System Center Endpoint Protection

インストールマニュアルとユーザガイド

Red Hat Enterprise Linux Server 5, 6
SUSE Linux Enterprise 10, 11
CentOS 5, 6
Debian Linux 5, 6
Ubuntu Linux 10.04, 12.04
Oracle Linux 5, 6

Microsoft

Microsoft® System Center Endpoint Protection

目次

概要

System Center Endpoint Protectionをご使用いただき、ありがとうございます。Microsoftの最先端スキャンエンジンは、比類のない検査速度および検出率と、非常に小さいフットプリントの組み合わせにより、どのようなLinux OSサーバーに対しても理想的な選択肢となっています。

主要な機能

オンデマンドスキャナー

オンデマンドスキャナーを起動するには、特権ユーザー(通常はシステム管理者)がコマンドラインインターフェイスまたはWebインターフェイスから、あるいはオペレーティングシステムの自動スケジューリングツール(例えばcron)を使用します。*オンデマンド*という用語は、ファイルシステムオブジェクトがユーザーまたはシステムの要求によって検査されることを言います。

リアルタイム保護

リアルタイム保護は、ユーザーまたはオペレーションシステム(またはこの両方)がファイルシステムオブジェクトへのアクセスを試みたときに、その都度起動されます。またこれによって、*オンアクセス*という用語の用法も明確になります。つまり、ファイルシステムオブジェクトへのアクセスが試行された場合に必ず検査が起動するからです。

システムの主要な機能

拡張エンジンアルゴリズム

Microsoftのウイルス対策スキャンエンジンのアルゴリズムは、最大の検出率と最速の検査時間を実現します。

マルチプロセッシング

System Center Endpoint Protectionは、シングルプロセッサーおよびマルチプロセッサーのどちらのユニットでも稼動するように開発された製品です。

アドバンスドヒューリスティック

System Center Endpoint Protectionには、Win32ワーム、バックドア感染、およびその他の形式のマルウェアに対する独自のアドバンスドヒューリスティックが組み込まれています。

組み込み機能

組み込みアーカイバーは、外部プログラムを必要とすることなく、アーカイブされたオブジェクトを解凍します。

速度と効率

システムの速度と効率を高めるため、System Center Endpoint Protectionのアーキテクチャーは、実行中のデーモン(常駐プログラム)をベースにしており、すべての検査要求がそこに送信されます。

拡張セキュリティ

すべての実行デーモン(scep_dacを除く)は、セキュリティの強化のために非特権ユーザーアカウントのもとで稼動します。

選択的設定

システムは、ユーザーまたはクライアント /サーバーに基づいた選択的設定をサポートします。

複数のログレベル

システムのアクティビティーおよび侵入物に関する情報の入手のために、複数のログレベルを設定できます。

Webインターフェイス

設定と管理は、直感的で使いやすいWebインターフェイスを介して実行できます。

外部ライブラリ不要

System Center Endpoint Protectionインストールには、LIBC以外の外部ライブラリやプログラムは不要です。

ユーザー指定の通知

侵入物の検出やその他の重要イベントが発生したときに、特定のユーザーに通知するようにシステムを設定できます。

システム要件は僅少

System Center Endpoint Protectionが効率よく稼動するためには、16MBのハードディスクスペースと32BのRAMを必要とするのみです。2.2.x、2.4.xおよび2.6.x Linux OSカーネルバージョンでスムーズに動作します。

パフォーマンスとスケーラビリティ

小型の小オフィス向けサーバーから、数千人のユーザーを擁する企業向けISPサーバーにいたるまで、System Center Endpoint Protectionは、Microsoftセキュリティ製品の比類のないセキュリティ機能に加えて、UNIXベースのソリューションに対して期待されるパフォーマンスとスケーラビィティーを備えています。

用語と略語

ここでは、本書で使用されている用語と略語を説明します。太字は、製品コンポーネント名と、新たに定義された用語と略語用に予約されていることに注意してください。この章で定義する用語と略語については、本書の後半でさらに詳しく述べます。

SCEP

SCEPは、MicrosoftによってLinuxオペレーティングシステム用に開発されたセキュリティ製品の標準の頭字語です。またこれは、製品を含むソフトウェアパッケージの名前でもあります。

SCEP daemon

SCEPシステムの主要な制御および検査デーモン: *scep_daemon*.

SCEPベースディレクトリ

ウイルス定義データベースが組み込まれたSCEP読み込み可能モジュールを格納するディレクトリ。以降このディレクトリを参照する場合、略語*@BASEDIR@*を使用します。*@BASEDIR@*の値(オペレーティングシステムによって異なります)を以下に一覧表示します。

```
Linux: /var/opt/microsoft/scep/lib
```

SCEP設定ディレクトリ

System Center Endpoint Protectionの設定に関連するすべてのファイルを格納するディレクトリ。以降このディレクトリを参照する場合、略語*@ETCDIR@*を使用します。*@ETCDIR@*の値(オペレーティングシステムによって異なります)を以下に一覧表示します。

```
Linux: /etc/opt/microsoft/scep
```

SCEP設定ファイル

メインのSystem Center Endpoint Protection設定ファイル。このファイルの絶対パスは次のとおりです。

@ETCDIR@/scep.cfg

SCEPバイナリーファイルディレクトリ

関連するSystem Center Endpoint Protectionバイナリーファイルを格納するディレクトリ。以降このディレクトリを参照する場合、略語*@BINDIR@*を使用します。*@BINDIR@*の値(オペレーティングシステムによって異なります)を以下に一覧表示します。

```
Linux: /opt/microsoft/scep/bin
```

SCEPシステムバイナリーファイルディレクトリ

関連するSystem Center Endpoint Protectionシステムバイナリーファイルを格納するディレクトリ。以降このディレクトリを参照する場合、略語*@SBINDIR@*を使用します。*@SBINDIR@*の値(オペレーティングシステムによって異なります)を以下に一覧表示します。

```
Linux: /opt/microsoft/scep/sbin
```

SCEPオブジェクトファイルディレクトリ

関連するSystem Center Endpoint Protectionオブジェクトファイルおよびライブラリを格納するディレクトリ。以降このディレクトリを参照する場合、略語*@LIBDIR@*を使用します。*@LIBDIR@*の値(オペレーティングシステムによって異なります)を以下に一覧表示します。

```
Linux: /opt/microsoft/scep/lib
```

インストール

System Center Endpoint Protectionは、バイナリーファイルでとして配布されます。

```
scep.i386.ext.bin
```

上記のバイナリーファイルで、*'ext*はLinux OSディストリビューション依存のサフィックスとなっています。つまり、「deb」はDebian、「rpm」はRedHatおよびSuSE、「tgz」はその他のLinux OSディストリビューションを対象としています。

製品をインストールまたはアップグレードするには、以下のコマンドを使用します。

```
sh ./scep.i386.ext.bin
```

製品のユーザーライセンス受諾契約が表示されます。受諾契約を確認すると、インストールパッケージが現行作業ディレクトリに入れられて、パッケージのインストール、アンインストール、またはアップグレードに関する情報が画面に表示されます。

パッケージのインストールが完了したら、以下のコマンドを使って、メインSCEPサービスが実行中であることを検証できます。

```
ps -C scep_daemon
```

ENTERキーを押した後、以下(または類似)のメッセージが表示されるはずです。

```
 PID TTY TIME CMD
2226 ? 00:00:00 scep_daemon
2229 ? 00:00:00 scep_daemon
```

少なくとも2つのSCEPデーモンプロセスがバックグラウンドで稼動しています。最初のPIDは、システムのプロセスおよびスレッドマネージャーを表します。その他のものは、SCEPスキャンプロセスを表します。

言語パックのインストール

System Center Endpoint Protectionで必要な言語パックをインストールするには、以下のコマンドを使用します。

```
sh ./scep-lang.lng.bin
```

ここで、*'lng*はインポートするファイルの言語コードに置き換えてください。

*Installation completed successfully*通知が表示されたら、適宜LANGシステム変数を更新し、必要に応じて環境をアップデートします。これで、言語パックのインストールが終了します。

各言語パックには、以下のものが含まれます。

- 各国語版のWebインターフェイス
- SCEPエージェントおよびコマンドの各国語版のコンソール出力
- 各国語版のPDF文書

アーキテクチャの概要

System Center Endpoint Protectionを正常にインストールし終わったら、そのアーキテクチャをよく知る必要があります。

システムは、次のようなパーツで構成されています。

コア

System Center Endpoint Protectionのコアは、SCEPデーモン(scep_daemon)です。このデーモンは、SCEP APIライブラリのlibscep.soおよびSCEPロードモジュールem00X_xx.datを使用して、検査、エージェントデーモンプロセスの保守、サンプル提出システムの保守、ログ、通知、などのシステムの基本タスクを提供します。詳細は、*scep_daemon(8)*マニュアルページを参照してください。

エージェント

SCEPエージェントモジュールの目的は、SCEPをLinuxサーバー環境に統合することにあります。

ユーティリティ

ユーティリティモジュールは、簡単で効率の良いシステム管理を実現します。これは、ライセンス管理、隔離管理、システム設定、およびアップデートなどのシステムタスクを担当します。

設定

セキュリティシステムは、正しく設定することが非常に重要です。この章ではこの後、関係するすべてのコンポーネントについて説明します。*scep.cfg*ファイルを熟知することも強くお勧めします。このファイルには、System Center Endpoint Protectionの設定にとって不可欠な情報が格納されているからです。

製品のインストールが正常に完了したら、すべての設定コンポーネントはSCEP設定ディレクトリに保管されます。このディレクトリは、次のようなファイルで構成されます。

@ETCDIR@/scep.cfg

これは、本製品の機能のすべての重要な局面を制御する最も重要な設定ファイルです。scep.cfgファイルはいくつかのセクションで構成され、それぞれが各種のパラメータを収めています。このファイルには、1つのグローバルセクションといくつかの「エージェント」セクションが格納されています。そこでは、すべてのセクション名が大括弧で囲まれています。グローバルセクションのパラメータは、SCEPデーモンの設定オプションと、SCEP検査エンジン設定の規定値の定義に使用されます。エージェントセクションのパラメータは、コンピュータまたはその周辺(またはこの両方)におけるさまざまなタイプのデータフローを傍受したり、検査に備えてその準備をしたりするのに使用されるモジュールの設定オプションの定義に使用されます。システム設定に使用されるさまざまなパラメータに加えて、ファイルの編成を規制するルールもあることに注意してください。このファイルを最も効果的に編成する方法の詳細は、*scep.cfg(5)*および*scep_daemon(8)*マニュアルページと、関連エージェントのマニュアルページを参照してください。

@ETCDIR@/certs

このディレクトリは、認証用としてSCEP Webインターフェイスによって使用される証明書を保管するのに使用されます。詳細は、*scep_wwwi(8)*マニュアルページを参照してください。

@ETCDIR@/scripts/daemon_notification_script

SCEP設定ファイルパラメータ*'exec_script*でこのスクリプトを有効にすると、ウイルス対策システムによって侵入が検出された際に、このスクリプトが実行されます。これは、そのイベントに関する電子メール通知をシステム管理者に送信するのに使用されます。

ファイルシステムサービスとの統合

この章では、ウイルスとワームのファイルシステムへの感染に対する最も効果的な保護を実現する、オンデマンドおよびリアルタイムの保護設定について説明します。System Center Endpoint Protectionの検査機能は、オンデマンドスキャナーコマンド*'scep_scan'*と、オンアクセススキャナーコマンド*'scep_dac*によって実行されます。LinuxバージョンのSystem Center Endpoint Protectionには、プリロードのライブラリモジュール*libscep_pac.so*を使用する、追加のオンアクセススキャナー技法が備えられています。これらのコマンドすべてについて、以下に説明します。

オンデマンドスキャナー

オンデマンドスキャナーを起動するには、特権ユーザー(通常はシステム管理者)がコマンドラインインターフェイスまたはWebインターフェイスから、あるいはオペレーションシステムの自動スケジューリングツール(例えばcron)を使用します。*オンデマンド*という用語は、ユーザーまたはシステムからの要求によって検査されるファイルシステムオブジェクトのことを言います。

オンデマンドスキャナーを実行するのに特別な設定は必要ありません。SCEPパッケージのインストールが正しく完了したら、コマンドラインインターフェイスまたはスケジューラツールを使用してオンデマンドスキャナーを実行できます。コマンドラインからオンデマンドスキャナーを実行するには、次のような構文を使用します。

```
@SBINDIR@/scep_scan [option(s)] FILES
```

ここでFILESは、検査するディレクトリまたはファイル(またはこの両方)のリストです。

SCEPオンデマンドスキャナーの使用時には、複数のコマンドラインオプションを使用できます。全オプションの一覧を見るには、*scep_scan(8)*マニュアルページを参照してください。

Dazukoによるリアルタイム保護

リアルタイム保護は、ユーザーまたはオペレーションシステム(またはこの両方)からのファイルシステムオブジェクトへのアクセスによって起動されます。またこれは、*オンアクセス*という用語の説明にもなります。つまり、ファイルシステムオブジェクトへのアクセスが試行された場合に必ず、スキャナーが起動されるということです。

SCEPオンアクセススキャナーで使用される技法は、Dazuko (ダズコ)カーネルモジュールで稼動し、カーネル呼び出しの傍受をベースとします。Dazukoプロジェクトはオープンソースですが、それはソースコードが無料で配布されることを意味します。そのためユーザーは、各自のカスタムカーネル用にカーネルモジュールをコンパイルできます。Dazukoカーネルモジュールはどのscep製品にも含まれないので、オンアクセスコマンド*scep_dac*を使用するには、事前にこのモジュールをコンパイルしてカーネル内にインストールする必要があることに注意してください。Dazukoの技法により、オンアクセススキャンは、使用されているファイルシステムタイプを選びません。またこれは、Network File System (NFS)、Nettalk、およびSambaを介したファイルシステムオブジェクトの検査にも適しています。

重要: オンアクセススキャナーの設定と使用に関する詳細情報に進む前に、スキャナーとは、主に外部的にマウントされたファイルシステムの保護のために開発されテストされていることに注意を促したいと思います。外部的にマウントされたのでない複数のファイルシステムがある場合、システムがハングアップしないよう、ファイルアクセス制御からそれらを除外する必要があります。通常除外するディレクトリの例には、*'/dev*ディレクトリと、SCEPによって使用されるすべてのディレクトリがあります。

操作の原則

リアルタイム保護*scep_dac* (SCEP Dazuko-powered file Access Controller)は、ファイルシステムを継続的に監視および制御する常駐プログラムです。どのファイルシステムオブジェクトも、カスタマイズ可能なファイルアクセスイベントのタイプに基づいて検査されます。現行バージョンでは、以下のイベントタイプがサポートされます。

オープンイベント

このファイルアクセスタイプを有効にするには、scep.cfgファイルの**[fac]**セクションで、*'event_mask'*パラメータの値をopenに設定します。それによって、DazukoアクセスマスクのON_OPENビットが有効になります。

クローズイベント

このファイルアクセスタイプを有効にするには、scep.cfgファイルの**[fac]**セクションで、*'event_mask'*パラメータの値を

closeに設定します。それによって、DazukoアクセスマスクのON_OPENビットが有効になります。それによって、DazukoアクセスマスクのON_OPENビットとON_CLOSE_MODIFIEDビットが有効になります。

注意: 一部のOSカーネルバージョンは、ON_CLOSEイベントの傍受をサポートしていません。そのような場合、クローズイベントは*scep_dac*によって監視されません。

実行イベント

このファイルアクセスタイプを有効にするには、scep.cfgファイルの**[fac]**セクションで、*'event_mask'*パラメータの値をexecに設定します。それによって、DazukoアクセスマスクのON_EXECビットが有効になります。

リアルタイム保護によって、オープン、クローズ、および実行後のファイルはすべて、まず*scep_daemon*によってウイルスを検査されることになります。検査結果に応じて、個々のファイルへのアクセスが拒否または許可されます。

インストールと設定

*scep_dac*を初期化する前に、実行中のカーネル内でDazukoカーネルモジュールをコンパイルしてインストールする必要があります。Dazukoのコンパイルとインストールの方法の詳細は、以下を参照してください。

http://www.dazuko.org

Dazukoのインストールが完了したら、SCEP設定ファイル(scep.cfg)の**[global]**および**[fac]**セクションを見直して編集します。リアルタイム保護は、正しく機能するように、このファイルの**[fac]**セクション内の*'agent_type*オプションの設定から独立していることに注意してください。さらに、リアルタイム保護によって監視されるファイルシステムオブジェクト(つまりディレクトリとファイル)も定義する必要があります。このためには、同じく**[fac]**セクション内に置かれている*'ctl_incl*および*'ctl_excl*オプションのパラメータを定義します。scep.cfgファイルに変更を加えた後、SCEPデーモンの再読み込みによって、新たに作成した設定を強制的に再読み取りすることができます。

ヒント

*scep_dac*デーモンの初期化の前に必ずDazukoモジュールが読み込まれるようにするには、以下の手順を実行します。

カーネルモジュール用に予約されている以下のディレクトリのどちらかに、Dazukoモジュールのコピーを置きます。

```
/lib/modules
```

あるいは

```
/modules
```

カーネルユーティリティ'depmod'および'modprobe'を使用して(BSD OSの場合は、'kldconfig'および'kldload'を使用します)依存関係を処理し、新たに追加されたDazukoモジュールを正常に初期化します。

scep_daemon初期化スクリプト'/etc/init.d/scep_daemon'内で、デーモンの初期化ステートメントの前に以下の行を挿入します。

```
/sbin/modprobe dazuko
```

BSD OSの場合には、行

```
/sbin/kldconfig dazuko
```

を、'/usr/local/etc/rc.d/scep_daemon.sh'スクリプトに挿入します。

警告! この手順は、必ずここに示した順番どおりに実行することが非常に重要です。カーネルモジュールは、カーネルモジュールディレクトリ内にないと正しく読み込まれないので、システムがハングする原因になります。

プリロードLIBCライブラリを使用するリアルタイム保護

ここまで、Linux/BSDファイルシステムサービスの使用時のDazukoで稼動するリアルタイム保護の統合について説明してきました。次のような場合に基幹システムを保守するシステム管理者を含め、Dazukoの使用はどのような状況下でも可能というわけではありません。

- 実行中のカーネルに関連したソースコードまたは設定ファイル(またはこの両方)を利用できない場合
- カーネルがモジュール方式ではなく一体構造になっている場合
- 単に、該当するOSをDazukoモジュールがサポートしていない場合

これらの場合はいずれも、プリロードLIBCライブラリをベースにしたオンアクセス検査技法を使用する必要があります。詳細は、このセクションの以下のトピックを参照してください。このセクションは、Linux OSユーザーのみを対象とし、プリロードライブラリ*'libscep_pac.so*を使用するオンアクセススキャナーの操作、インストール、および設定に関する情報を記載していることに注意してください。

操作の原則

リアルタイム保護*libscep_pac.so*(SCEP Preload library based file Access Controller)は、システムの起動時にアクティブ化される共有オブジェクトライブラリです。このライブラリは、FTPサーバーやSambaサーバーなどのファイルシステムサーバーによってLIBC呼び出しで使用されます。各ファイルシステムオブジェクトは、カスタマイズ可能なファイルアクセスイベントのタイプに基づいて検査されます。現行バージョンでは、以下のイベントタイプがサポートされます。

オープンイベント

このファイルアクセスタイプは、esest.cfgファイル(**[fac]**セクション)内の*'event_mask'*パラメーターにワード*'open*'がある場合に有効化されます。

クローズイベント

このファイルアクセスタイプは、scep.cfgファイル(**[fac]**セクション)内の*'event_mask'*パラメーターにワード*'close*'がある場合に有効化されます。この場合、LIBCのすべてのファイル記述子とFILEストリームのclose関数が傍受されます。

実行イベント

このファイルアクセスタイプは、scep.cfg(**[fac]**セクション)内の*'event_mask'*パラメーターにワード*'exec*'がある場合に有効化されます。この場合、LIBCのすべてのexec関数が傍受されます。

オープン、クローズ、および実行の完了したすべてのファイルは、SCEPデーモンによってウイルス検査が実行されます。その検査結果に応じて、特定のファイルへのアクセスが拒否または許可されます。

インストールと設定

*libscep_pac.so*ライブラリモジュールは、プリロードライブラリの標準インストールメカニズムを使ってインストールされます。環境変数*'LD_PRELOAD*'は、*libscep_pac.so*ライブラリの絶対パスに定義する必要があります。詳細は、*ld.so(8)*マニュアルページを参照してください。

注意: *'LD_PRELOAD*環境変数は、リアルタイム保護の制御下に置かれるネットワークサーバーデーモンプロセス(ftp、Samba、など)に対してのみ定義されることが重要です。一般的に、オペレーションシステムのすべてのプロセスに対するLIBC呼び出しをプリロードすることは推奨されません。その場合、システムのパフォーマンスが大幅に低下したり、システムがハングする原因になる可能性があるからです。こうした理由から、'/etc/ld.so.preload'ファイルの使用や、'LD_PRELOAD'環境変数のエクスポートをグローバルに行わないでください。いずれの場合も、関連したすべてのLIBC呼び出しが設定変更されて、初期化中にシステムのハングアップにつながる可能性があります。

確実に特定のファイルシステム内の関連アクセス呼び出しのみが傍受されるようにするには、次のような行を使って、実行可能ステートメントを設定変更します。

```
LD_PRELOAD=@LIBDIR@/libscep_pac.so COMMAND COMMAND-ARGUMENTS
```

ここで'COMMAND COMMAND-ARGUMENTS'は、元の実行可能ステートメントです。

SCEP設定ファイル(scep.cfg)の**[global]**および**[fac]**セクションを見直して編集します。オンアクセススキャナーが正しく

機能するためには、プリロードライブラリの制御下に置かれる必要のあるファイルシステムオブジェクト(つまり、ディレクトリおよびファイル)を定義する必要があります。そのためには、SCEP設定ファイルの**[fac]**セクション内で*'ctl_incl*および*'ctl_excl*オプションのパラメーターを定義します。scep.cfgファイルに変更を加えた後、SCEPデーモンの再読み込みによって、新たに作成した設定を強制的に再読み取りすることができます。

ヒント

ファイルシステムの起動後すぐにリアルタイム保護をアクティブ化するには、適切なネットワークファイルサーバー初期化スクリプト中に、*'LD_PRELOAD*環境変数を定義する必要があります。

例: Sambaサーバーの起動後すぐに、すべてのファイルシステムアクセスイベントをオンアクセススキャナーで監視するとします。Sambaデーモン初期化スクリプト(/etc/init.d/smb)のステートメント

```
daemon /usr/sbin/smbd $SMBDOPTIONS
```

を、以下の行に置き換えます。

```
LD_PRELOAD=@LIBDIR@/libscep_pac.so daemon /usr/sbin/smbd $SMBDOPTIONS
```

このようにすれば、Sambaの制御下にある選択したファイルシステムオブジェクトは、システムの起動時に検査されます。

重要なSCEPメカニズム

オブジェクト処理ポリシー

オブジェクト処理ポリシーメカニズムは、検査済みオブジェクトの状態に基づいたフィルタリングを提供します。この機能は、以下の設定オプションをベースにします。

- action_av
- action_av_infected
- action_av_notscanned
- action_av_deleted

これらのオプションの詳細は、*scep.cfg(5)*マニュアルページを参照してください。

処理後のオブジェクトはいずれも、まず*'action_av*オプションの設定にしたがって処理されます。このオプションを*'accept*(または*'defer*? *'discard*? *'reject*)に設定すると、オブジェクトは受諾(または遅延、破棄、拒否)されます。このオプションを*'scan*'に設定すると、オブジェクトに対してウイルス侵入物の検査が実行され、*'av_clean_mode*オプションを*'はい*に設定すると、オブジェクトは駆除されます。それ以外に、オブジェクト処理をさらに評価するために、設定オプション*'action_av_infected*? *'action_av_notscanned*'、および*'action_av_deleted*も考慮に入れられます。この3つのアクションオプションの結果として*'accept*アクションがとられた場合、そのオブジェクトは受諾されます。それ以外の場合、オブジェクトはブロックされます。

ユーザー固有設定

ユーザー固有設定メカニズムの目的は、カスタマイズおよび機能をさらに強化することにあります。これを使ってシステム管理者は、ファイルシステムオブジェクトにアクセスするユーザーに基づいて、SCEPウイルス対策スキャナーのパラメーターを定義できます。

この機能の詳しい解説は、*scep.cfg(5)*マニュアルページに記載されています。ここでは、ユーザー固有設定の簡単な例だけを示します。

この例では、*scep_dac*モジュールを使用して、/homeディレクトリの下にマウントされた外部ディスクのON_OPENおよびON_EXECアクセスイベントを制御することを目的とします。モジュールは、SCEP設定ファイルの**[fac]**セクション内で設定できます。以下を参照してください。

```
[fac]
agent_type = "dazuko"
event_mask = "open"
ctl_incl = "/home"
action_av = "scan"
```

個々のユーザーを対象に検査設定を指定するには、個々の検査ルールを格納する特殊設定ファイル名を*'user_config'*パラメーターに指定する必要があります。ここに示した例では、特殊設定ファイルは*'scep_dac_spec.cfg'*という名前であり、SCEP設定ディレクトリ(このディレクトリはオペレーティングシステムをベースにします。用語と略語ページを参照)内に置かれます。

```
[fac]
agent_type = "dazuko"
event_mask = "open"
ctl_incl = "/home"
action_av = "scan"
user_config = "scep_dac_spec.cfg"
```

*'user_config*ファイルパラメーターを**[fac]**セクションで指定し終わったら、SCEP設定ディレクトリ内に*'scep_dac_spec.cfg*ファイルを作成する必要があります。最後に、任意のスキャンルールを追加します。

```
[username]
action_av = "reject"
```

特別セクションの最上部に、個々のルールの適用先のユーザー名を入力します。この設定により、ファイルシステムへのアクセスを試みる他のすべてのユーザーを正常に処理できます。つまり、他のユーザーからアクセスされるすべてのファイルシステムオブジェクトに対して侵入物の検査が実行されます。ただし、ユーザー*'username*のアクセスは拒否(ブロック)されます。

スケジューラ

スケジューラは、その機能の一環として、指定の時刻または特定のイベントの発生時にスケジュールされたタスクの実行や、事前定義の設定およびプロパティをもつタスクの管理および起動などを行います。タスクの設定とプロパティを使用して、起動の日時に影響を与えることができますが、タスクの実行中にカスタムプロファイルの使用を取り入れることによって、タスクの適用範囲を拡大することもできます。

*'scheduler_tasks*オプションは既定でコメントになっているので、既定のスケジューラ設定が適用されます。SCEP設定ファイルでは、すべてのパラメーターとタスクはセミコロンで区切られています。他のセミコロン(および円記号)はすべて、円記号でエスケープする必要があります。各タスクには6つのパラメーターがあり、その構文は次のとおりです。

- id - 固有番号。
- name - タスクの説明。
- flags - 指定したスケジューラタスクを無効化する特別フラグは、ここに設定できます。
- failstart - スケジュールした日にタスクを実行できなかった場合にどうするかを指示します。
- datespec - 6つのフィールド(crontabと同様ですが「年」が拡張されています)、繰り返し日、またはイベント名を持つ定期的な日付の指定。
- command - コマンドの絶対パスに、引数または @'プレフィックスの付いた特殊コマンド名を続けて指定できます(たとえばウイルス対策のアップデート: *@update*).

```
#scheduler_tasks = "id;name;flags;failstart;datespec;command;id2;name2;...";
```

datespecオプションの代わりに、以下のイベント名を使用できます。

- start - デーモンの起動。
- startonce - デーモンは起動しますが、1日1回が限度です。
- engine - エンジンの正常アップデート。
- login - Webインターフェイスログオンの起動。
- threat - 脅威の検出。
- notscanned - 検査済みファイルではありません。

現在のスケジューラ設定を表示するには、Webインターフェイスを使用するか、または以下のコマンドを実行します。

```
cat @ETCDIR@/scep.cfg | grep scheduler_tasks
```

スケジューラとそのパラメーターの詳しい解説は、*scep_daemon(8)*マニュアルページのスケジューラのセクションを参照してください。

Webインターフェイス

Webインターフェイスを使用すると、SCEPセキュリティシステムを簡単に設定および管理できます。このモジュールは、独立したエージェントであり、明示的に有効にする必要があります。*Webインターフェイス*を速やかに設定するには、SCEP設定ファイルに以下のオプションを設定し、SCEPデーモンを再起動します。

```
[wwwi]
agent_enabled = yes
listen_addr = address
listen_port = port
username = name
password = pass
```

イタリック体の文字をお客様独自の値に置き換え、ブラウザーを *'https://address:port'* (httpsであることに注意)に移動します。*'username/password*を指定してログインします。基本的な用法の解説はヘルプページを参照し、*scep_wwwi*の技術的な詳細は*scep_wwwi(1)*マニュアルページを参照してください。

Webインターフェイスでは、SCEPデーモンにリモートからアクセスして簡単に配置できます。この強力なユーティリティーにより、設定値を簡単に読み書きできます。

図6-1. System Center Endpoint Protection - ホーム画面

System Center Endpoint ProtectionのWebインターフェイスウィンドウは、2つのメインセクションに分かれています。プライマリウィンドウは、選択されたメニューオプションとメインメニューの内容を表示します。上部にある水平バーを使って、以下のメインオプション間を移動できます。

- [**ホーム**]- 基本システムおよびMicrosoft製品の情報を示します
- [**設定**] - ここでSystem Center Endpoint Protectionのシステム設定を変更することができます
- [**コントロール**] - 簡単なタスクを実行したり、scep_daemonで処理されるオブジェクトについてのグローバル統計を表示したりすることができます
- [**ヘルプ**] - System Center Endpoint ProtectionWebインターフェイスについての詳細使用方法が利用できます
- **ログアウト** - 現在のセッションを終了するのに使用します

重要: Webインターフェイスの[**設定**]セクションで変更を行った後は、必ず[**変更を保存**]ボタンをクリックして、新しい設定を保存してください。設定を適用するには、左ペインの[**変更を適用**]をクリックして、SCEPデーモンを再起動する必要があります。

リアルタイム保護設定の例

SCEPを設定するには、2とおりの方法があります。ここでは、そのどちらかの方法を使用して、「プリロードLIBCライブラリを使用するリアルタイム保護」の章に説明されているアクセス制御機能を設定する方法を示します。ご自分に最も適したオプションを選択できます。

- SCEP設定ファイルを使用する方法:

```
[fac]
agent_type = "preload"
event_mask = "open"
ctl_incl = "/home"
action_av_deleted = "reject"
action_av = "scan"
action_av_infected = "reject"
```

- Webインターフェイスを使用する方法:

図6-3. [SCEP] - [設定] > [オンアクセススキャナー]

Webインターフェイスで変更を行った場合は、必ず[**変更を保存**]ボタンをクリックして、その設定を保存してください。新しい変更を適用するには、[**設定**]セクションパネル内の[**変更を適用**]ボタンをクリックします。

オンデマンドスキャナー

ここでは、オンデマンドスキャナーを実行してウイルスを検査する方法に関する例を示します。

- [**コントロール**] > [**オンデマンド検査**]と移動します。
- 検査するディレクトリへのパスを入力します。
- [**ファイルを検査**]ボタンをクリックして、コマンドラインスキャナーを実行します。

図6-4. [SCEP] - [コントロール] > [オンデマンドスキャナ]

Microsoftコマンドラインスキャナーが自動的にバックグラウンドで実行されます。検査の進捗状況を見るには、[**表示**]リンクをクリックします。別のブラウザーウィンドウが開きます。

スケジューラ

スケジューラタスクを管理するには、SCEP設定ファイルを使用する(「スケジューラ」の章を参照)か、またはWebインターフェイスを使用します。

図6-5. [SCEP」 -[グローバル] > [スケジューラ]

スケジュールタスクを有効化または無効化するには、チェックボックスをクリックします。既定では、次のスケジュールタスクが表示されます。

- **ログの保守** - ハードディスクの容量を節約するために、古いログは自動的に削除されます。スケジューラは、ログのデフラグを開始します。このプロセスによって、空のログエントリは除去されます。これにより、ログでの作業時の速度があがります。大量のエントリがログ内にある場合、著しい改善が見られます。
- **スタートアップファイルのチェック** - ウイルス定義データベースのアップデートの正常完了後、メモリーと実行中のサービスを検査します。
- **毎週の検査** - 全ファイルシステムを毎週検査します(既定では月曜日の午前2時)。このタスクはユーザーがカスタマイズできます。
- **定期的に自動アップデート** - コンピューターの最大限のセキュリティを確保するためには、System Center Endpoint Protectionを定期的にアップデートするのが最善の方法です。詳細は、SCEPアップデートユーティリティーを参照してください。
- **脅威の通知** - 既定では、各脅威はsyslogにログ記録されます。それ以外に、外部(通知)スクリプトが実行されるようにSCEPを設定して、脅威が検出されたことを電子メールでシステム管理者に通知することができます。

統計

すべてのアクティブなSCEPエージェントの統計をここで見ることができます。[**統計**]の概要は、10秒ごとに更新されます。

図6-6. [SCEP」 -[コントロール] > [統計]

ログ

SCEPには、syslogを介したシステムデーモンログが備わっています。*Syslog*は、プログラムメッセージのログ記録用の標準であり、ネットワークやセキュリティのイベントなどのシステムイベントをログ記録するのに使用できます。

機能を参照するメッセージ:

```
auth, authpriv, daemon, cron, ftp, lpr, kern, mail, ..., local0, ..., local7
```

メッセージには、その送信者によって優先順位とレベルが割り当てられます。

```
Error, Warning, Summall, Summ, Partall, Part, Info, Debug
```

ここでは、syslogのログ出力の設定および読み出しの方法を説明します。*'syslog_facility*オプション(既定値は*'daemon*)は、ログで使用するsyslog機能を定義します。syslogの設定を変更するには、SCEP設定ファイルを編集するか、またはWebインターフェイスを使用します。*'syslog_class'*パラメーターの値を変更して、ログクラスを変更します。これらの設定は、syslogに熟達している場合のみ変更することをお勧めします。syslog設定の例は、以下を参照してください。

```
syslog_facility = "daemon"
syslog_class = "error:warning:summall"
```

ログファイルの名前と場所は、syslogのインストール先と設定(例: rsyslog? syslog-ng、など)によって異なります。syslog出力ファイルの標準ファイル名は、例えば、*'syslog*? *'daemon.log'*、などです。syslogアクティビティーを追跡するには、コンソールから次のようなコマンドの１つを実行します。

```
tail -f /var/log/syslog
tail -100 /var/log/syslog | less
cat /var/log/syslog | grep scep | less
```

重要: System Center Operations Managerを使用したLinux SCEP製品の監視が正しく機能するには、まずSCEP構成ファイルまたはSCEP Webインターフェイスを介して有効化する必要があります。上述の設定ファイル中の*'scom_enabled'*パラメータが、*'scom_enabled = yes'*と設定されていることを確認するか、または[**設定**] > [**グローバル**] > [**デーモンオプション**] > [**SCOMは有効です**]の下のWebインターフェイスの該当する設定を変更します。

SCEPセキュリティシステムアップデート

SCEPアップデートユーティリティー

System Center Endpoint Protectionの機能を維持するには、ウイルス定義データベースを最新に保つ必要があります。*scep_update*ユーティリティーは、まさにその目的で開発されたものです。詳細は、*scep_update(8)*マニュアルページを参照してください。HTTPプロキシーを通してインターネットにアクセスするサーバーの場合、追加の設定オプション*'proxy_addr*? *'proxy_port*も定義する必要があります。HTTPプロキシーへのアクセスにユーザー名とパスワードが必要な場合、*'proxy_username*および*'proxy_password*オプションもこのセクションで定義する必要があります。アップデートを開始するには、次のコマンドを入力します。

@SBINDIR@/scep_update

エンドユーザーに対して最大限のセキュリティを確保するために、Microsoftチームは常に世界中からウイルス定義の収集に努めています。新しいパターンが非常に頻繁にウイルス定義データベースに追加されています。そのため、定期的にアップデートを起動することをお勧めします。アップデートの頻度を指定できるようにするには、SCEP設定ファイルの**[global]**セクションの*'scheduler_tasks*'オプションで、*'@update*'タスクを設定する必要があります。また、スケジューラを使用して、アップデート頻度を設定することもできます。ウイルス定義データベースを正常にアップデートするには、SCEPデーモンが稼動している必要があります。

SCEPのアップデートプロセスの説明

アップデートプロセスは、次の2つの段階で構成されます。まず、事前コンパイル済みのアップデートモジュールをMicrosoft Serverからダウンロードします。

アップデートプロセスの2番目の段階では、ローカルミラーに保管されているモジュールからSystem Center Endpoint Protectionスキャナーで読み込み可能なモジュールをコンパイルします。通常、次のようなSCEP読み込みモジュールが作成されます。ローダーモジュール(em000.dat)、スキャナモジュール (em001.dat)、ウイルス定義データベースモジュール (em002.dat)、アーカイブサポートモジュール (em003.dat)、アドバンスドヒューリスティックモジュール (em004.dat)、など。モジュールは、次のようなディレクトリ内に作成されます。

@BASEDIR@

ご意見をお聞かせください

本書は、System Center Endpoint Protectionのインストール、設定、および保守での必要事項をよくご理解いただくことためのものです。ただし、本書の目標は、資料の品質および有効性を改善し続けることにあります。本書のいずれかのセクションが不明瞭または不完全と思われた場合、以下のカスタマーサポートにご連絡ください。

support.microsoft.com

弊社は最大限のサポートを提供するよう努力しており、製品に関して何らかの問題が起きた場合はサポートを行います。

付録A. PHP ライセンス

The PHP License, version 3.01 Copyright (c) 1999 - 2006 The PHP Group. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, is permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. The name "PHP" must not be used to endorse or promote products derived from this software without prior written permission. For written permission, please contact group@php.net.
4. Products derived from this software may not be called "PHP", nor may "PHP" appear in their name, without prior written permission from group@php.net. You may indicate that your software works in conjunction with PHP by saying "Foo for PHP" instead of calling it "PHP Foo" or "phpfoo"
5. The PHP Group may publish revised and/or new versions of the license from time to time. Each version will be given a distinguishing version number. Once covered code has been published under a particular version of the license, you may always continue to use it under the terms of that version. You may also choose to use such covered code under the terms of any subsequent version of the license published by the PHP Group. No one other than the PHP Group has the right to modify the terms applicable to covered code created under this License.
6. Redistributions of any form whatsoever must retain the following acknowledgment: "This product includes PHP software, freely available from <http://www.php.net/software/>".

THIS SOFTWARE IS PROVIDED BY THE PHP DEVELOPMENT TEAM "AS IS" AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE PHP DEVELOPMENT TEAM OR ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.