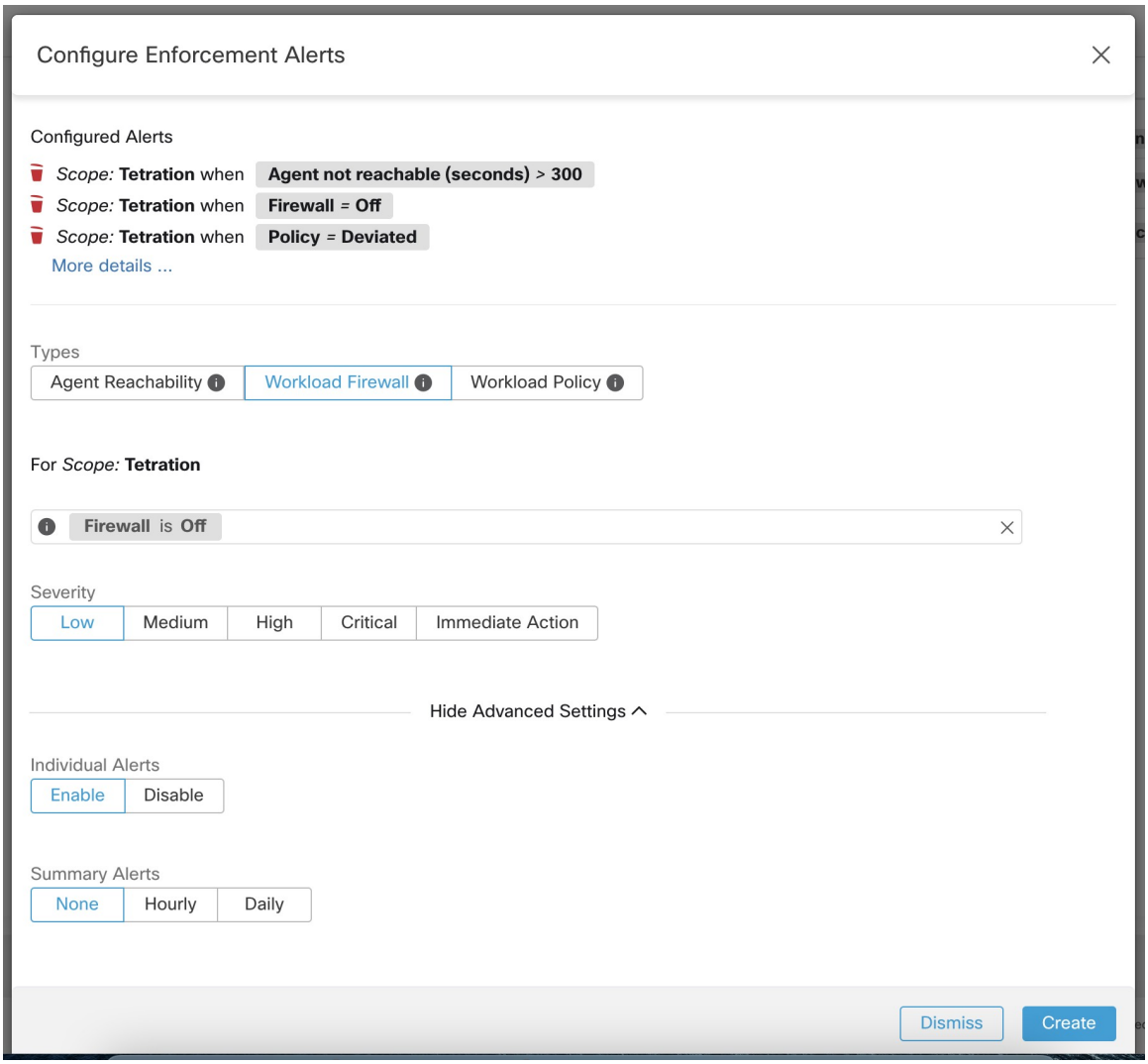
図 *26* : 適用アラートの設定



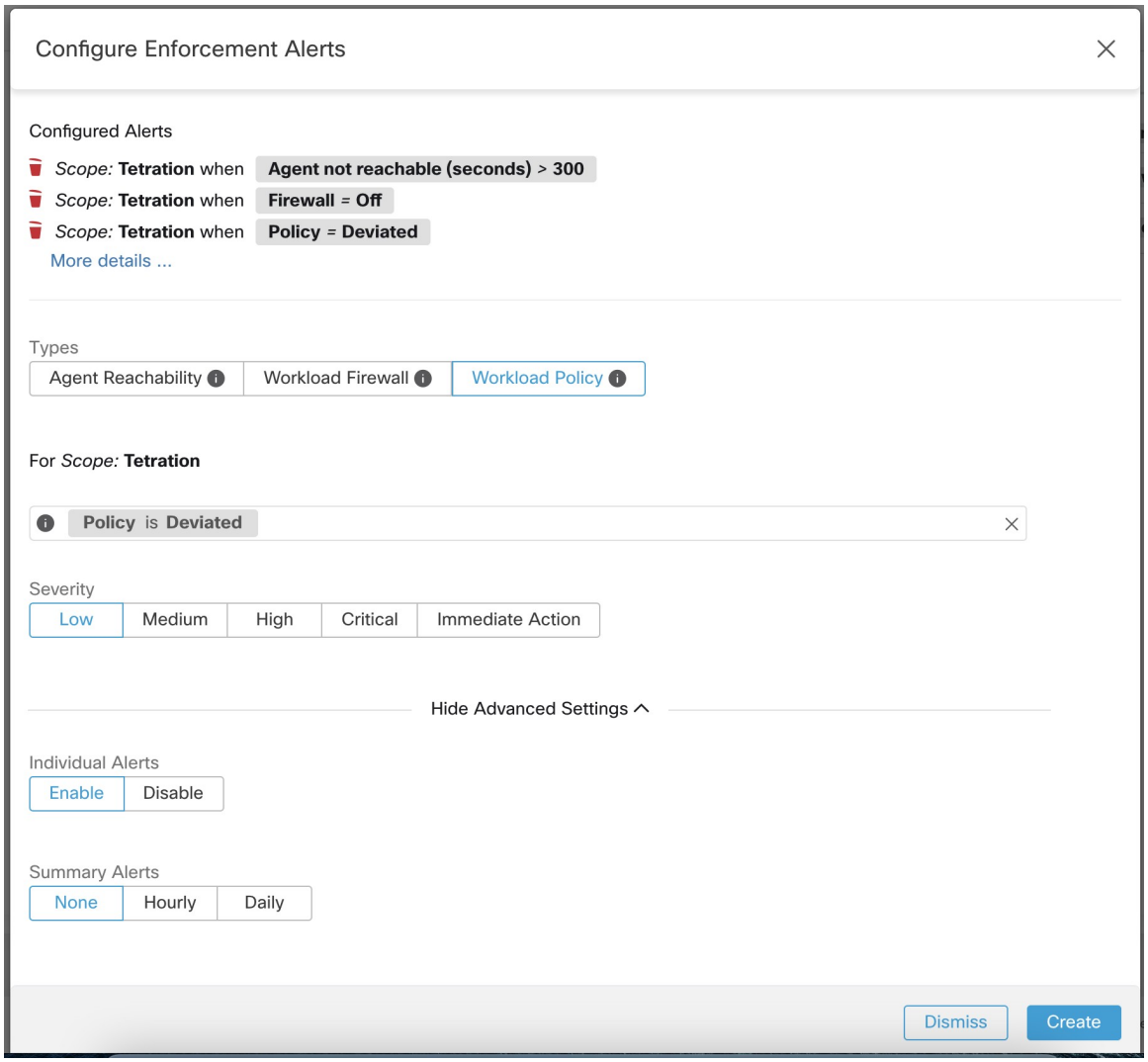
適用アラートの設定には、3 つの異なるタイプのアラートを設定する機能があり、ユーザーはアラートの重大度やその他のタイプごとの設定パラメータを設定できます。



ポリシーの適用が有効になっているエージェントに到達できない場合の適用アラートの設定。このアラートは、エージェントが設定された秒数を超えて Secure Workload クラスタと通信しなかった場合にトリガーされます。

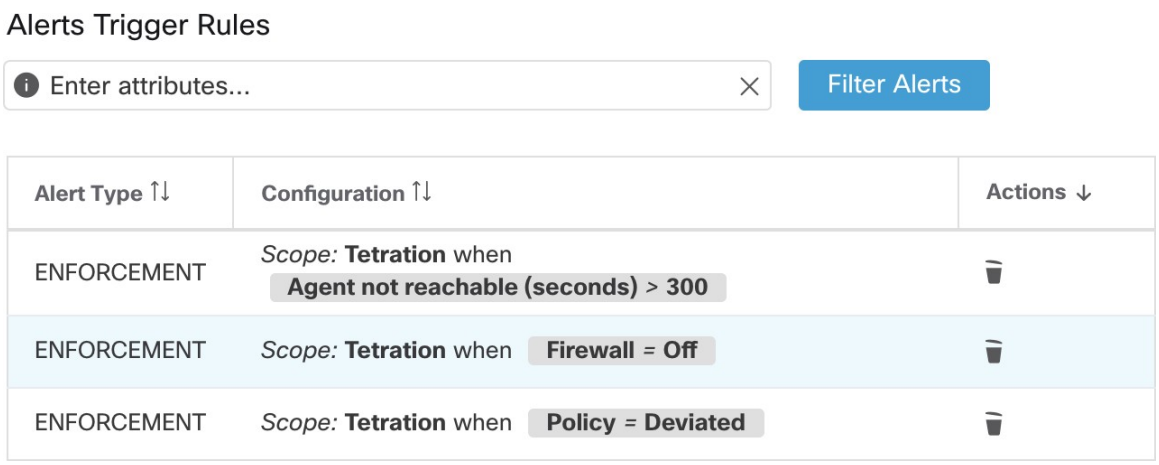


ワークロードファイアウォールがオフになったことを検知する適用アラートを設定します。このアラートは、ワークロードで適用が設定されているものの、ワークロードファイアウォールがオフであることが検知された場合にトリガーされます。この状態では、Secure Workload エージェントがトラフィックポリシーを適用できなくなるためです。



ワークロードポリシーからの逸脱が発生した場合の適用アラートの設定。このアラートは、ワークロード ファイアウォール ルールからの逸脱が発生した場合にトリガーされます。

図 *27* : アラート設定ページでの設定済み適用アラートの表示



適用 UI アラートの詳細

図 *28* : 適用アラートの詳細

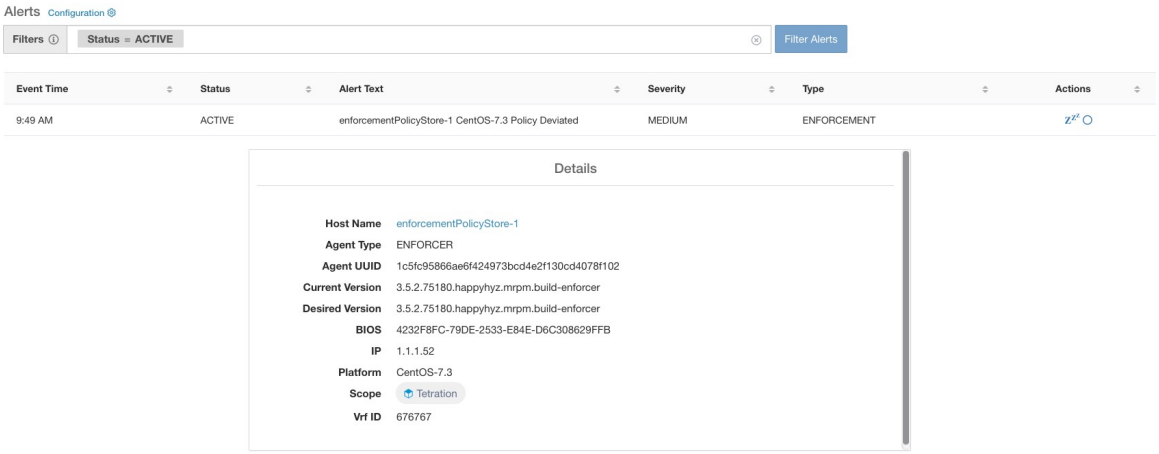
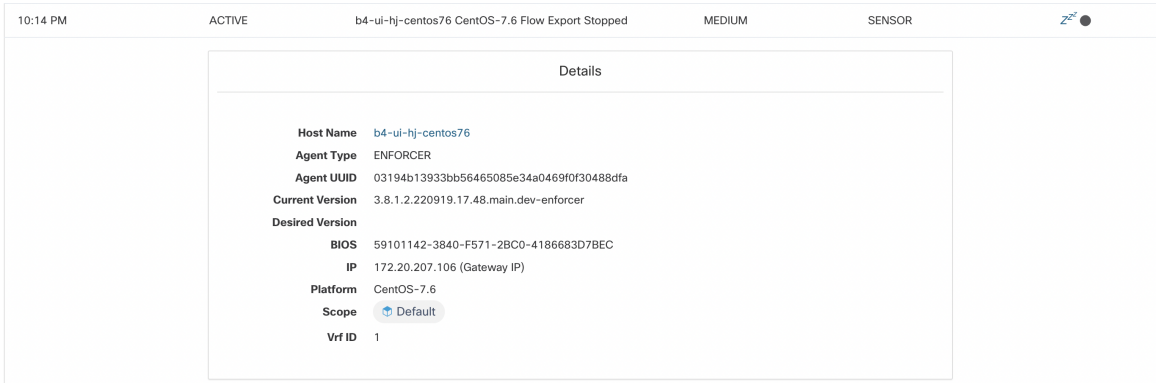


図 *29* : ホストでプロキシが有効になっている場合の適用アラートの詳細



適用アラートの詳細

一般的なアラート構造とフィールドに関する情報については、「共通アラート構造」を参照してください。alert_details フィールドは構造化されており、適用アラートの次のサブフィールドが含まれています。

フィールド	アラートタイプ	書式	説明
AgentType	*all*	string	インストールタイプに応じて「ENFORCER」または「SENSOR」
HostName	*all*	string	エージェントが展開されているホスト名
IP	*all*	string	ノードまたはゲートウェイの IP アドレス
Bios	*all*	string	ノードの BIOS UUID
プラットフォーム	*all*	string	ノードのプラットフォームまたは OS 情報
CurrentVersion	*all*	string	ノード上のエージェントのソフトウェアバージョン
DesiredVersion	*all*	string	エージェントに必要なソフトウェアバージョン
LastConfigFetchAt	*all*	integer	エージェントが最後に HTTPS リクエストを送信したときの UNIX タイムスタンプ

適用アラートの alert_details の例

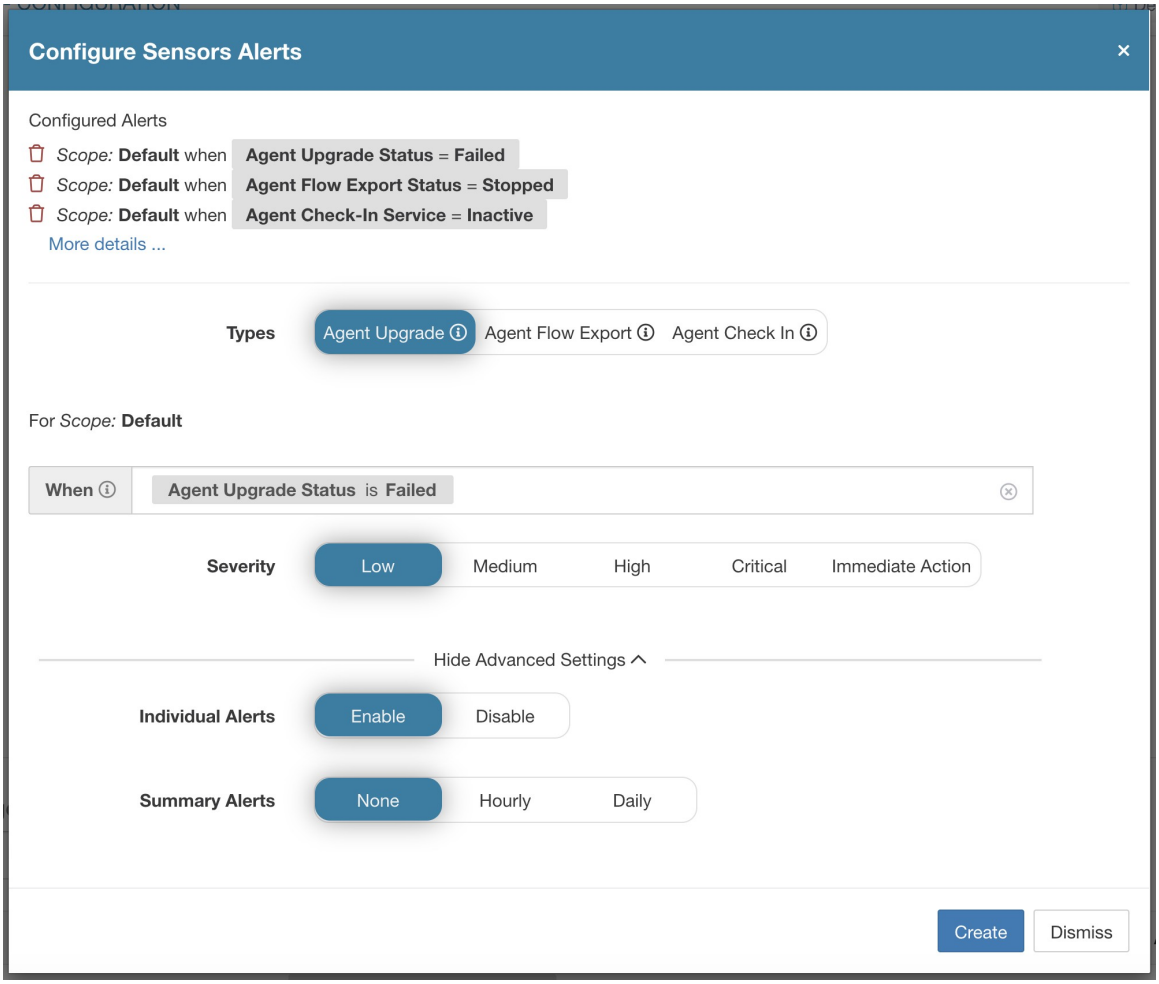
```
{
"AgentType":"ENFORCER",
"Bios":"72EF1142-03A2-03BC-C2F8-F600567BA320",
"CurrentVersion":"3.5.1.1.mrpm.build.win64-enforcer",
"DesiredVersion":"",
"HostName":"win2k12-production-db",
"IP":"172.26.231.193",
"Platform":"MSServer2012R2Standard"
}
```

センサーアラート

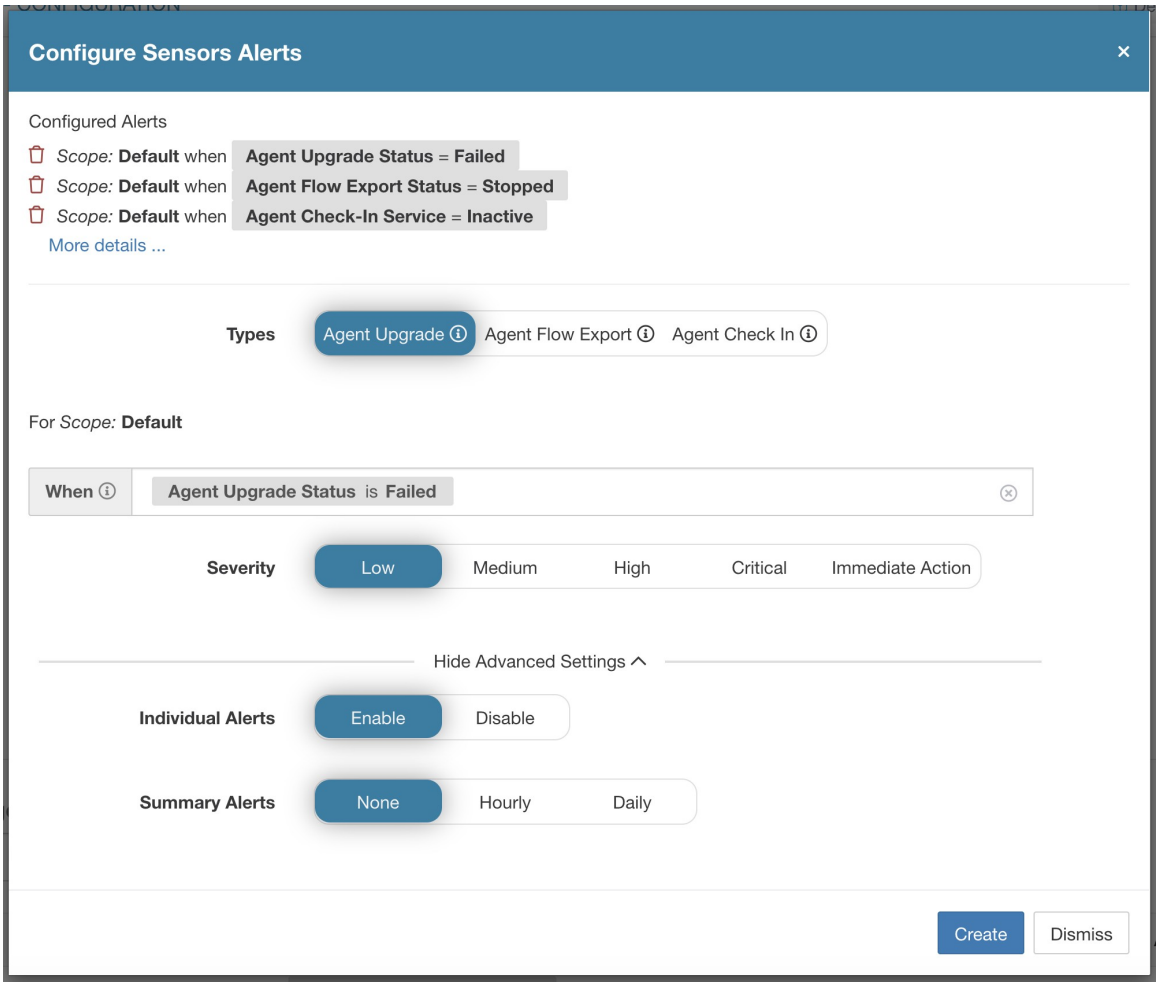
（注） リリース 3.5 以降、センサーアラートはアラート設定モデルを使用して設定できます。

センサーアラートはアラート設定モデルを使用して設定できます。モデルに関する一般的な情報については、「アラート設定モーダル」を参照してください。

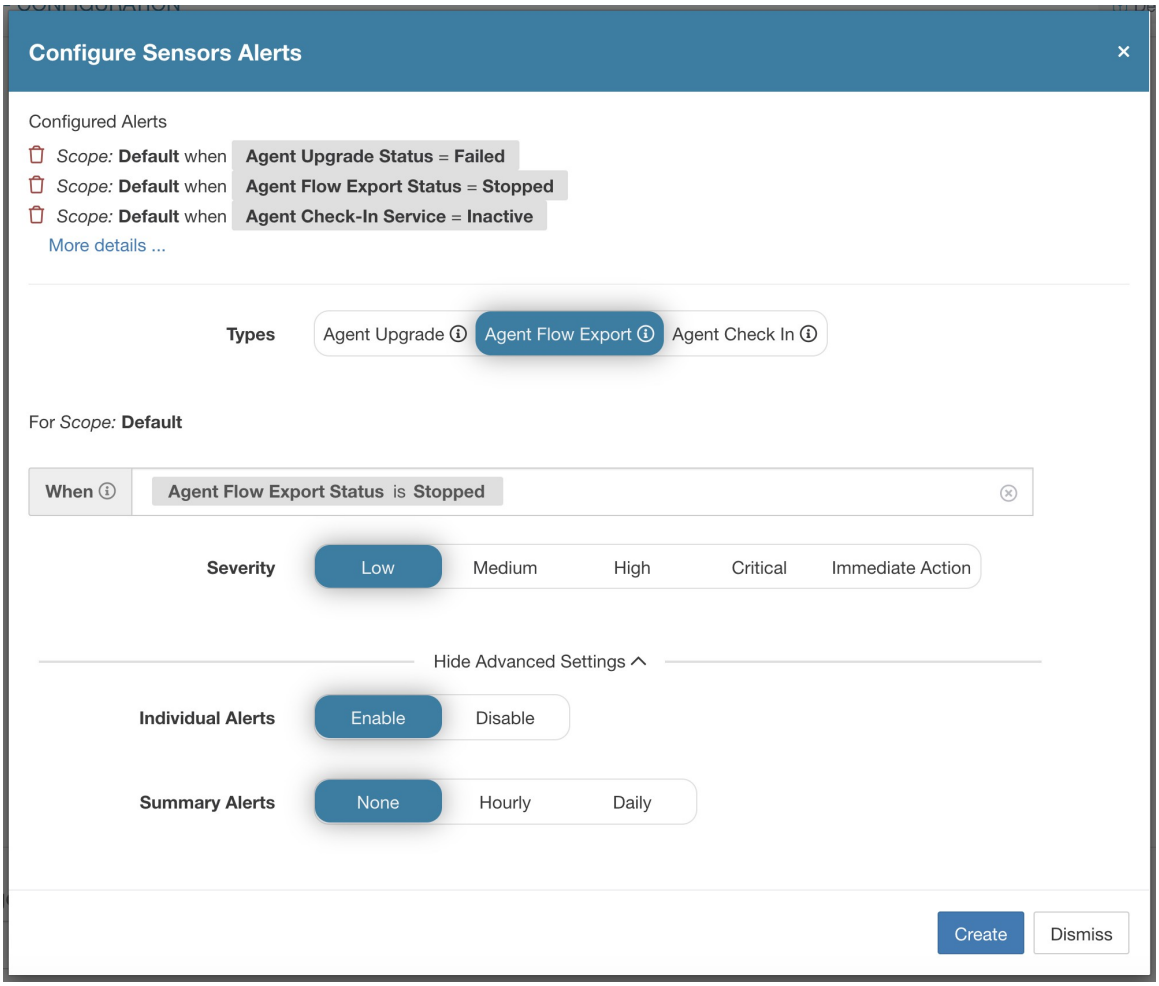
図 30 : センサーアラートの設定



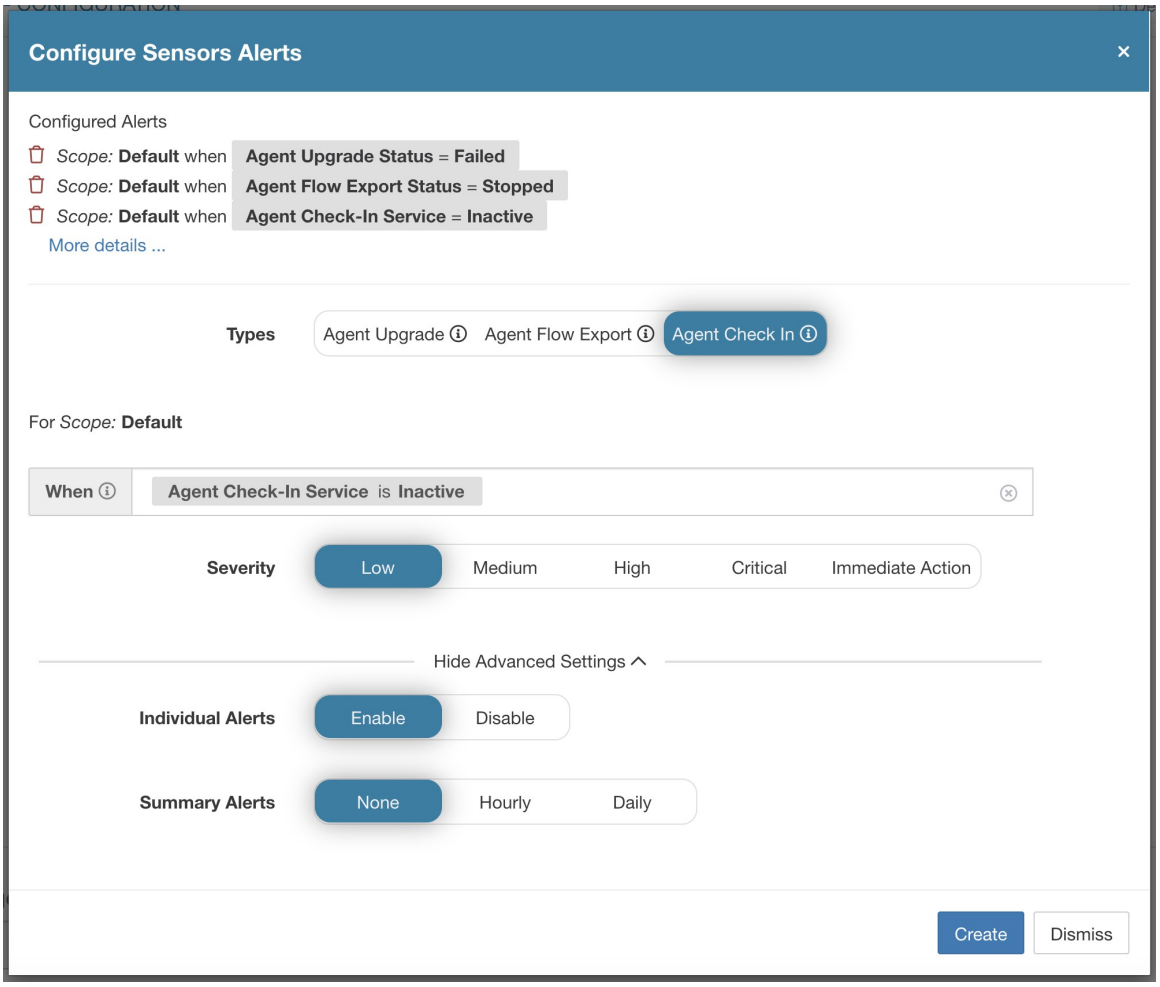
センサーアラート設定には、3 つの異なるタイプのアラートを設定する機能があり、ユーザーはアラートの重大度やその他のタイプごとの設定パラメータを設定できます。



エージェントがアップグレードに失敗したときに報告するようにセンサーアラートを設定します。このアラートは、エージェントによる目的のバージョンへのアップグレードに失敗した場合にトリガーされます。

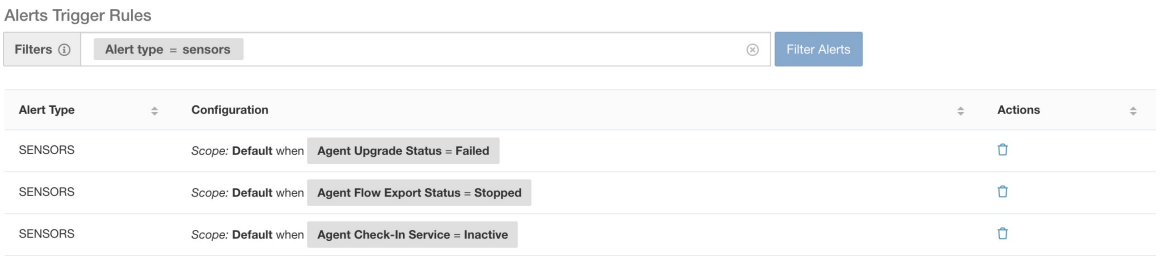


エージェントフローのエクスポートが停止したことを検出するようにセンサーアラートを設定します。このアラートは、エージェントとクラスタ間の接続がどこかでブロックされ、フローやその他のシステム情報の送信または配信が妨げられている場合にトリガーされます。



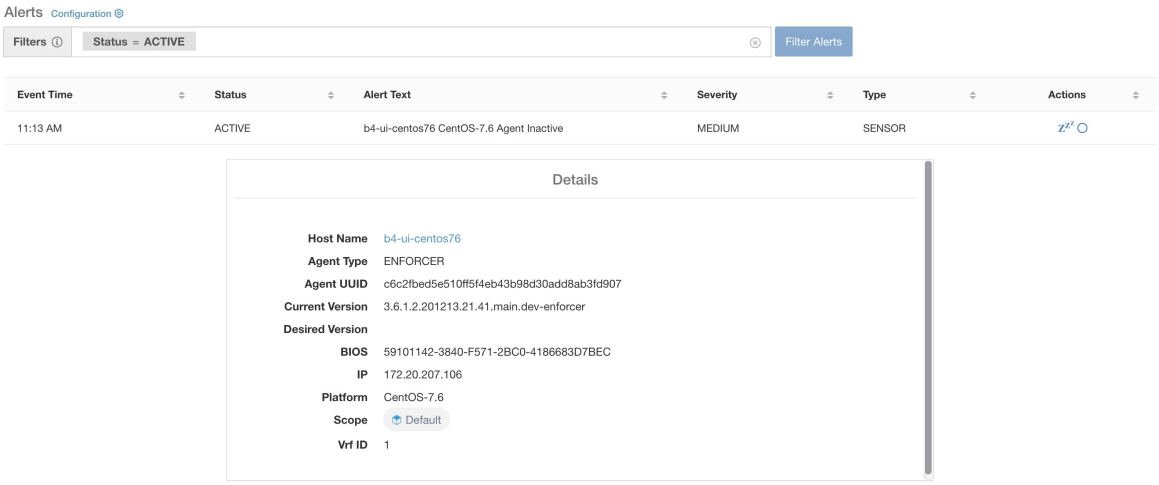
エージェントのチェックインのタイムアウトを検出するようにセンサーアラートを設定します。このアラートは、クラスタがエージェントからのチェックイン要求を 90 分以上受信していない場合にトリガーされます。

図 *31* : アラート設定ページでの設定済みセンサーアラートの表示



センサー UI アラートの詳細

図 *32* : センサーアラートの詳細



センサーアラートの詳細

一般的なアラート構造とフィールドに関する情報については、「共通アラート構造」を参照してください。alert_details フィールドは構造化されており、センサーアラートの次のサブフィールドが含まれています。

フィールド	アラートタイプ	書式	説明
AgentType	*all*	string	インストールタイプに応じて「ENFORCER」または「SENSOR」
HostName	*all*	string	エージェントが展開されているホスト名
IP	*all*	string	ノードまたはゲートウェイの IP アドレス
Bios	*all*	string	ノードの BIOS UUID
プラットフォーム	*all*	string	ノードのプラットフォームまたは OS 情報
CurrentVersion	*all*	string	ノード上のエージェントのソフトウェアバージョン

フィールド	アラートタイプ	書式	説明
DesiredVersion	*all*	string	エージェントに必要なソフトウェアバージョン
LastConfigFetchAt	*all*	integer	エージェントが最後にHTTPSリクエストを送信したときのUNIXタイムスタンプ

センサーアラートの alert_details の例

```
{
"AgentType":"SENSOR",
"Bios":"72EF1142-03A2-03BC-C2F8-F600567BA320",
"CurrentVersion":"3.5.1.1.mrpm.build.win64-sensor",
"DesiredVersion":"",
"HostName":"win2k12-production-db",
"IP":"172.26.231.193",
"Platform":"MSServer2012R2Standard"
}
```

ソフトウェアエージェントのトラブルシューティング

ここでは、ソフトウェアエージェントの展開中や運用中にお客様が直面する可能性のあるいくつかの潜在的な問題について取り上げます。また、問題のトラブルシューティングに使用できる方法、およびお客様が適用できるいくつかの解決策を記載します。

一般

ログファイル：ログファイルは通常、<install-location>/logs または <install-location>/log フォルダに保存されます。これらのログファイルは、Secure Workload サービスによって監視およびローテーションされます。

エージェントの展開

Linux

Q：コマンド「rpm -Uvh tet-sensor-1.101.2-1.el6-dev.x86_64.rpm」を実行すると、エージェントのインストールに失敗し、次のエラーがスローされました。

```
error: can’t create transaction lock on /var/lib/rpm/.rpm.lock (Permission denied).
```

A：エージェントをインストールするための適切な権限がないようです。rootに切り替えるか、sudo を使用してエージェントをインストールしてください。

Q：「sudo rpm -Uvh tet-sensor-1.0.0-121.1b1bb546.el6-dev.x86_64.rpm」を実行すると、次のエラーが発生しました。

```
Preparing. . . ########################################### [100%]
which: no lsb_release in (/sbin:/bin:/usr/sbin:/usr/bin:/usr/X11R6/bin)
error: %pre(tet-sensor-site-1.0.0-121.1b1bb546.x86_64) scriptlet failed, exit status 1
error: install: %pre scriptlet failed (2), skipping tet-sensor-site-1.0.0-121.1b1bb546
```

A：システムはエージェントをインストールするための要件を満たしていません。今回のケースでは、lsb_release ツールはインストールされていません。詳細については sw_agents_deployment-label セクションを参照し、必要な依存関係をインストールしてください。

Q：「sudo rpm -Uvh tet-sensor-1.0.0-121.1b1bb546.el6-dev.x86_64.rpm」を実行すると、次のエラーが発生しました。

```
Unsupported OS openSUSE project
error: %pre(tet-sensor-1.101.1-1.x86_64) scriptlet failed, exit status 1
error: tet-sensor-1.101.1-1.x86_64: install failed
warning: %post(tet-sensor-site-1.101.1-1.x86_64) scriptlet failed, exit status 1
```

A：お使いの OS では、ソフトウェアエージェントの実行がまだサポートされていません（今回のケースでは、「openSUSE project」はサポートされていないプラットフォームです）。詳細については sw_agents_deployment-label セクションを参照してください。

Q：すべての依存関係がインストールされており、適切な権限でインストールを実行しました。インストールはうまくいき、エラーはスローされませんでした。エージェントのインストールが本当に成功したことを確認するにはどうすればよいですか。

A：エージェントのインストール後に、このコマンドを実行して確認できます。

```
$ ps -ef | grep -e tet-sensor -e tet-engine
root 12655 1 0 08:26 ? 00:00:00 tet-engine
root 12659 12655 0 08:26 ? 00:00:00 tet-engine check_conf
root 12660 12655 0 08:26 ? 00:00:00 tet-sensor -f sensor.conf
```

3 つのエントリが表示されます。2 つは tet-engine プロセス用で、1 つは tet-sensor プロセス用です。それらのエントリが実行されていない場合は、ディレクトリ /usr/local/tet が存在しているかどうかを確認してください。存在していない場合は、インストールが失敗している可能性があります。

Windows

Q：PowerShell エージェントインストーラ スクリプトを実行すると、次のいずれかのエラーが表示されます。

1. 基礎となる接続がクローズしました。受信時に予期せぬエラーが発生しました。
2. 共通のアルゴリズムがないため、クライアントとサーバーが通信できません

A：ホストとサーバーで設定されている SSL/TLS プロトコルが不一致である可能性が高いです。次のコマンドを使用して、SSL/TLS バージョンを確認できます。

```
[Net.ServicePointManager]::SecurityProtocol
```

サーバーと一致するように SSL/TLS を設定するには、次のコマンドを使用できます（これは永続的な変更ではなく、現在の PowerShell セッションでの一時的な変更であることに注意してください）。

```
[Net.ServicePointManager]::SecurityProtocol =
[System.Net.SecurityProtocolType]'Ssl3,Tls,Tls11,Tls12'
```

Q：ダウンロードしたバンドルから MSI インストーラを実行すると、次のエラーが表示されます。

```
This installation package could not be opened. Verify that the package exists and that
you can access it, or contact the application vendor to verify that this is a valid
Windows Installer package.
```

A：*C:\Windows\Installer* パスが存在することを確認してください。コマンドラインから MSI インストーラを実行する場合は、msi ファイルを指定するときに相対パスを含めないようにしてください。正しい構文の例を示します。

```
msiexec /i "TetrationAgentInstaller.msi" /l*v "msi_install.log" /norestart
```

Q：基盤となる NIC が Nutanix VirtIO ネットワークドライバの場合、Windows Sensor ソフトウェアのアップグレードに失敗するようです。

A：Npcap 0.9990 と、Nutanix VirtIO ネットワークドライバ 1.1.3 以前のバージョンの間には、非互換性の問題があります。また、Receive Segment Coalescing が有効になっています。

この問題を解決するには、Nutanix VirtIO ネットワークドライバをバージョン 1.1.3 以降にアップグレードします。

Q：Windows Sensor をインストールしました。そのセンサーが登録されていないようであり、sensor_id ファイルには uuid-invalid-platform が含まれています。

A：Windows の PATH 変数に system32 がない可能性があります。system32 が PATH にあるかどうかを確認してください。ない場合は、次のコマンドを実行します。

```
set PATH=%PATH%;C:\Windows\System32\
```

Kubernetes

Kubernetes Daemonset のインストール中にインストーラスクリプトが失敗する場合、多くの理由が考えられます。

Q：ノードから到達可能なイメージを提供する Docker レジストリはありますか。

A：Cisco Secure Workload クラスタからイメージをプルするクラスタに関するダイレクトまたは HTTPS プロキシの問題をデバッグします。

Q：コンテナランタイムは SSL/TLS の安全でないエラーを報告していますか。

A：コンテナランタイムの適切な場所にあるすべての Kubernetes ノードに、Secure Workload HTTPS CA 証明書がインストールされていることを確認します。

Q：Docker レジストリ認証とイメージダウンロードの許可が失敗しましたか。

A：各ノードから、Helm チャートによって作成されたシークレットからの Docker プルシークレットを使用して、Daemonset 仕様のレジストリ URL から Docker が手動でイメージをプルするようにしてみてください。手動でのイメージプルも失敗した場合は、問題をさらにデバッグ

するため、Secure Workload クラスタの registryauth サービスからログをプルする必要があります。

Q：Kubernetes クラスタは Secure Workload アプライアンス内で正常にホストされていますか。

A：クラスタの [サービスステータス（service status）] ページをチェックして、関連するすべてのサービスが正常であることを確認します。[探索（explore）] ページから dstool スナップショットを実行し、生成されたログを取得します。

Q：Docker イメージビルダデーモンは実行されていますか。

A：dstool ログから、ビルドデーモンが実行されていることを確認します。

Q：Docker イメージをビルドするジョブは失敗しますか。

A：dstool ログから、イメージがビルドされていないことを確認します。Docker ビルドポッドログを使用して、ビルドキットを構築中のエラーをデバッグできます。適用コーディネータのログを使用して、ビルドの失敗をさらにデバッグすることも可能です。

Q：Helm チャートを作成するジョブは失敗しますか。

A：dstool ログから、Helm チャートが構築されていないことを確認します。適用コーディネータのログには Helm 構築ジョブの出力が含まれます。Helm チャートの構築ジョブの失敗の正確な理由をデバッグするために使用できます。

Q：インストール bash スクリプトが壊れていましたか。

A：インストール bash スクリプトのダウンロードを再試行します。bash スクリプトには、追加のバイナリデータが含まれています。bash スクリプトをテキストエディタで編集したり、テキストファイルとして保存したりすると、バイナリデータの特殊文字がテキストエディタによって破損または変更される可能性があります。

Q：Kubernetes クラスタ設定：亜種とフレーバーが多すぎます。クラシック K8 をサポートしています。

A：お客様が Kubernetes の亜種を実行している場合、展開のさまざまな段階で多くの障害モードが発生する可能性があります。失敗の段階を分類します：kubectl コマンド実行の失敗、helm コマンド実行の失敗、ポッドイメージのダウンロードの失敗、ポッドの特権モードオプションの拒否、ポッドイメージの信頼コンテンツ署名の失敗、ポッドイメージのセキュリティスキャンの失敗、ポッドバイナリ実行の失敗（アーキテクチャの不一致）、ポッドを実行しても Secure Workload サービスの開始が失敗、Secure Workload サービスが開始しても異常な動作環境が原因でランタイムエラーが発生。

Q：Kubernetes RBAC 資格情報が失敗していますか。

A：特権 DaemonSet を実行するには、K8s クラスタに対する管理者権限が必要です。kubectl 設定ファイルに、ターゲットクラスタおよびそのクラスタの管理者と同等のユーザーを指すデフォルトのコンテキストがあることを確認します。

Q：BusyBox イメージは、すべてのクラスタノードから利用可能またはダウンロード可能ですか。

A：接続の問題を修正し、BusyBoxイメージがダウンロードできることを手動でテストします。ポッド仕様で使用されているBusyBoxの正確なバージョンは、すべてのクラスタノードで使用可能（事前シード済み）またはダウンロード可能である必要があります。

Q：インストール中に、API サーバーおよびetcdエラーまたは通常のタイムアウトが発生しましたか。

A：Kubernetesクラスタ内のすべてのノードでDaemonSetポッドがインスタンス化されるため、クラスタのCPU/ディスク/ネットワークの負荷が突然スパイクする可能性があります。この問題は、お客様固有のインストールの詳細に大きく依存します。過負荷が原因で、インストールプロセス（すべてのノードでプルされてディスクに書き込まれるイメージ）に時間がかかりすぎたり、Kubernetes API サーバー、またはSecure Workload Docker レジストリエンドポイントが過負荷になったり、プロキシサーバー（設定されている場合）が一時的に過負荷になったりする可能性があります。すべてのノードでイメージのプルが完了し、Kubernetesクラスタノードの CPU/ディスク/ネットワークの負荷が軽減されるのを少し待ってから、インストールスクリプトを再試行します。API サーバーと Kubernetes コントロールプレーンからの etcd エラーは、Kubernetes コントロールプレーンノードがアンダープロビジョニングであるか、アクティビティの突然のスパイクの影響を受けている可能性があることを示しています。

Q：Secure Workload エージェントの操作でランタイムの問題が発生していますか。

A：ポッドが正しく展開され、エージェントの実行が開始されているが、ランタイムの問題が発生している場合は、Linuxエージェントのトラブルシューティング セクションを参照してください。Kubernetesの展開が正常にインストールされ、ポッドが開始された後のトラブルシューティング手順は同じです。

異常タイプ

これらは、Secure Workloadエージェントの使用および管理時にワークフローで発生する最も一般的な問題です。

非アクティブなエージェント

エージェントによるクラスタサービスへのチェックが停止しています。この現象は次のような原因で発生する可能性があります。

- ホストがダウンしている可能性がある
- ネットワーク接続が壊れているか、ファイアウォールルールによってブロックされている
- エージェントサービスが停止している

すべてのプラットフォーム

- ホストがアクティブで正常であることを確認します
- エージェントサービスが稼働状態であることを確認します
- クラスタへのネットワーク接続が機能していることを確認します

アップグレードの失敗

エージェントのアップグレードに失敗しました。これは、次のようないくつかのケースでトリガーされます。

- チェックインスクリプトがパッケージのダウンロードを試行したときにパッケージが見つからない - アップグレードパッケージを解凍できないか、パッケージ内のインストーラを検証できません。
- OS の問題または依存関係によるインストールプロセスの失敗。

Windows

- CA ルート証明書が存在しない：証明書の問題
- エージェントが最初に MSI インストールパッケージを使用して手動でインストールされた場合は、ユーザーガイドの「プラットフォームが現在サポートされているかどうかを確認する」で、Windows エディションがサポートされているプラットフォームのリストと一致するかどうかを確認します。
- OS が Windows インストーラ操作用に正しく構成されていることを確認します：Windows Installer の問題
- ホストに十分な空きディスク領域があることを確認します。

Linux

- 最後のエージェントのインストール以降にホスト OS がアップグレードされている場合は、ユーザーガイドの「プラットフォームが現在サポートされているかどうかを確認する」で、現在のリリースがサポートされているプラットフォームのリストと一致していることを確認します。
- 前回のインストール以降、必要な依存関係が変更されていないことを確認します。これらの依存関係を再確認するには、-no-install オプションを指定してエージェント インストーラ スクリプトを実行します。
- ホストに十分な空きディスク領域があることを確認します。

AIX

- 前回のインストール以降、必要な依存関係が変更されていないことを確認します。これらの依存関係を再確認するには、-no-install オプションを指定してエージェント インストーラ スクリプトを実行します。
- ホストに十分な空きディスク領域があることを確認します。

コンバート失敗

現在のエージェントタイプが目的のエージェントタイプと一致しないため、コンバートの試行がタイムアウトしました。この問題は、エージェントがパッケージをダウンロードするために check_in を実行したとき、または wss サービスが convert_commnad をエージェントにプッシュできなかったときに、通信の問題が原因で発生する可能性があります。

すべてのプラットフォーム

- ユーザーガイドの「プラットフォームが現在サポートされているかどうかを確認する」で、現在のリリースおよびエージェントタイプが、サポートされているプラットフォームのリストと一致していることを確認します。

機能の変換

エージェントをあるタイプ（優れた可視性など）から別のタイプ（適用など）に変換する能力は、すべてのエージェントが利用できるわけではありません。変換を実行できないエージェントで変換が要求された場合、異常が報告されます。

同期されていないポリシー

エージェントによって最後に報告された現在のポリシー（NPC）バージョンが、クラスターで生成された現在のバージョンと一致しません。これは、エージェントとクラスタ間の通信エラー、エージェントがローカルファイアウォールでポリシーを適用できない、またはエージェントの適用サービスが実行されていないことが原因である可能性があります。

Windows

- 適用モードが WAF の場合、ファイアウォールの有効化、ルールの追加（ルールの保持をオフの状態で）、またはデフォルトアクションの設定を妨げる GPO がホストに存在しないことを確認します。GPO の設定
- ホストとクラスタの間に接続があることを確認します。 SSL のトラブルシューティング
- 生成されたルール数が **2000** 未満であることを確認します。
- WindowsAgentEngine サービスが実行されていることを確認します。*sc query windowsagentengine*
- 利用可能なシステムリソースがあることを確認します。

Linux

- *iptables* および *ipset* コマンドを使用して、iptables および ipset が存在することを確認します。
- ホストとクラスタの間に接続があることを確認します。 SSL のトラブルシューティング
- tet-enforcer プロセスが実行されていることを確認します。*ps -ef | grep tet-enforcer*

AIX

- *ipf -V* コマンドを使用して、ipfilter がインストールされ、実行されていることを確認します。
- ホストとクラスタの間に接続があることを確認します。 SSL のトラブルシューティング
- tet-enforcer プロセスが実行されていることを確認します。*ps -ef | grep tet-enforcer*

フローのエクスポート：Pcap オープン

Secure Workload エージェントが Pcap デバイスをオープンしてフローをキャプチャできない場合、エージェントログにエラーが表示されます。Pcapデバイスが正常に開かれた場合は、次のように報告されます。

Windows ログ：*C:\Program Files\Cisco Tetration\Logs\TetSen.exe.log*

```
I0609 15:25:52.354 24248 Started capture thread for device <device_name>
I0609 15:25:52.354 71912 Opening device {<device_id>}
```

Linux ログ：*/usr/local/tet/logs/tet-sensor.log*

```
I0610 03:24:22.354 16614 Opening device <device_name>
[2020/06/10 03:24:23:3524] NOTICE: lws_client_connect_2: <device_id>: address 172.29.
‹→136.139
```

フローエクスポート：HTTPS 接続

エージェントとクラスタ間の接続は外部でブロックされているため、フローやその他のシステム情報が配信されません。これはホスト上のネットワーク ファイアウォール、SSL復号化サービス、またはサードパーティのセキュリティエージェントに関する 1 つ以上の設定の問題が原因で発生します。

- エージェントとクラスタの間に既知のファイアウォールまたはSSL復号化セキュリティデバイスがある場合は、すべての Secure Workload コレクタと VIP の IP アドレスへの通信が許可されていることを確認してください。オンプレミスクラスタの場合、コレクタのリストは、Secure Workload Web インターフェイスの左側にあるナビゲーションバーの、**[トラブルシューティング（Troubleshoot）] > [仮想マシン（Virtual Machines）]** の下に表示されます。collectorDatamover-* を参照してください。Secure Workload クラウドの場合、許可する必要があるすべての IP アドレスがポータルにリストされます。
- SSL 復号化があるかどうかを判断するために、openssl s_client を使用して接続を確立し、返された証明書を表示できます。チェーンに証明書が追加されると、エージェントのローカル CA によって拒否されます。 SSL のトラブルシューティング

証明書の問題

Windows

MSI インストーラの証明書の問題

MSI インストーラは、コード署名証明書を使用して署名されています。

MSI インストーラの場合、バージョン 3.6.x 以降および 3.5.1.31 以降

- リーフ証明書：Cisco Systems, Inc
- 中間証明書：DigiCert Trusted G4 Code Signing RSA4096 SHA384 2021 CA1
- ルート証明書：DigiCert Trusted Root G4

MSI インストーラの場合、以前のバージョン

- リーフ証明書：Cisco Systems, Inc
- 中間証明書：Symantec Class 3 SHA256 Code Signing CA
- ルート証明書：VeriSign Class 3 Public Primary Certification Authority - G5

タイムスタンプ証明書を使用します。

MSI インストーラの場合、バージョン 3.6.x 以降および 3.5.1.31 以降

- リーフ証明書：Symantec SHA256 TimeStamping Signer - G3
- 中間証明書：Symantec SHA256 TimeStamping CA
- ルート証明書：VeriSign Universal Root Certification Authority

MSI インストーラの場合、以前のバージョン

- リーフ証明書：Symantec SHA256 Timestamping Signer - G2
- 中間証明書：Symantec SHA256 Timestamping CA
- ルート証明書：VeriSign Universal Root Certification Authority

MSIインストーラのデジタル署名が無効な場合、Windows Sensor のインストールまたはアップグレードは失敗します。次の場合、デジタル署名は無効です。

- MSI インストーラの署名ルート証明書または MSI インストーラのタイムスタンプルート証明書が「信頼されたルート証明機関」ストアにない
- MSI インストーラの署名ルート証明書または MSI インストーラのタイムスタンプルート証明書が期限切れまたは失効している

問題 1

エージェントのインストールが check_conf_update.log で「TetrationAgentInstallaer.msi is not signed properly, aborting（TetrationAgentInstallaer.msi は正しく署名されていません。中止します）」というエラーにより失敗することがある

解像度

- コマンドプロンプトからコマンド *certmgr* を実行します。
- MSI インストーラの署名ルート証明書または MSI インストーラのタイムスタンプルート証明書が [信頼されたルート証明機関（Trusted Root Certification Authorities）] ストアにあるかどうかを確認します。
- 証明書を [信頼されたルート証明機関（Trusted Root Certification Authorities）] ストアに移動します。

問題 2

Windows Sensor のアップグレードが、check_conf_update.log CERT_TRUST_STATUS.dwErrorStatus: 0x04000024 の次のエラーで失敗する

CERT_TRUST_STATUS.dwInfoStatus: 0x04000024 SignTool Error: WinVerifyTrust returned error: 0x800B010C

証明書が発行者によって明示的に取り消されています。

解像度

- コマンドプロンプトからコマンド *certmgr* を実行します。
- MSI インストーラの署名ルート証明書または MSI インストーラのタイムスタンプルート証明書が [信頼されたルート証明機関（Trusted Root Certification Authorities）] ストアにあるかどうかを確認します。
- 証明書を [信頼されたルート証明機関（Trusted Root Certification Authorities）] ストアにコピーします。

問題 3

Windows Sensor のアップグレードが、check_conf_update.log の次のエラーで失敗する

アップグレードパッケージの検証に失敗しました。” “error code after running check_conf_update = 16” を終了します。

または

signtool verify /pa /v TetrationAgentInstaller.msi によってこのエラーが生成されます。

SignTool Error: WinVerifyTrust returned error: 0x80096005

タイムスタンプの署名または証明書、あるいはその両方を検証できなかったか、形式が正しくありません。

解像度

- コマンドプロンプトからコマンド *certmgr* を実行します。
- MSI インストーラの署名ルート証明書および MSI インストーラのタイムスタンプルート証明書が「信頼されたルート証明機関」ストアにあるかどうかを確認します。

証明書が見つからない場合は、他のマシンからインポートします。

証明書をインストールするには、次の手順を実行します。

まず、稼働しているいずれかのサーバーから、証明書 VeriSign Universal Root Certificate Authority をエクスポートします。以下の手順に従います。

- コマンドプロンプトからコマンド *certmgr* を実行します。
- [信頼されたルート証明機関（Trusted Root Certification Authorities）] の下の証明書 [VeriSign Universal Root Certification Authority] を右クリックし、[すべてのタスクのエクスポート（All tasksExport）] に移動します。
- エクスポートされた証明書を稼働していないサーバーにコピーしてから、証明書をインポートします。

証明書をインストールするには、次の手順を実行します。

まず、稼働しているいずれかのサーバーから、証明書 VeriSign Universal Root Certificate Authority をエクスポートします。以下の手順に従います。

- コマンドプロンプトからコマンド *certmgr* を実行します。
- [信頼されたルート証明機関（Trusted Root Certification Authorities）] の下で [証明書（certificates）] タブを右クリックし、[すべてのタスクのインポート（All tasksImport）] に移動します。
- コピーしたルート証明書を選択し、ストアに追加します。

NPCAP インストーラの証明書の問題

Windows 2012、Windows 2012 R2、Windows 8、Windows 8.1 が該当します

NPCAP バージョン：1.55

NPCAP 署名の証明書:

- リーフ証明書：Insecure.Com LLC
- 中間証明書：DigiCert EV Code Signing CA (SHA2)
- ルート証明書：DigiCert High Assurance EV Root CA

NPCAP タイムスタンプ証明書：

- リーフ証明書：DigiCert Timestamp 2021

- 中間証明書：DigiCert SHA2 Assured ID Timestamping CA
- ルート証明書：DigiCert Assured ID Root CA

問題 1

Windows エージェントのインストールに失敗し、msi_installer.log に以下のエラーメッセージが表示されることがある

```
CheckServiceStatus : Exception System.InvalidOperationException: Service npcap was not
found on computer
'.'. -> System.ComponentModel.Win32Exception: The specified service does not exist as
an installed service
```

対処法

- コマンドプロンプトからコマンド *certmgr* を実行します。
- [信頼できるルート認証局（Trusted Root Certification Authority）] ストアの [DigiCert High Assurance EV Root CA] を選択します。
- 証明書が見つからない場合は、他のマシンからインポートします。

証明書をインストールするには、次の手順を実行します。

稼働中のいずれかのサーバーから証明書「DigiCert High Assurance EV Root CA」を最初にエクスポートします。以下の手順に従います。

- コマンドプロンプトからコマンド *certmgr* を実行します。
- [信頼できるルート認証局（Trusted Root Certification Authorities）] の下にある [DigiCert High Assurance EV Root CA] 証明書を右クリックします。
- エクスポートされた証明書を非稼働中のサーバーにコピーしてから、証明書をインポートします。

証明書をインストールするには、次の手順を実行します。

- コマンドプロンプトからコマンド *certmgr* を実行します。
- [信頼できるルート認証局（Trusted Root Certification Authorities）] の下にある [証明書（certificates）] タブを右クリックし、[すべてのタスクのインポート（All tasksImport）] に移動します。
- コピーしたルート証明書を選択し、ストアに追加します。

Windows 2008 R2 に適用

NPCAP バージョン：0.991

NPCAP 署名の証明書：

- リーフ証明書：Insecure.Com LLC

- 中間証明書：DigiCert EV Code Signing CA
- ルート証明書：DigiCert High Assurance EV Root CA

NPCAP タイムスタンプ証明書：

- リーフ証明書：DigiCert Timestamp Responder
- 中間証明書：DigiCert Assured ID CA-1
- ルート証明書：VeriSign DigiCert Assured ID Root CA

問題 1

Windows エージェントのインストールに失敗し、msi_installer.log に以下のエラーメッセージが表示されることがある

```
CheckServiceStatus : Exception System.InvalidOperationException: Service npcap was not
found on
computer '.'. —> System.ComponentModel.Win32Exception: The specified service does not
exist as an
installed service
```

対処法

- コマンドプロンプトからコマンド *certmgr* を実行します。
- [信頼できるルート認証局（Trusted Root Certification Authority）] ストアの [DigiCert High Assurance EV Root CA] を選択します。
- 証明書が見つからない場合は、他のマシンからインポートします。

証明書をインストールするには、次の手順を実行します。

稼働中のいずれかのサーバーから証明書「DigiCert High Assurance EV Root CA」を最初にエクスポートします。以下の手順に従います。

- コマンドプロンプトからコマンド *certmgr* を実行します。
- [信頼できるルート証明局（Trusted Root Certification Authorities）] の下にある [DigiCert High Assurance EV Root CA] 証明書を右クリックします。
- エクスポートされた証明書を非稼働中のサーバーにコピーしてから、証明書をインポートします。

証明書をインストールするには、次の手順を実行します。

- コマンドプロンプトからコマンド *certmgr* を実行します。
- [信頼できるルート認証局（Trusted Root Certification Authorities）] の下にある [証明書（certificates）] タブを右クリックし、[すべてのタスクのインポート（All tasksImport）] に移動します。
- コピーしたルート証明書を選択し、ストアに追加します。

Windows ホストの名前変更

シナリオ 1：Windows ホストの名前を変更した後、IP アドレスと VRF 情報が表示されない問題を修正する手順：

- TaaS UI から（IP アドレスと VRF 情報が欠落している新しいホスト名を持つ）エントリを削除します。
- Windows ホストから「Cisco Secure Workload エージェント」をアンインストールし、「Cisco Tetration」ディレクトリを削除します（通常、該当するパスは「C:Program FilesCisco Tetration」）。
- Windows ホストに「Cisco Secure Workload エージェント」をインストールします。

上記の手順を実行すると、TaaS UI でエージェントが IP アドレスと VRF 情報を使用して正常に登録されます。

シナリオ 2：計画された Windows ホストの（事前）名前変更の手順：

- Windows ホストから「Cisco Secure Workload エージェント」をアンインストールし、「Cisco Tetration」ディレクトリを削除します（通常、該当するパスは「C:Program FilesCisco Tetration」）。
- Windows ホストの名前を変更して再起動します。
- Windows ホストに「Cisco Agent」をインストールします（新しいホスト名を使用）。Secure Workload

計画されたホストの名前変更に関する上記の手順に従って、エージェントを新しいホスト名で TaaS UI に登録する必要があります。

プラットフォームが現在サポートされているかどうかを確認する

Windows

- コマンド winver.exe を実行します。
- 「サポートされているプラットフォームと要件」にリストされている内容とこのリリースを比較します。

Linux

- *cat /etc/os-release* を実行します。
- 「サポートされているプラットフォームと要件」にリストされている内容とこのリリースを比較します。

AIX

- コマンド *uname -a* を実行します
- 注：メジャーバージョンとマイナーバージョンが逆になっています。

```
p7-ops2> # uname -a
AIX p7-ops2 1 7 00F8AF944C00
```

- この例では、ホスト名の後の最初の数字がマイナーバージョン、2 番目の数字がメジャーバージョンであるため、AIX バージョン 7.1 になります。「サポートされているプラットフォームと要件」に記載されている内容とこのリリースを比較します

Windows Installer の問題

- *C:\Windows\Installer* ディレクトリが存在することを確認してください。これはファイルエクスプローラには表示されません。最も簡単な確認方法は、CMD セッションで次を実行する方法です。*dir C:\Windows\Installer*
- *Windows Installer* サービスが無効になっていないかどうかを確認します。サービスを手動に設定する必要があります。
- Windows Installer によって他のエラーがレポートされていないかどうかを確認します。**[Windowsログ（Windows Logs）]** > **[アプリケーション（Application）]** > **[送信元（Source）** > **[MsiInstaller]** で Windows システムイベントログを確認します。

必要な Windows サービス

以下は、無効になっている場合にエージェントのインストール問題に関係するサービスのリストです。優れた可視性と適用エージェントの初期インストール時およびアップグレード時に、これらのサービスが稼働中であることを推奨します。

表 *9*: 必要な *Windows* サービス

サービス	インストールの目的
デバイス セットアップ マネージャ	Npcap フィルタドライバーのインストール用のデバイスドライバ管理。
デバイスのインストールサービス	Npcap フィルタドライバのインストールにも使用されます。
Windows インストーラ	エージェントの MSI パッケージのインストールに必要です。
Windows 用ファイアウォール	WAF 適用モードに必要です。
Application Experience	システム上の機能の実行可能ファイルを決定するために使用されます。

（注） アプリケーション体験サービスは、Windows Server 2008、2008R2、2012、2012R2、および Windows 7 のみが対象です。無効にすると、Npcap のインストール中にファイルのロックが発生し、インストールに失敗する可能性があります。

Npcap の問題

Npcap は Windows エージェント専用の PCAP ツールです。エージェントサービスが開始してから 10 秒後に、Npcap のインストールまたはサポートされているバージョンへのアップグレードが試行されます。Npcap サービスのインストールまたはアップグレードが失敗した場合、エージェントは 30 分以内にインストールを再試行します。3 回失敗すると、エージェントは Npcap を以前のサポート対象バージョンにロールバックしようとします（使用可能な場合）。その後、エージェントが Npcap のインストールを試みることはありません。*C:\Program Files\Cisco Tetration\Logs\TetUpdate.exe.log* および *C:\Program Files\Cisco Tetration\Logs\npcap_install.log* を確認して、エラーを特定できます。

Npcap がアップグレードされない（手動またはエージェント経由）

- プロセスが現在 Npcap ライブラリを使用している場合、Npcap は正常にアンインストールされないことがあります。実行状況を調べるには、次のコマンドを実行します。

```
PS C:\Program Files\Npcap> .\NPFInstall.exe -check_dll
WindowsSensor.exe, Wireshark.exe, dumpcap.exe
```

プロセスが一覧表示される場合は、Npcap アップグレードを続行する前にそれらのプロセスを停止する必要があります。Npcap を使用しているプロセスがない場合、上記のコマンドにより *<NULL>* のみが表示されます。

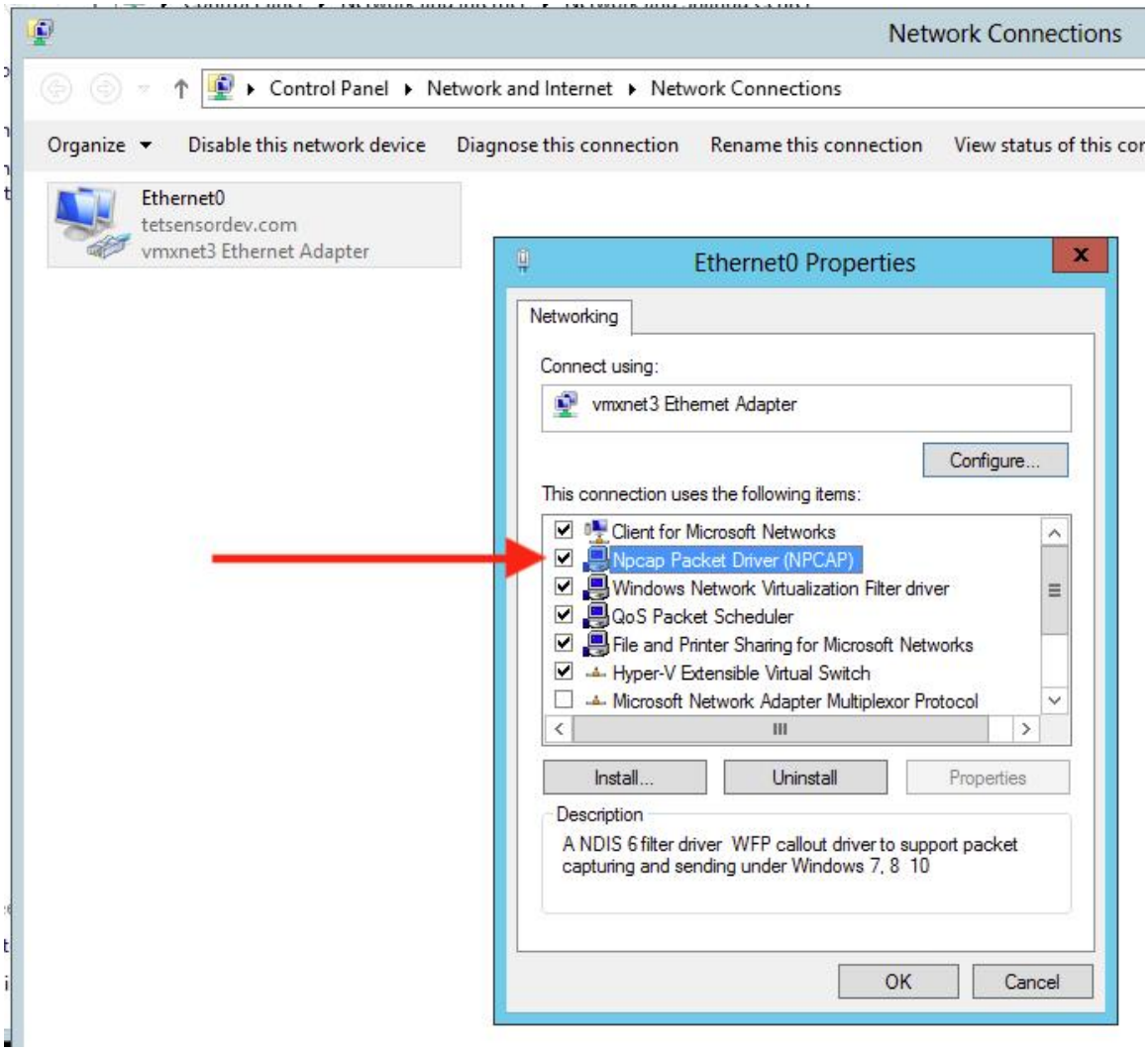
Npcap がインストールされない

- システムにインストールされている CA 証明書を確認します： NPCAP インストーラの証明書の問題
- Windows インストーラの問題を確認します：Windows Installer の問題
- システム上の他のユーザーがネットワークインターフェイスに変更を加えていないことを確認します。これが原因で COM ロックが発生し、NDIS ドライバのバインドが妨げられている可能性があります。

Npcap が完全にインストールされているかどうかの確認

ステップ 1 **[コントロールパネル（コントロールパネル）]** > **[プログラムと機能（Programs and Features）]** をチェックして、Npcap がインストールされているアプリケーションとしてリストされているかどうかを確認します。

ステップ 2 Npcap パケットドライバが問題の NIC にバインドされていることを確認します（チェックマークが表示されます）。



ステップ 3 ドライバが正しくインストールされているかどうか確認します。

```
C:\Windows\system32>pnputil -e | findstr Nmap
Driver package provider : Nmap Project
```

ステップ 4 ドライバサービスがインストールされ、実行中であるかどうかを確認します。

```
C:\Windows\system32>sc query npcap
SERVICE_NAME: npcap
        TYPE : 1 KERNEL_DRIVER
        STATE : 4 RUNNING
```

ステップ 5 レジストリエントリが存在するかどうかを確認します（Npcapがすでに存在することを確認するためにエージェントで使用されます）。

```
C:\Windows\system32>reg query HKLM\software\wow6432node\npcap
HKEY_LOCAL_MACHINE\software\wow6432node\npcap
         AdminOnly REG_DWORD 0x1
```

```
 WinPcapCompatible REG_DWORD 0x0
(Default) REG_SZ C:\Program Files\Npcap
```

ステップ 6 インストールされている Npcap プログラムファイルがすべて存在するかどうかを確認します。

```
C:\Windows\system32>dir "c:\program files\npcap"
 Directory of c:\program files\npcap
04/29/2020 02:42 PM <DIR> .
04/29/2020 02:42 PM <DIR> ..
01/22/2019 08:16 AM 868 CheckStatus.bat
11/29/2016 03:43 PM 1,034 DiagReport.bat
12/04/2018 11:12 PM 8,908 DiagReport.ps1
01/09/2019 09:22 PM 2,959 FixInstall.bat
04/29/2020 02:42 PM 134,240 install.log
01/11/2019 08:52 AM 9,920 LICENSE
03/14/2019 08:59 PM 10,434 npcap.cat
03/14/2019 08:57 PM 8,657 npcap.inf
03/14/2019 09:00 PM 74,040 npcap.sys
03/14/2019 08:57 PM 2,404 npcap_wfp.inf
03/14/2019 09:00 PM 270,648 NPFInstall.exe
04/29/2020 02:42 PM 107,783 NPFInstall.log
03/14/2019 09:01 PM 175,024 Uninstall.exe
         13 File(s) 806,919 bytes
          2 Dir(s) 264,417,628,160 bytes free
```

ステップ 7 .sys ドライバファイルが Windows のドライバフォルダにあるかどうかを確認します。

```
C:\Windows\system32>dir "C:\Windows\System32\Drivers\npcap.sys"
Directory of C:\Windows\System32\Drivers
03/14/2019 09:00 PM 74,040 npcap.sys
                 1 File(s) 74,040 bytes
```

NPCAP のインストールまたはアップグレードの実行中のネットワーク接続の問題

Windows 2016 のみに適用

サードパーティの LWF（ライトウェイトフィルタ）ドライバー（netmon など）がある場合、またはセットアップでチーミングアダプタが構成されており、エージェントの展開中にNPCAPがインストールされている場合、次のような状況が発生する可能性があります。

RDP が再接続される

NetBios サービスが再起動する

同様なネットワーク接続に関する問題

これは、Windows 2016 OS のバグが原因です。

Npcap との NIC チーミングの互換性の問題

チーミング NIC 機能は、物理 NIC（Intel、Broadcom、Realtek、MS 仮想アダプタなど）とチーミングドライバ構成 (スイッチベースでロードバランシングまたはフェールオーバーを備え、複数の NIC にパケットを分散するアルゴリズム) を基盤としています。

一部の NPCAP バージョンでは、特に下位のチーミング NIC へのバインド時に、チーミング NIC との互換性の問題があります。

現在の Secure Workload センサーソフトウェアは、Microsoft がサポートする NIC チーミングを使用してテストされています。

```
NIC type : Intel(R) 82574L Gigabit Network Connection
Teaming Mode : Switch Independent
Load Balancing Mode: Address Hash
OS : Windows 2012 , Windows 2012 R2, Windows 2016, Windows 2019
NPCAP version: 1.55
```

.

（注） Windows 2008R2 は、Microsoft がサポートする NIC チーミングに対応していません。

ネットワークフローを報告しない VDI インスタンス VM

TetSensor サービスは、NPCAP サービスが実行されているときに、複製された VM のネットワークフローをキャプチャしないことがあります。この問題は、MSIインストーラを使用して[nostart]フラグなしでエージェントがインストールされた場合、またはVMテンプレートかゴールデンイメージでPowerShellインストーラを使用し、[goldenImage]フラグなしでエージェントがインストールされた場合に発生することがあります。

この場合、Secure Workload エージェントサービスは VM テンプレートで実行を開始します。NPCAPがインストールされ、VMテンプレートのネットワークスタックにバインドされます。新しい VM が VM テンプレートから複製されると、NPCAP は新しく複製された VM のネットワークスタックに正しくバインドされません。その結果、NPCAPはネットワークフローをキャプチャできません。

解像度

ステップ 1 **管理者**権限で **cmd.exe** コマンドを実行します。

ステップ 2 **sc stop tetsensor** コマンドを実行して、TetEnforcer サービスを停止します。

ステップ 3 **sc stop tetenforcer** コマンドを実行して、TetEnforcer サービスを停止します。

ステップ 4 **C:\Program Files\Npcap\NPFInstall.exe -check_dll** を実行して、他のアプリケーションが Npcap を使用していないことを確認します。

ステップ 5 Npcap を使用してプロセスを停止します。

ステップ 6 **C:\Program Files\Npcap\NPFInstall.exe -r** を実行してバインディングを再起動します。

ステップ 7 **sc start tetsensor** コマンドを実行して、TetSensor サービスを開始します。

ステップ 8 **sc start tetenforcer** コマンドを実行して、TetEnforcer サービスを開始します。

Npcap でのネットワークパフォーマンス

Windows TetSensor サービスが実行されていると、ネットワークパフォーマンスが影響を受けることが確認されています。Windows TetSensor サービス（tetsen.exe）は、Npcap を使用して

ネットワークフローをキャプチャします。ネットワークフローをキャプチャするNpcapの実装と、tetsen.exeへのネットワークフローが、ネットワークパフォーマンスに影響を及ぼします。

TetSensor をインストール後にネットワークパフォーマンスを比較：クライアント：Windows 2016

Npcap 1.55

TetSensor 構成：適用モード WFP を使用した会話モード

サーバー：Windows 2016

Npcap 1.55

TetSensor 構成：適用モード WFP を使用した会話モード

cmd の実行：iperf3.exe -c<server_ip> -t 40

表 *10 : 121071*：*Npcap 1.55* でのネットワークパフォーマンス

設定	ネットワーク パフォーマンス
TetSensor 未インストール Npcap なし	[ID] インターバル転送帯域幅 [4] 0.00 ～ 40.00 秒 18.2 ギガバイト 3.90 ギガバイト/秒（送信者） [4] 0.00 ～ 40.00 秒 18.2 ギガバイト 3.90 ギガバイト/秒（受信者）
TetSensor インストール済み Npcap インストール済み	[ID] インターバル転送帯域幅 [4] 0.00 ～ 40.00 秒 17.3 ギガバイト 3.72 ギガバイト/秒（送信者） [4] 0.00 ～ 40.00 秒 17.3 ギガバイト 3.72 ギガバイト/秒（受信者）

Npcap 0.9990 でのネットワークパフォーマンス

TetSensor をインストール後にネットワークパフォーマンスを比較：クライアント：Windows 2016

Npcap 0.9990

TetSensor 構成：適用モード WFP を使用した会話モード

サーバー：Windows 2016

Npcap 0.9990

TetSensor 構成：適用モード WFP を使用した会話モード

cmd の実行：iperf3.exe -c<server_ip> -t 40。表：Npcap 0.9990 のネットワークパフォーマンス

class longtable

設定	ネットワーク パフォーマンス
TetSensor インストール済み Npcap インストール済み	[ID] インターバル転送帯域幅 [4] 0.00 ～ 40.00 秒 16.3 ギガバイト 3.50 ギガバイト/秒（送信者） [4] 0.00 ～ 40.00 秒 16.3 ギガバイト 3.50 ギガバイト/秒（受信者）

（注） パフォーマンスは、インストールされている Windows Npcap バージョン、Windows OS、およびネットワーク構成によって異なる場合があります。

OS のパフォーマンスや安定性の問題

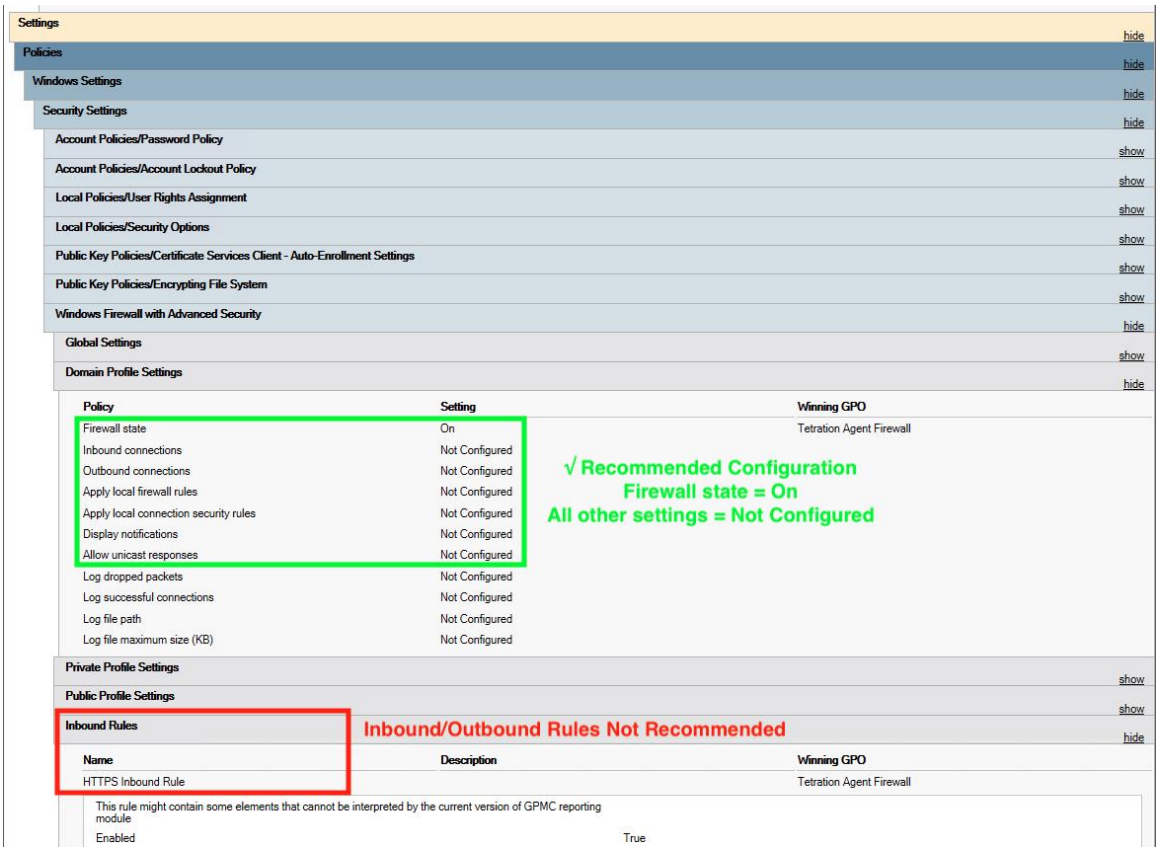
インストールされている NPCAP のバージョンや構成が Secure Workload ソフトウェアでサポートされていない場合、OS で未知のパフォーマンス問題や安定性の問題が発生する可能性があります。

サポートされている NPCAP バージョン：0.991 および 1.55

GPO の設定

ポリシーを適用するエージェントでは、ローカル設定または GPO のいずれかを使用してファイアウォールのみを有効にする必要があります。他のすべての GPO 設定は行わず、「未設定」のままにしておく必要があります。

- GPO 設定が適用をブロックしているかどうかを確認するには、*C:\Program Files\Cisco Tetration\Logs\TetEnf.exe.log* のログを確認し、次のエラー例がないかを探します。
- **「ルールを保持 = いいえ」の設定でルールの競合が発生:**「グループポリシーに設定されているファイアウォールがあります。Secure Workload エージェントには、これを削除する権限がありません。」
- **ファイアウォールがオフに設定されている:**「GPO が DomainProfile のファイアウォールを無効にしています」
- **デフォルトアクションが設定されている:**「グループポリシーは、DomainProfile のデフォルト受信アクションと競合しています」
- ホストに適用されている GPO ポリシーを確認するには、*gpresult.exe /H gpreport.html* を実行し、生成された HTML レポートを開きます。以下の例では、「ルールを保持」が「いいえ」に設定されている場合、*Secure Workload* エージェントのファイアウォールが適用する受信ルールは、適用エージェントと競合します。



クラスタ通信へのエージェント

Secure Workload エージェントは、複数のチャネルを介してクラスタへの接続を維持します。エージェントの種類によって、接続数は異なります。

接続のタイプ

- **WSS**：クラスタへのポート 443 を介した永続的なソケット接続
- **チェックイン**：15 ～ 20 分ごとにクラスタへの HTTPS コールを行い、現在の構成を確認し、更新を確認し、クラスタに対してエージェントのアクティブ状態を更新します。これは、アップグレードの失敗も報告します。
- **フローエクスポート**：フローメタデータをクラスタに送信するための、ポート 443（TaaS）または 5640（オンプレミス）を介した永続的な SSL 接続
- **適用**：ポート 443（Taas）または 5660（オンプレミス）を介した永続的な SSL 接続により、適用ポリシーを取り込み、適用状態を報告します。

接続状態の確認

Teration UI は、非アクティブな（チェックインしなくなった）エージェント、（[統計（Stats）] の [エージェントワークロードプロファイル（Agent Workload Profile）] ページにある）エクスポートされていないフロー、または失敗した適用のいずれかを報告します。エラーに応じてワークロードのさまざまなログを確認し、問題の原因を特定することができます。

非アクティブなエージェント

Windows ログ：*C:\Program Files\Cisco Tetration\Logs\check_conf_update.log*

Linux ログ：*/usr/local/tet/logs/check_conf_update.log*

HTTP 応答コード 304 が想定されており、設定の変更がないことを意味します。エラーコード =2 も同様に想定されます。その他の HTTP 応答コードは、Secure Workload クラスタ上の WSS サービスと通信する際の問題を示します。

```
Tue 06/09/2020 17:25:25.08 check_conf_update: "curl did not return 200 code, it's 304,
‹→ exiting"
Tue 06/09/2020 17:25:25.08 check_conf_update: "error code after running check_conf_
‹→update = 2"
```

- [304] 想定されており、設定の変更はありません。チェックインに成功しました
- [401] 登録に失敗しました。アクティベーションキー（TaaS） がありません
- [403] エージェントはすでに同じ UUID でクラスタに登録されています
- [000] SSLでの接続の問題を示します。curl が WSS サーバーに到達できなかったか、証明書に問題があります。SSLのトラブルシューティングで、SSLのトラブルシューティングを確認してください。

エクスポートされていないフロー

Windows ログ：*C:\Program Files\Cisco Tetration\Logs\TetSen.exe.log*

Linux ログ：*/usr/local/tet/logs/tet-sensor.log*

以下は WSS への接続が成功したことを示します

```
cfgserver.go:261] config server: StateConnected, wss://<config_server_ip>:443/wss/
‹→<sensor_id>/forensic, proxy:
```

以下はコレクタへの接続が成功したことを示します

```
collector.go:258] next collector: StateConnected, ssl://<collector_ip>>:5640
```

WSS またはコレクタへの接続中にエラーが発生した場合は、ファイアウォールの設定を確認するか、エージェントと Secure Workload の間で SSL 復号化が行われていないかどうかを確認してください。 SSL のトラブルシューティング を参照してください。

ポリシー適用の失敗

Windows ログ：*C:\Program Files\Cisco Tetration\Logs\TetEnf.exe.log*

Linux ログ：*/usr/local/tet/logs/tet-enforcer.log*

```
ssl_client.cpp:341] Successfully connected to EFE server
```

EFE サーバーへの接続中にエラーが発生した場合は、ファイアウォールの設定を確認するか、エージェントと Secure Workload の間で SSL 復号化が行われていないかどうかを確認してください。 SSL のトラブルシューティング を参照してください。

SSL のトラブルシューティング

エージェント通信の概要

Secure Workload エージェントは、TLS を使用して Secure Workload Cloud SaaS サーバーへの TCP 接続を保護します。これらの接続は、次の 3 つの特徴的なチャネルに分類されます。

- エージェント -> ポート TCP/443（TLS）（sensorVIP）経由の Cisco Secure Workload SaaS 制御チャネル

  これは、エージェントが Secure Workload に登録できるようにする低ボリュームの制御チャネルであり、設定のプッシュとソフトウェアアップグレード通知も処理します。

- エージェント -> TCP/443（TLS）（コレクタ）経由の Cisco Secure Workload SaaS フローデータ

  フローデータは、抽出されたフローメタデータ情報であり、このデータは一度に 16 個の IP アドレスのセット 1 つに送信されます。2 番目の IP アドレスのセットはスタンバイ用です。これは、実際のサーバートラフィックの約 1 ～ 5% を占めます。

- エージェント -> TCP/443（TLS）（efe）を介した Cisco Secure Workload SaaS 適用データ

  適用データチャネルは、ポリシーをセンサーにプッシュし、適用統計を収集するために使用される低ボリュームの制御チャネルです。

センサーは、エージェントとともにインストールされているローカル CA に対して、Secure Workload クラウドの制御、データ、および適用サーバーからの TLS 証明書を検証します。他の CA は使用されないため、エージェントに送信される他の証明書は検証に失敗し、エージェントは接続されません。結果的に、エージェントは登録、チェックイン、フローの送信、または適用ポリシーの受信を行わなくなります。

エージェント通信の IP トラフィックの構成

多くの場合に使用される一般的な構成は、エージェント（ワークフロー）と Secure Workload TaaS の間に境界ファイアウォールと、場合によってはプロキシを配置することです。

（注） Secure Workload は、オンボーディング中にもゲートウェイ/NAT IP 情報を収集し、テナント作成時に情報を自動的に追加します。ポータルで新しい IP アドレスを追加するか IP アドレスを変更する場合、変更には Secure Workload スタッフによる確認と承認が必要です。

TaaS ポータルでゲートウェイ/NAT IP アドレスを追加することに加えて、アウトバウンドトラフィックと変更されていないトラフィックを許可するために、ネットワークにさらに変更が必要になる場合があります。

境界ファイアウォールで TLS/HTTPS を介したアウトバウンドポート 443 を許可する

復号 Web プロキシが使用されている場合は、Web プロキシでプロキシバイパスと SSL/TLS バイパスを構成します。

（注） データセンターで透過的な Web プロキシを使用している場合は、特定の SaaS IP アドレスをルーティングし、バイパスルールを構成する必要があります。センサーは、自動 HTTPS リダイレクトを実行できない接続です。

エージェントの通信先 IP のリストは、TaaS ポータルで入手できます。ファイアウォールのアウトバウンド構成とプロキシバイパスに追加する IP には、collector-n、efe-n（適用が展開されている場合のみ）、および sensorVIP というラベルが付けられます。通常、エージェント通信用に追加する IP は 17 ～ 33 個ありますが、TaaS 構成によってはそれ以上または以下になる可能性があります。

SSL/TLS 接続のトラブルシューティング

前のセクションで説明したとおり、エージェント通信の SSL/TLS 復号化をバイパスするために、明示的または透過的な Web プロキシを設定することが重要です。バイパスが設定されていない場合、これらのプロキシは復号化を試みる可能性があります

SSL/TLS トラフィックは、自身の証明書をエージェントに送信することによるトラフィックです。エージェントはローカル CA のみを使用して証明書を検証するため、これらのプロキシ証明書によって接続エラーが発生します。

症状には、エージェントがクラスタに登録できない、エージェントがチェックインしない、エージェントがフローを送信しない、（適用が有効になっている場合）エージェントが設定の適用を受信しないなどがあります。

（注） 以下のトラブルシューティング手順は、デフォルトのインストールパスが使用されたことを前提としています。Windows: C:Program FilesCisco Tetration Linux: /usr/local/tet. エージェントを別の場所にインストールした場合は、手順をその場所に置き換えてください。

SSL/TLS 接続の問題は、エージェントログで報告されます。ログに SSL エラーがあるかどうかを確認するには、観察された関連する問題に対して次のコマンドを実行します。

登録、チェックイン

Linux

```
grep "NSS error" /usr/local/tet/log/check_conf_update.log
```

Windows（PowerShell）

```
get-content "C:\Program Files\Cisco Tetration\logs\check_conf_update.log" | select-
‹→string -pattern "SSL Certificate problem"
```

フロー（Flows）

報告される SSL/TLS 接続問題のほとんどは、エージェントの最初の接続および登録中に発生します。フローを送信するには、接続を試みる前に登録が完了している必要があります。ここで表示される SSL/TLS エラーは、センサーの VIP IP は許可されているが、コレクタの IP が許可されていないことが原因です。

Linux

```
grep "SSL connect error" /usr/local/tet/log/tet-sensor.log
```

Windows（PowerShell）

```
get-content "C:\Program Files\Cisco Tetration\logs\WindowsSensor*.log" | select-
‹→string -pattern "Certificate verification error"
```

施行

Linux

```
grep "Unable to validate the signing cert" /usr/local/tet/log/tet-enforcer.log
```

Windows（PowerShell）

```
get-content "C:\Program Files\Cisco Tetration\logs\WindowsSensor*.log" | select-
‹→string -pattern "Handshake failed"
```

上記のログチェックにSSLエラーが表示された場合は、次のコマンドを使用して、エージェントに送信されている証明書を確認できます。

Explicit Proxy - where a proxy is configured in user.cfg

Linux

```
curl -v -x http://<proxy_address>:<port> https://<sensorVIP>:443
```

Windows（PowerShell）

```
cd "C:\Program Files\Cisco Tetration"
.\curl.exe -kv -x http://<proxy_address>:<port> https://<sensorVIP>:443
```

透過型プロキシ：user.cfg プロキシ設定は不要です。このプロキシは、エージェントからインターネットへのすべての HTTP（S）トラフィック間で設定されたプロキシです。

Linux

```
openssl s_client -connect <sensorVIP from TaaS Portal>:443 -CAfile /usr/local/tet/
‹→cert/ca.cert
```

Windows（PowerShell）

```
cd C:\Program Files\Cisco Tetration
.\openssl.exe s_client -connect <sensorVIP from TaaS Portal>:443 -CAfile cert\ca.cert
```

openssl s_client respose で次のものを探しています

```
Verify return code: 0 (ok)
```

エラーが表示された場合は、証明書を調べてください。証明書（チェーン）の例には、次の証明書のみを含める必要があります（CN IP は一例です）。

証明書チェーン

```
0 s:/C=US/ST=CA/L=San Jose/O=Cisco Systems, Inc./OU=Tetration, Insieme BU/CN=129.146.
‹→155.109
i:/C=US/ST=CA/L=San Jose/O=Cisco Systems, Inc./OU=Tetration Analytics/CN=Customer CA
```

追加の証明書が表示される場合は、エージェントと Cisco Secure Workload の間に Web 復号化プロキシがある可能性があります。セキュリティグループまたはネットワークグループに連絡し、上記の「エージェント通信の IP トラフィックの設定」セクションでリストされている IP を使用して、プロキシバイパスが設定されていることを確認してください。

Windows 2016 サーバーで Windows Sensor のインストールスクリプトが失敗する：「基になる接続が切断されました。受信時に予期しないエラーが発生しました」というエラーメッセージが表示される場合があります。考えられる理由は、PowerShell で設定されている SSL/TLS バージョンである可能性があります。

実行されている SSL/TLS バージョンを調べるには、次のコマンドを実行します。

```
[Net.ServicePointManager]::SecurityProtocol
```

上記のコマンドの出力が次である場合：

```
Ssl3, Tls
```

次に、以下のコマンドを使用して許可されたプロトコルを変更し、インストールを再試行してください。

```
[Net.ServicePointManager]::SecurityProtocol = [System.Net.SecurityProtocolType]'Ssl3,
‹→Tls,Tls11,Tls12'
```

エージェントの操作

Q：エージェントを正常にインストールしましたが、[UIセンサーモニタリング（UI Sensor Monitoring）] ページに表示されません。

A：エージェントが動作を開始する前に、クラスタ内で実行されているバックエンドサーバーにエージェントを登録する必要があります。エージェントがUIページに表示されない場合は、登録に失敗したことが原因であると考えられます。いくつかのポイントをチェックすることで、登録に失敗した理由を確認できます。

- エージェントとバックエンドサーバー間の接続が正しく機能しているかどうかを確認します。
- curl リクエストをバックエンドサーバーに正しく送信できるかどうかを確認します。
- HAProxyアクセスとバックエンドサーバーのログをチェックして、登録リクエストがサーバーに届いたかどうかを確認します。
- ログファイルで curl リクエストから返されたエラーを確認します。

Q：エージェントがインストールされ、UI ページでエージェントを確認できましたが、[ソフトウェアバージョン（SW Ver）] 列に、バージョンを示す文字列ではなく [初期化中（initializing）] と表示されます。

A：エージェントを最初にインストールしてバックエンドサーバーに登録した後、エージェントがバージョンを報告するようになるまでさらに 30 分かかります。

Q：エージェントは適切にアップグレードされていますが、[ソフトウェアバージョン（SW Ver）]フィールドには長時間にわたり（数時間など）古いバージョンが表示されたままになっています。

A：エージェントが正常にアップグレードされると、エージェントは curl リクエストを送信して現在実行中のバージョンを報告し、同じリクエストで新しいバージョンがあるか確認しようとします。次のようないくつかの理由により、リクエストがバックエンドに到達できなかった可能性があります。

- リクエストがタイムアウトし、時間内に応答を取得できませんでした。
- ネットワークに問題が発生し、エージェントがバックエンドサーバーに接続できませんでした。

Q：RHEL/CentOS-6.x でエージェントを実行し、正常に動作しています。OS を RHEL/CentOS-7.x にアップグレードする予定です。アップグレード後もエージェントは動作しますか？

A：現在、OS をアップグレードするシナリオ（特にメジャーリリースのアップグレード）はサポートされていません。OS のアップグレード後にエージェントを動作させるには、次の手順を実行します。

- 既存のエージェントソフトウェアをアンインストールします。
- 証明書を含むすべてのファイルをクリーンアップします。
- UI に移動し、エージェントエントリを削除します。
- OS を目的のバージョンにアップグレードします。
- 新しい OS にエージェントソフトウェアをインストールします。

Q：RHEL/CentOS-6.x でエージェントを実行し、正常に動作しています。ホストの名前を変更する予定です。名前の変更/再起動後もエージェントは動作しますか？

A：エージェント ID は、ホスト名と bios-uuid を含むホストの一意性に基づいて計算されます。ホスト名を変更すると、ホストの ID が変更されます。次の操作を実行することをお勧めします。

- 既存のエージェントソフトウェアをアンインストールします。
- 証明書を含むすべてのファイルをクリーンアップします。
- UI に移動し、古いエージェントエントリを削除します。
- Windows ホストの名前を変更して再起動します。
- エージェントソフトウェアを再インストールします。

Q：Windows ホストで、ルールの追加/削除/変更によってファイアウォールの逸脱が発生しました。ルールを探すにはどうすればよいですか？

A：逸脱が検出されると、エージェントはファイアウォールイベントの最後の 15 秒間を「C:\Windows\System32\config\systemprofile\AppData\Roaming\tet\firewall_events」に記録します。逸脱の原因となったルールは、policy_dev_<policy id>_<timestamp>.txt として作成された最後のファイルで見つかります。

Q：Windows ホストにエージェントを正常にインストールしましたが、センサーからのフローの報告が表示されません。なぜですか？

A：Windows ホストでフローを収集するには、Npcap が必要です。Npcap は、エージェントが正常にインストールされてから 10 秒後にインストールされます。数分経ってもセンサーがフローを報告しない場合は、エージェントとバックエンドサーバー間の接続が正しく機能しているかどうか、および Npcap が正しくインストールされているかどうかを確認してください。Npcap の問題

Q：Windows ホスト（2008 R2）にエージェントを正常にインストールしましたが、tetsensor サービスの実行中にシステムクロックがドリフトします。なぜですか？

A：これは、Go および Windows 2008 R2 の既知の問題です。詳細については、「Golang および Win2008 R2」を参照してください。

tetsensor サービスの一部として実行されるプロセス tet-main.exe は、Go バージョン 1.15 を使用して構築されています。そのため、tetsensor サービスの実行中にシステムクロックがドリフトします。

この問題は、Windows 2008 R2 ワークロードが外部 NTP サーバーまたはドメインコントローラを NTP サーバーとして使用するように設定されている場合に発生します。

考えられる回避策：

1. NTP に定期的にクロックを同期させます：w32tm /resync /force
2. tet-main.exe を手動で無効にします。
   - 「管理者」権限で cmd.exe を実行します。
   - regedit.exe を実行します。
   - 「HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Services\TetSensor」に移動します。
   - 「ImagePath」をダブルクリックします。
   - 値を編集し、tet-main.exe を削除します。

     ビフォー：“C:\Program Files\Cisco Tetration\TetSenEngine.exe” TetSensor TetSen.exe “-f sensor_config” tet-main.exe ” ” TetUpdate.exe

     アフター：“C:\Program Files\Cisco Tetration\TetSenEngine.exe” TetSensor TetSen.exe “-f sensor_config” TetUpdate.exe
   - tetsensor サービスを再起動します。

（注） エージェントがアップグレードされるたびに、tet-main.exe を無効にしてください。

3. 外部 NTP サーバー設定を削除します。
   - コマンドを実行します：w32tm /config /update /manualpeerlist: /syncfromflags:manual /reliable:yes
   - Windows タイムサービス、W32Time を再起動します。

この問題の詳細については、「CSCwb8009」を参照してください。

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