

F-Secure Linux Protection

目次

1: はじめに.....	4
1.1 特長.....	5
1.2 主な機能.....	5
1.3 危険なコンテンツについて.....	6
1.3.1 ウイルス.....	6
1.3.2 不要と思われるアプリケーションと不要なアプリケーション.....	6
1.3.3 ワーム.....	7
1.3.4 トロイの木馬.....	7
1.3.5 バックドア.....	8
1.3.6 エクスプロイト.....	8
1.3.7 エクスプロイトキット.....	9
1.3.8 ルートキット.....	9
2: インストール.....	10
2.1 F-Secure Elements EndpointProtectionで使用するための製品をインストールする.....	11
2.1.1 製品のアクティベーションを遅らせる.....	12
2.2 製品をアンインストールする.....	12
3: 基本操作.....	14
3.1 マルウェア スキャン.....	15
3.1.1 リアルタイムスキャン.....	15
3.1.2 マニュアルスキャン.....	16
3.1.3 スケジュールスキャン.....	16
3.2 完全性検査.....	16
3.3 クライアントステータスの確認.....	18
3.4 ウイルス定義ファイルの自動更新.....	18
3.5 HTTPプロキシを使用する.....	19
3.6 例 完全性検査プロファイルの管理.....	20
3.7 F-Secure Elements Endpoint Protectionポータルでの基本.....	20
3.7.1 F-Secure Elements EndpointProtectionポータルでリアルタイムスキャンを使用する.....	21
3.7.2 F-Secure Elements EndpointProtectionポータルでの整合性チェックの使用.....	23
3.7.3 F-Secure Elements EndpointProtectionポータルを使用した自動更新オプションの設定.....	24
3.8 lsctlユーティリティを使用して設定を構成する.....	25
3.8.1 設定タイプと設定ツリー.....	26

3.8.2 設定の操作.....	27
3.8.3 入力および出力フォーマット.....	28
3.8.4 配列の操作.....	29
3.8.5 ファイルからの製品の設定をロードする.....	30
3.8.6 製品の状態を検査する.....	31
3.8.7 リアルタイムスキャンのコマンドライン設定.....	33
3.8.8 完全性検査のコマンドラインの設定.....	36
3.8.9 自動更新のコマンドライン設定.....	36
3.9 コマンドラインを使用する.....	38
3.9.1 サービスの開始と停止.....	38
3.9.2 コマンドラインからコンピュータを手動でスキャンする.....	38
3.9.3 コマンドラインから手動で製品を更新する.....	40
3.9.4 アンチウイルス保護の動作確認.....	41
3.9.5 コマンドラインから完全性検査を実行する.....	41

付録 A: 警告の深刻度.....42

付録 B: 製品と一緒にインストールされるサービス.....44

付録 C: クラウドサービス.....48

はじめに

トピック:

- ・ [特長](#)
- ・ [主な機能](#)
- ・ [危険なコンテンツについて](#)

本製品は、強力なリアルタイム保護機能(ウイルス・潜在的に望ましくないアプリケーションを対象)を備え、すぐに利用できる統合セキュリティソリューションとして提供します。また、不正なシステム改ざん、ユーザスペースおよびカーネルルートキットから保護するホスト型侵入防止システム(HIPS)機能も含まれています。

コンピュータウイルスは、コンピュータに保存されたデータを脅かす重大の脅威の1つです。一部には無害なウイルスもありますが、多くは実際に脅威でありデータを破壊します。

本ソリューションはF-Secure Elements Endpoint Protectionの管理ポータルを使用することで、展開と管理を行えます。ポリシーマネージャに接続せずにLinux Protectionを実行することもできます。その場合、コマンドラインツールを使用して製品の設定を管理できます。

1.1 特長

本製品は侵入を検知して防止する機能を備えており、コンピュータをマルウェア(悪意のあるソフトウェア)から保護します。

ユーザがインターネットからファイルをダウンロードすると(たとえばメールのリンクをクリックして)、そのファイルは開く前に自動的にスキャンされます。ファイルが感染している場合、製品がシステムをマルウェアから保護します。

- ・ 「リアルタイムスキャン」はファイルを開いたり、コピーしたり、インターネットからダウンロードしたりするときに実行されるウイルスと不要と思われるアプリケーションに対して継続的な保護を提供します。この機能は透過的に動作し、ハードディスク、リムーバブルメディアやネットワークドライブがアクセスされるたびにウイルスの検出を行います。感染したファイルにアクセスしようとする、リアルタイムスキャンは自動的にウイルスの実行を阻止します。
- ・ リアルタイムスキャンが特定のファイル/ディレクトリをスキャンするように設定されている場合、「マニュアルスキャン」を使用してシステム全体をスキャンすることができます。また、「スケジュールスキャン」を使用してシステム全体に定期的なスキャンを設定することも可能です。
- ・ 「自動更新」はウイルス定義ファイルとスパイウェアの定義ファイルを自動的に更新して、本製品の保護機能を常に最新の状態にします。本製品をインストールした後、この機能は有効になり、ウイルス定義ファイルは自動的に更新されるようになります。更新されるウイルス定義ファイルは F-Secure によって署名されています。

「ホスト型侵入防止システム (HIPS)」はホスト上の不審な動きを検知し、システムを保護します。

- ・ 「完全性検査」はシステムを不正な変更から保護します。この機能は既知の構成(正常で問題がないシステムの状態)に基づいて、システムの安全性を確認します。安全で問題がない既知の構成を記録するためには、ネットワークに接続していない状態で本製品をインストールすることを推奨します。保護したいファイルのベースラインを作成して、すべてのユーザに対してファイルの変更を阻止することができます。
- ・ 第三者がシステムに侵入してユーザアカウントを追加しようとする、ホスト型侵入防止システム (HIPS) は変更されたシステムファイルを検知し、管理者に警告を送ります。
- ・ 第三者がシステムに侵入して、各種ユーティリティを置き換えるユーザスペースのルートキットをインストールしようとする、ホスト型侵入防止システム (HIPS) は変更処理されたシステムファイルを検知し、管理者に警告を送ります。

1.2 主な機能

本製品はウイルスやワームなどマルウェアに対する強力な保護機能を備え、透過的に動作します。

本製品は、Linux に対応しているファイルシステムのファイルをスキャンします。

- ・ Linux 上の対応ファイルシステムに対するスキャン機能
- ・ 複数のスキャンエンジンを使用した高性能検出機能
- ・ 危険性のあるファイルを検出するヒューリスティックスキャンエンジン
- ・ 不要と思われるアプリケーションの検出と分類化
- ・ ユーザが保護をバイパスできないように設定可能
- ・ ファイルを開く/閉じる、実行する前に行われるウイルススキャン
- ・ スキャン対象(ファイル/ディレクトリ)、スキャン方法およびマルウェアを検出した場合の処理と通知方法
- ・ アーカイブファイルの再帰的スキャン

- ・ デジタル署名を利用したウイルス定義ファイルの更新
エンドユーザに対して透過的の動作
 - ・ ウイルス定義ファイルの自動更新
重大なシステムファイルの情報を保存し、アクセスされる前にファイルを自動で確認
 - ・ ファイルの無断変更(トロイの木馬からなど)を阻止
 - ・ システムファイルの変更を検出し、管理者に警告を通知
- ほとんどのシステムにはデフォルトの設定が適切です。
- ・ セキュリティポリシーを1つの場所から一元的に設定して配布する
感染ファイルの情報を管理者に通知するための豊富な監視と警告機能

1.3 危険なコンテンツについて

危険なアプリケーションとファイルはデータを破壊したり、コンピュータへの無断なアクセスを入手して個人情報を盗み取ろうとします。

1.3.1 ウイルス

「ウイルス」は自らファイルやディスクに付加し、継続的に自己増殖する寄生型のプログラムです。データを変更して、コンピュータを破壊する危険性をもっています。

ウイルスはほとんどの場合ユーザの知らないうちに感染します。一旦システムに侵入すると、ウイルスは増殖を試みます。ウイルスには次の特徴があります。

- ・ システムのリソースを悪用する
- ・ コンピュータのファイルを変更または破壊したりする
- ・ 侵入したコンピュータを利用して他のコンピュータに感染を移そうとする
- ・ コンピュータを不正な目的で利用する

1.3.2 不要と思われるアプリケーションと不要なアプリケーション

「不要と思われるアプリケーション」には、不快な、または望ましくないと思われる動作や特性があります。「不要なアプリケーション」には、デバイスやデータに深刻な影響を与える動作や特性があります。

次の条件がある場合、アプリケーションは不要である可能性があります。

- ・ **プライバシーや生産性に影響を与えます** -たとえば、個人情報の漏洩や、不正な操作を行います。
- ・ **デバイスのリソースに過度の負担をかけます** -たとえば、過剰にストレージやメモリの容量を使用します。
- ・ **デバイスのセキュリティやそのデバイスに保存されている情報を侵害します** -たとえば、予期しないコンテンツやアプリケーションにさらされます。

これらの動作や特性がデバイスやデータに与える影響は、軽いものから重大なものまでさまざまです。しかし、このアプリケーションをマルウェアとして分類するほど有害なわけではありません。

アプリケーションに、重大な影響を与える動作または特性がある場合、そのアプリケーションは「不要なアプリケーション」とみなされます。このようなアプリケーションはより注意深く扱われます。

本製品は、PUAかUAかによってアプリケーションを異なる方法で処理します。

- ・ **不要と思われるアプリケーション** -製品がアプリケーションの実行を自動的にブロックします。アプリケーションを確実に信頼できる場合、スキャンから除外するようにF-Secure製品を設定できます。ブロックされたファイルをスキャンから除外するには管理者権限が必要です。

- ・ **不要なアプリケーション** - 製品がアプリケーションの実行を自動的にブロックします。

1.3.3 ワーム

「ワーム」は、ネットワーク上にあるデバイスから別のデバイスに、自分自身のコピーを送信するプログラムです。一部のワームは、影響を受けたデバイス上で有害な動作も実行します。

多くのワームは、ユーザに魅力的に見えるように設計されています。画像、動画、アプリケーション、その他の有用なプログラムやファイルのように思うかもしれません。この偽装の目的は、ユーザを引き付け、ワームをインストールさせることです。他のワームは完全なステルス設計で、ユーザに気付かれることすらなく、ワーム自体をインストールするデバイス(またはそれにインストールされたプログラム)の脆弱性を悪用できます。

ワームは、一度インストールされると、デバイスの物理リソースを使用して自身のコピーを作成し、それらのコピーをネットワーク経由で届く範囲の他のデバイスに送信します。大量のワームのコピーが送信されると、デバイスのパフォーマンスが低下する可能性があります。ネットワーク上の多くのデバイスが影響を受け、ワームのコピーを送信すると、ネットワーク自体が混乱する可能性があります。一部のワームは、影響を受けたデバイスに保存されているファイルを変更したり、他の有害なアプリケーションをインストールしたり、データを盗むなど、直接害を与えることもできます。

ほとんどのワームは、一種類のネットワークにのみ感染します。比較的まれですが、2種類以上のネットワークに拡散できるものもあります。通常、ワームは、次のネットワークに拡散しようと試みます(これ以外にアクセスが低いものを標的にするものもあります)。

- ・ ローカル ネットワーク
- ・ メール ネットワーク
- ・ ソーシャルメディア サイト
- ・ Peer-to-peer (P2P) 接続
- ・ SMS/MMS メッセージ

1.3.4 トロイの木馬

「トロイの木馬」は、魅力的な機能や特徴を提供している、あるいは提供していると見せかけるプログラムですが、バックグラウンドで静かに有害な動作を行います。

ギリシャの伝説のトロイの木馬にちなんで名付けられたトロイの木馬は、ユーザに魅力的に見えるように設計されています。ゲーム、スクリーンセーバー、アプリケーションのアップデート、その他の有用なプログラムやファイルのように見えるかもしれません。一部のトロイの木馬は、人気のあるプログラムや有名なプログラムを模倣あるいは完全にコピーし、より信頼性を高く見せています。この偽装の目的は、ユーザがトロイの木馬をインストールするよう誘導することです。

インストールされると、トロイの木馬は「罫」を使用し、正当であるという錯覚を維持することもできます。たとえば、スクリーンセーバーアプリケーションや文書ファイルに偽装されたトロイの木馬は、画像または文書を表示します。ユーザがこれらの罫に気を取られている時に、トロイの木馬は、バックグラウンドで他の動作を静かに実行します。

トロイの木馬は、通常、デバイスに有害な変更(ファイルの削除や暗号化、プログラム設定の変更など)を行ったり、そこに保存されている秘密データを盗み出したりします。トロイの木馬は、実行する動作によって区別できます。

- ・ **Trojan-downloader (ダウンローダー型トロイの木馬)**: リモートサイトに接続して他のプログラムをダウンロードしてインストールします。
- ・ **Trojan-dropper (埋め込み型トロイの木馬)**: 1つまたは複数の追加プログラムが含まれており、それをインストールします。
- ・ **Trojan-pws (パスワード窃盗型トロイの木馬)**: デバイスに保存されたパスワードや Web ブラウザに入力されたパスワードを盗み出します。

- ・ **Banking-trojan (バンキング型トロイの木馬)**: オンラインバンキング ポータルのユーザ名とパスワードを特定する特殊なトロイの木馬です。
- ・ **Trojan-spy (スパイ型トロイの木馬)**: デバイスのアクティビティを監視し、詳細情報をリモートサイトに転送します。

1.3.5 バックドア

「バックドア」は、対象となるプログラム、デバイス、ポータルまたはサービスのセキュリティ機能を回避するために使用できる機能または特別に細工されたプログラムです。通常、アタッカーが不正なアクセスを取得したり、有害なアクションを実行するために使用されます。

プログラム、デバイス、ポータル、またはサービスの機能は、その設計や実装がセキュリティリスクをもたらす場合、バックドアと見なすことができます。たとえば、パスワードがハードコードされ、オンラインポータルの管理者しか知らない秘密のアクセスポイントは、バックドアと見なすことができます。

バックドアとして特別に作られたプログラムは、通常、対象プログラム、デバイス、ポータル、またはサービスのコードの欠陥を利用します。欠陥は、バグ、脆弱性、または文書化されていない機能である可能性があります。

バックドアは、通常、アタッカーが不正アクセスを取得したり、アクセス制限、認証、暗号化などのセキュリティ機能を回避するための有害なアクションを実行するために使用されます。

1.3.6 エクスプロイト

「エクスプロイト」(脆弱性を利用したソースコード)とは、プログラムの欠陥を利用して予期せぬ動作を実行するオブジェクトまたはメソッドであり、アタッカーが有害な行為を行える条件を生み出します。

エクスプロイトは、オブジェクトまたはメソッドのいずれかになります。たとえば、巧妙に細工されたプログラム、コードや文字列はすべてオブジェクトです。コマンドの特定のシーケンスがメソッドです。

エクスプロイトは、プログラムの欠陥または抜け穴(脆弱性とも呼ばれます)を悪用するために使用されます。すべてのプログラムが異なるため、各エクスプロイトはその特定のプログラムに合わせて慎重に調整する必要があります。

アタッカーがコンピュータやデバイスに影響を与えることができるように、アタッカーが脆弱性を利用したソースコードを送り込むいくつかの方法があります。

- ・ **ハッキングされた、または巧妙に細工されたプログラムに埋め込む**-プログラムをインストールして起動すると、脆弱性を利用した攻撃が開始されます。
- ・ **メールに添付された文書ファイルに埋め込む**-添付ファイルを開くと、攻撃が開始されます。
- ・ **ハッキングされた Web サイトや有害な Web サイトに忍ばせる**-サイトにアクセスすると、その脆弱性と利用した攻撃が開始されます。

エクスプロイトを起動すると、強制的にクラッシュしたり、システムのストレージやメモリを改ざんしたりするなど、予期しない動作が発生します。これにより、アタッカーがデータを盗んだり、OSの制限された部分にアクセスするなど、他の有害な措置を実行できるような条件が生じる可能性があります。

1.3.7 エクスプロイトキット

「エクスプロイトキット」は脆弱性を管理して、脆弱性のあるコンピュータまたはデバイスに危険なプログラムを送り込むためのツールキットです。

エクスプロイトキットには、エクスプロイトが複数含まれおり、それぞれが、プログラム、コンピュータ、またはデバイスの欠陥(脆弱性)を悪用します。キット自体は、通常、有害なサイトやハッキングされたサイトで配置されているため、サイトを訪れるコンピュータやデバイスがその影響を受けることがあります。

新しいコンピュータやデバイスが仕掛けられたサイトに接続すると、エクスプロイトキットは、キット内のエクスプロイトの攻撃から影響を受ける可能性のある脆弱性を探索します。検出された場合、キットはその脆弱性を利用するためにエクスプロイトを起動します。

コンピュータやデバイスに侵入した後、エクスプロイトキットはペイロードをそのコンピュータに送り込むことができます。これは通常、コンピュータまたはデバイスにインストールされて起動される別の有害なプログラムで、次々に他の不正な操作を実行します。

エクスプロイトキットは、モジュールとして設計され使いやすいため、不正操作者はツールキットにエクスプロイトやペイロードを簡単に追加・削除できます。

1.3.8 ルートキット

「ルートキット」はマルウェアの検出を困難にするプログラムです。

ルートキットはシステムに侵入したあとに第三者によって使用される一連のプログラムです。通常、ルートキットはユーザに気づかれないように侵入し、検出および削除されないように実行中のプロセスやファイルとデータをOSから隠ぺいします。ルートキットはほとんどの場合、悪質な目的に利用されません。

ユーザスペースのルートキットに対する保護

第三者がシステムのアクセスを得て、システムの各種ユーティリティを置き換えてユーザスペースのルートキットをインストールしようとする時、ホスト型侵入防止システム(HIPS)が変更処理されたシステムファイルを検知し、管理者に通知します。

第 2 章

インストール

トピック:

ここでは、製品をインストールする方法について説明します。

- [F-Secure Elements EndpointProtection](#)で使用するための製品をインストールする
- [製品をアンインストールする](#)


2.1 F-Secure Elements Endpoint Protectionで使用するための製品をインストールする

製品をF-Secure Elements Endpoint Protection管理モードでインストールすると、F-Secure Elements Endpoint Protectionポータルを使用して製品のインストールを一元管理できます。

インストールプロセスは、製品のインストールとアクティベーションの2つのステップで構成されています。

製品は3つの異なるインストールパッケージ形式で配布されます。Linuxディストリビューションに基づいて、正しいパッケージ形式を選択してください。

- DEBパッケージは、DebianおよびUbuntuシステムと互換性があります。
- RPMパッケージは、AlmaLinux、Amazon Linux、CentOS、Oracle Linux、Red Hat Enterprise Linux、およびSUSE Linux Enterprise Serverシステムと互換性があります。
- F-Secureインストーラを使用すると、サポートされているシステムに製品を自動的にインストールしてアクティブにできます。

 **注:** F-Secure Elements Endpoint Detection and Responseのセンサーに互換性がないため、次のLinuxディストリビューションはサポートされていません。


- SUSE Linux Enterprise Server 12 (SP5)
- SUSE Linux Enterprise Server 15
- UEKカーネルを搭載したOracleLinux

1. 製品を各ホストにインストールします

a) `root`としてホストにログインします。

b) 必要な dependencies (依存関係) がインストールされていることを確認します。


- Amazon 2、CentOS 7、Oracle Linux 7、RHEL 7 `fuse-libs`、`libcurl`、`python`
- AlmaLinux 8、CentOS 8、CentOS Stream 8、Oracle Linux 8、RHEL 8 `fuse-libs`、`libcurl`、`python36`
- Debian 9、Ubuntu 16.04 `libfuse2`、`libcurl3`、`python`
- Debian 10、Ubuntu 18.04 `libfuse2`、`libcurl4`、`python`
- Debian 11、Ubuntu 20.04 `libfuse2`、`libcurl4`、`python3`
- SUSE Linux Enterprise Server 12 `libfuse2`、`libcurl4`、`python3`

 **注:** F-Secure Elements Endpoint Detection and Responseには、`auditd`サービスがシステムにインストールされ、実行されている必要があります。

c) インストールコマンドを実行します。

- RPMベースのディストリビューション `rpm -Uvh f-secure-linuxsecurity.rpm`
- DEBベースのディストリビューション `dpkg -i f-secure-linuxsecurity.deb`
- F-Secureインストーラの使用 `./f-secure-linuxsecurity-installer --subscription-key SUBSCRIPTION-KEY --profile-id PROFILE-ID`

プロファイルIDの追加はオプションですが、追加することにより、インストールは自動的に正しいプロファイルに関連付けられます。ポータルのプロファイルエディタビューにある数値のプロファイルIDを使用します。

 **注:** F-Secure インストーラを使用する場合は、インストールパッケージの名前を変更してサブスクリプションキーとプロファイルIDを提供することもできます 例 `f-secure-linuxsecurity-installer-[SUBSCRIPTION-KEY]-[profile=ID]`、実際のサブスクリプションは [] で囲み、プロファイルIDを `profile=` の後に指定します。

名前を変更したインストールパッケージでは、追加のコマンドライン引数なしでインストールパッケージを実行できます。

2. 次のコマンドを実行して、製品をアクティブにします。

注: F-Secureインストーラを使用する場合、この手順は必要ありません。



```
/opt/f-secure/linuxsecurity/bin/activate --psb --subscription-key
SUBSCRIPTION-KEY
```

注: アクティベーション処理中に HTTP プロキシを使用するには、環境変数



FSECURE_HTTP_PROXY_HOST および FSECURE_HTTP_PROXY_PORT を指定してください。これらの変数が設定されると、アクティベーションツールは、操作中にこのプロキシを使用するように製品を設定します。本製品は、HTTP プロキシの基本認証をサポートしています。指定したプロキシが認証を必要とする場合は、環境変数 FSECURE_HTTP_PROXY_USER および FSECURE_HTTP_PROXY_PASS を使用して、プロキシの認証情報を指定してください。

関連タスク

[HTTPプロキシを使用する](#) ページ 19

本製品には単一のグローバルHTTPプロキシ設定があり、本製品が外部サービスへのHTTPリクエストを行う際に常に使用されます。

2.1.1 製品のアクティベーションを遅らせる

インストールまたは展開中に製品をすぐにライセンス認証する代わりに、次にコンピュータの電源を入れたときにライセンス認証が行われるように設定できます。

たとえば、複数の仮想マシンインスタンスに使用される仮想マシンテンプレート内に製品をインストールする場合は、アクティベーションを遅らせることができます。このようなセットアップでは、テンプレートに製品をインストールし、次に仮想マシンを起動するときにアクティベーションをスケジュールできます。このアプローチにより、個々の仮想マシンを使用時に個別にアクティベーションを行うことができます。

次回コンピュータを起動したときにアクティブ化が行われるように設定するには、次の手順に従います。

- .rpm または .deb インストールパッケージをインストールしている場合、activate コマンドに --next-boot 引数を追加します。

```
/opt/f-secure/linuxsecurity/bin/activate --next-boot --psb --subscription-key SUBSCRIPTION-KEY
```

- F-Secure インストーラを使用している場合は、--next-boot 引数をインストーラーコマンドに追加します。例

```
./f-secure-linuxsecurity-installer --next-boot --subscription-key SUBSCRIPTION-KEY
```

これにより、コンピュータが次に起動したときに Linux Protection がアクティブになるように設定されます。

2.2 製品をアンインストールする

本製品をコマンドラインからアンインストールすることができます。

1. root としてホストにログインします。
2. 次のアンインストールコマンドを実行します。

- RHEL-based distributions: `rpm -e f-secure-linuxsecurity`
- Debian ベースのディストリビューション: `dpkg -r f-secure-linuxsecurity`

上記のアンインストール コマンドが失敗し (アンインストールが失敗)、`rm -rf /opt/f-secure /etc/opt/f-secure /var/opt/f-secure` コマンドを実行してください。このコマンドは、残っているファイルや `systemd` サービス (あれば) を削除します。最後に、システムを再起動して、停止しているすべてのサービスとプロセスが終了することを確認します。

`uninstall` コマンドは、製品が作成した設定ファイルまたはログファイルを削除しません。残っているファイルがないか製品ディレクトリを確認し、不要である場合にはファイルを削除します。

基本操作

トピック：

- [マルウェア スキャン](#)
- [完全性検査](#)
- [クライアントステータスの確認](#)
- [ウイルス定義ファイルの自動更新](#)
- [HTTPプロキシを使用する](#)
- [例 完全性検査プロファイルの管理](#)
- [F-Secure Elements Endpoint Protectionポータルを使い方の基本](#)
- [lsctlユーティリティを使用して設定を構成する](#)
- [コマンドラインを使用する](#)

製品の設定は、`lsctl`コマンドラインツールを使用して、または(集中管理コンピュータの場合)F-Secure Elements EPP ポータルを使用して構成できます。

3.1 マルウェア スキャン

本製品はコンピュータをファイルの破壊、個人情報の搾取、不正な目的などから保護します。

この製品は次のことも行います。

- ・ バックグラウンドでのリアルタイムスキャンと、マニュアルスキャンによるオンデマンドスキャンの両方で、マルウェアからコンピュータを保護することができます。
- ・ 指定したファイルやディレクトリ、リムーバブルメディア ポータブルドライブなど、ダウンロードしたコンテンツなどを自動的にスキャンすることができます。本製品は、マルウェアを示す可能性のあるあらゆる変化について、コンピュータを監視します。

リアルタイムスキャンはデフォルトで有効になっていますが、スキャンを実行する前に、スキャンに含めるファイルとディレクトリのリストをElements Endpoint Protection EPP ポータルに追加する必要があります。

関連概念

[リアルタイムスキャン](#) ページ15

「リアルタイムスキャン」はファイルを開く際に自動でスキャンが実行される継続的なウイルス保護です。リアルタイムスキャンがマルウェアを検出した場合、ファイルのアクセスは自動的にブロックされます。

[マニュアルスキャン](#) ページ16


この製品は、コンピューター全体をスキャンして、有害なファイルや不要なアプリケーションがないことを確認できます。

[スケジュールスキャン](#) ページ16

スケジュールスキャンは、コンピューターが使用されていないときにコンピューターをスキャンし、コンピューターがクリーンであることを定期的に確認するのに役立ちます。

3.1.1 リアルタイムスキャン

「リアルタイムスキャン」はファイルを開く際に自動でスキャンが実行される継続的なウイルス保護です。リアルタイムスキャンがマルウェアを検出した場合、ファイルのアクセスは自動的にブロックされます。

 **注:** リアルタイムでスキャンする必要がある個々のファイルとディレクトリは、Elements Endpoint Protection EPP ポータルで指定する必要があります。

リアルタイムスキャンは次のように動作します。

1. コンピュータ上でファイルがアクセスされます。
2. コンピュータがファイルにアクセスできる前に、ファイルに対してマルウェアのスキャンがすぐに行われます。
3. マルウェアを検出した場合、マルウェアがシステムに蔓延できないようにファイルのアクセスがブロックされます。
4. 設定に基づいて、リアルタイムスキャンは感染ファイルを処理します (名前の変更または削除)。

スキャンしたファイルは、クリーン(安全)、マルウェア、不要と思われるアプリケーション、または不審として分類されます。クリーンファイルはリアルタイムスキャンの影響を受けません。感染したファイルは開くことが阻止されます。

リアルタイムスキャンがファイルのスキャンにかかる時間と使用するシステムリソースはファイルのコンテンツ、場所、およびタイプによって異なります。

次のようなファイルはスキャンが通常より長くかかります。

- ・ ZIP形式などの圧縮ファイル (デフォルトでは圧縮ファイルはスキャン対象外)

- ・ ネットワーク上のファイル
- ・ 大きなファイルが影響を受ける可能性があります。

リアルタイムスキャンが多くのファイルを同時に処理している場合、システムの動作は低下することがあります。

関連タスク

[F-Secure Elements Endpoint Protectionポータルでリアルタイムスキャンを使用する](#) ページ21

リアルタイムスキャンは、ファイルにアクセスした際にスキャンを行い、マルウェアが含まれるファイルへのアクセスをブロックします。

3.1.2 マニュアルスキャン

この製品は、コンピューター全体をスキャンして、有害なファイルや不要なアプリケーションがないことを確認できます。

コンピューター全体のスキャンは完了するまでに長い時間がかかる可能性があるため、ファイルとディレクトリのセットを制限することで、これにかかる時間は短くなります。他のすべてのファイルの場所とディレクトリは、Elements Endpoint Protection EPP ポータルを介してスキャンから除外できます。

手動スキャンをよりきめ細かく制御するために、`fsanalyze`コマンドを使用できます。

関連タスク

[マニュアルスキャンを設定する](#) ページ22

手動スキャンの除外とアクションをで設定できます F-Secure Elements Endpoint Protectionポータル。

[コマンドラインからコンピューターを手動でスキャンする](#) ページ38

製品がインストールされているLinuxホストのコマンドラインから手動でコンピューターに対するマルウェアスキャンを行えます。

3.1.3 スケジュールスキャン

スケジュールスキャンは、コンピューターが使用されていないときにコンピューターをスキャンし、コンピューターがクリーンであることを定期的に確認するのに役立ちます。

特定のファイルやディレクトリ、またはリムーバブルメディアのスキャンは、特定の時間帯および特定の曜日に実行されるようにスケジュールできます。

関連タスク

[スケジュールスキャンの作成](#) ページ23

スケジュールスキャンを使用して、管理対象コンピューターに対してマルウェアのスキャンを定期的に行うことができます。


3.2 完全性検査

「完全性検査」は、重要なファイルを不正な変更処理から保護します。

保護しているファイルのパーミッション(アクセス権)がどのように指定されていても、保護しているファイルに対する変更を検出・ブロックするように設定できます。

完全製チェックを使用するには、保護するファイルをベースラインのリストに追加する必要があります。このリストのファイルは、不正な変更から保護されます。

完全性検査はディスク上のファイルを「ベースライン」の属性と比較します。ベースラインは暗号化および電子署名されたファイルのプロパティを表します。監視対象のファイルに対して変更処理があった場合、管理者へ通知を送信します。

 **注:** 完全性検査のいくつかの側面は、使用中のファイルシステムの種類に依存します。完全にサポートされているファイルシステムには、Ext4、ZFS、BTRFS、NFS、CIFSなどがあります。完全性検査は、最新のほとんどのファイルシステムで利用可能なファイル属性を利用しますが、すべて利用するわけではありません。

次の表では、ベースラインに含まれるファイルについて、完全性検査がどのようにさまざまなイベントを処理するかを詳細に説明します。

イベント	完全性検査のオプション
信頼できないプロセスがファイルを変更する	<p>読み取りおよび書き込みアクション設定で構成されます。</p> <ul style="list-style-type: none"> 書き込みアクションがdenyに設定されている場合、変更の試みはブロックされ、アラートは送信されません。 書き込みアクションがallowに設定されている場合、完全性検査は変更のアラートを管理者に送信します。 読み取りアクションがdenyに設定されている場合、完全性検査は、変更後のファイルを開こうとする試みをブロックします。
信頼できるプロセスがファイルを変更する	<p>プロセスのプログラムファイルがベースラインに含まれており、改ざんされていない場合は、信頼されたプロセスとみなされます。このような場合、完全性検査は何も行いません。</p>
ファイルを削除する	<p>完全性検査は、イベントに対するアラートの発生を防ぐことはできません。また、アラームも送信しません。</p>
ファイルの名前を変更する	<p>完全性検査は、イベントに対するアラートの発生を防ぐことはできません。また、アラームも送信しませんが、名前が変更されたファイルの監視は続行されます。</p>
ファイルへのハードリンクまたはソフトリンクを作成する	<p>完全性検査はリンクに従います。</p>
ファイルを置き換える	<p>完全性検査は、イベントに対するアラートの発生を防ぐことはできません。また、アラームも送信しません。</p>
ファイルのアクセス許可を変更する	<p>変更はブロックされませんが、読み取りおよび書き込みアクションの設定によって、ファイルを開く後続の試行がブロックされるか、アラートが表示されるかが決まります。</p>

イベント	完全性検査のオプション
ファイル所有者を変更する	変更はブロックされませんが、読み取りおよび書き込みアクションの設定によって、ファイルを開く後続の試行がブロックされるか、アラートが表示されるかが決まります。
ファイル属性を変更する	これにはSELinuxラベルが含まれます。完全性検査は、イベントに対するアラートの発生を防ぐことはできません。また、アラームも送信しません。

注: 完全性検査が送信するアラートについて



- tamperingアラートは、プロセスがすでに改ざんされているファイルにアクセスしていることを意味します。
- tampering-actionアラートは、プロセスがファイルを改ざんしていることを意味します。これらのアラートは、書き込みアクションがallowに設定されている場合にのみ送信されます。
- 書き込みアクションがallowに設定されている場合、読み取りアクションの設定に関係なく、すべてのアラートが送信されます。

3.3 クライアントステータスの確認

NS[**完全なステータス更新を送信する**]要求は、クライアントのインストールのステータスを確認するためのF-Secure Elements Endpoint ProtectionポータルEPPを介したオプションです。

を選択することにより[**完全なステータス更新を送信する**]のオプション**端末**ページでは、クライアントはバックエンドと同期し、現在のステータスに関する利用可能な情報でポータルを更新します。

リクエストは、利用可能な更新を同時にチェックします。クライアントが更新をすぐにインストールするように構成されている場合、クライアントは利用可能な更新もインストールします。

要するに、[**完全なステータス更新を送信する**]リクエストにより、次のことが可能になります。

- クライアントのステータス情報をチェックし、
- 利用可能なアップデート製品とエンジンをチェックし、
- クライアント側で構成されている場合は、利用可能な更新をインストールします。

事実上、これは定期的に更新をチェックする方法です。

3.4 ウイルス定義ファイルの自動更新

自動更新機能は、管理対象ネットワーク内のLinuxホストに対するセキュリティ保護を維持します。

自動更新機能を使用することで、管理対象ホストはポリシー設定に基づいて最新のアップデートを取得するようになります。デフォルトでは、自動更新は有効になっています。新しいウイルス定義は1日に数回公開可されるため、この機能を有効にしておくことを強く推奨します。

F-Secureは、バックエンドシステムとインストールされた製品間の専用サービス接続を「チャンネル」と呼び、マルウェア定義データベース、スキャンエンジン、および実際の製品へのアップデートを配信します。各チャンネルは、特定の製品コンポーネントまたは機能のアップデートを提供します。

自動更新の設定は、製品が使用する3つのF-Secure製品チャンネル linuxsecurity、fsbg、および baseguard に適用されます。自動更新の設定は、スキャンエンジンなどの他のチャンネルのアップデートが利用可能になるとすぐに適用されるため、影響はありません。自動更新の設定は、スキャンエンジンなどの他のチャンネルの更新には影響しません。自動更新をオフにすると、すべてのチャンネルの更新が無効になります。

また、製品チャンネルのアップデートは、アップデートが公開されてから1週間後に適用されます。

ウイルス定義ファイルの最新情報は次のURLでご覧になれます。 <https://dbtracker.f-secure.com/>

3.5 HTTPプロキシを使用する

本製品には単一のグローバルHTTPプロキシ設定があり、本製品が外部サービスへのHTTPリクエストを行う際に常に使用されます。


グローバルHTTPプロキシ設定は、製品を起動する前に環境変数を使用するか、製品を起動した後に `lsctl` ユーティリティを使用して設定することができます。

環境変数を使用してプロキシ設定を定義した場合、製品はローカル設定のオーバーライドとして設定を保存します。製品の有効化が完了したら、次のコマンドで設定を表示することができます `lsctl get http_proxy`

ローカルオーバーライドは、リモート管理サービス F-Secure Elements Endpoint Protection またはポリシーマネージャ から取得される構成よりも優先されます。プロキシ設定をリモートで設定するには、管理サービスで設定をロック済みとして設定する必要があります。設定がロックされている場合、ローカル構成が存在する場合でも、管理サービスから配布されるプロファイルまたはポリシーに設定されている値が常に使用されます。

グローバルHTTPプロキシを指定すると、本製品はElements Endpoint Protectionとの通信にプロキシを使用します。プロキシの設定は、Elements Endpoint Protectionのポータルでも変更できます。

1. HTTPプロキシを設定します。

 **注:** 本製品は、HTTPプロキシの基本認証をサポートしています。指定したプロキシが認証を必要とする場合は、環境変数 `FSECURE_HTTP_PROXY_USER` および `FSECURE_HTTP_PROXY_PASS` を使用して、プロキシの認証情報を指定してください。

- 製品をアクティベートする前にプロキシを設定するには、以下のコマンドを実行して必要な環境変数をエクスポートし、アクティベーションまたはインストーラーコマンドを実行した際に使用できるようにします。

```
export FSECURE_HTTP_PROXY_HOST=<host address>
export FSECURE_HTTP_PROXY_PORT=<host port number>
```

- アクティベーションの後にプロキシを設定するには、以下の `lsctl` コマンドを実行します。

```
/opt/f-secure/linuxsecurity/bin/lsctl set http_proxy host <host address>
/opt/f-secure/linuxsecurity/bin/lsctl set http_proxy port <host port number>
/opt/f-secure/linuxsecurity/bin/lsctl set http_proxy enabled true
```

2. 次のコマンドを実行して、プロキシ設定を確認します。

```
/opt/f-secure/linuxsecurity/bin/lsctl get http_proxy
```

これにより、現在の設定が返されます。

```
{
  "enabled": true,
  "host": <host address>,
}
```

```
"port": <host port number>
}
```

関連タスク

[F-Secure Elements EndpointProtectionで使用するための製品をインストールするページ](#)11

製品をF-Secure Elements Endpoint Protection管理モードでインストールすると、F-Secure Elements Endpoint Protectionポータルを使用して製品のインストールを一元管理できます。

3.6 例 完全性検査プロフィールの管理

lsctlを使用できる一般的なタスクのこの例は、ic profiles配列を介して完全性検査プロフィールを変更する方法を説明します。

次の手順は、完全性検査プロフィールを追加して、/opt ディレクトリ内のファイルへの書き込みを防ぐ方法の概要を示しています。

1. 次のコマンドを実行します。

```
/opt/f-secure/linuxsecurity/bin/lsctl set ic enabled yes
/opt/f-secure/linuxsecurity/bin/lsctl add --file - ic profiles <<EOF
{
  "path": "/opt",
  "verify_attributes": {
    "mode": false,
    "user": false,
    "group": false,
    "size": false,
    "mtime": false
  },
  "read_action": "allow",
  "write_action": "deny"
}
EOF
```

この例では、複数行入力を入力するためにシェルの文書構文を使用しています。

2. プロフィールを編集するには、次のコマンドを実行します。

```
/opt/f-secure/linuxsecurity/bin/lsctl set --prompt ic profiles /opt
```

これにより、/optプロフィールの現在の値をテンプレートとして指定したテキストエディタが開きます。

3. すべての完全性検査を削除するには、以下のコマンドを実行してic profilesを空の配列に設定します。

```
/opt/f-secure/linuxsecurity/bin/lsctl set ic profiles '[]'
```

3.7 F-Secure Elements Endpoint Protectionポータルの使い方の基本

F-Secure Elements Endpoint Protectionポータルは、F-Secure 製品の設定変更と統計情報を表示するために使用されます。

「[プロファイル > Linux Protection](#)」ページにある設定を使用して、本製品を構成してください。

F-Secure Elements Endpoint Protectionポータルの詳細については、[F-Secure Elements Endpoint Protection管理者ガイド](#)を参照してください。

3.7.1 F-Secure Elements Endpoint Protectionポータルでリアルタイムスキャンを使用する

リアルタイムスキャンは、ファイルにアクセスした際にスキャンを行い、マルウェアが含まれるファイルへのアクセスをブロックします。

1. Elements Endpoint Protectionポータルで**プロファイル** > **Linux Protection** ページに移動します。
2. 編集するプロファイルの名前をクリックします。
3. [リアルタイムスキャン] ページで、[リアルタイムスキャン] をオンにします。
4. [スキャンするファイルとフォルダ] で、[パスを追加] をクリックしてスキャンするパスを入力します。

個々のファイルまたはディレクトリには絶対パス名を使用してください。リストされているディレクトリにはサブディレクトリも再帰的に含まれています。ワイルドカードは使用できません。また、動的リンクは追跡されません。

注: デフォルトでは、このフィールドは空です (何もスキャンされません)。



5. [スキャンから除外されたファイルとフォルダ] で、[除外するパスを追加] をクリックして、スキャンから除外するファイルとフォルダを入力します。
ファイルとフォルダの除外には、ファイルまたはディレクトリの完全な絶対パス名も必要です。ワイルドカードは使用できません。
6. 実行可能フラグがオンになっているファイルのみをスキャンする場合、[実行可能ファイルのみスキャン] をオンにします。
7. 「リアルタイムスキャンの処理」で、[マルウェアの処理] と [不審なファイルの処理] を設定します。
 - ・ **名前の変更** 感染したファイルの名前を変更します。名前が変更されたファイルには、検出のタイプに応じて、.malware または .suspected という拡張子が付いています。
 - ・ **削除** 感染したファイルを削除します。
 - ・ **何もしない** 感染したファイルに対してアクションを実行しません。
8. [保存して発行] をクリックします。
9. [OK] をクリックします。

不要と思われるアプリケーションのスキャン

リアルタイムスキャンとマニュアルスキャンを設定して、マルウェアに加えて不要と思われるアプリケーション(PUA: potentially unwanted application)を処理できます。


1. Elements Endpoint Protectionポータルで**プロファイル** > **Linux Protection** ページに移動します。
2. 編集するプロファイルの名前をクリックします。
3. [リアルタイムスキャン] ページで、[リアルタイムスキャン] がオンになっていることを確認します。
4. [リアルタイムスキャン] と [マニュアルスキャンの両方のページで、[不要と思われるアプリケーションをスキャンする] をオンにします。
5. 「リアルタイムスキャンのアクション」と「マニュアルスキャンのアクション」で [不要と思われるアプリケーションのアクション] を設定します。
 - ・ **Rename** 感染したファイルの名前を変更します。名前が変更されたファイルには、.pua 拡張子が付きます。
 - ・ **削除** 感染したファイルを削除します。
 - ・ **何もしない** 感染したファイルに対してアクションを実行しません。
6. [保存して発行] をクリックします。

7. [OK]をクリックします。


アーカイブファイルをスキャンする

圧縮アーカイブファイルをチェックするために、リアルタイムスキャンとマニュアルスキャンを設定できます。


アーカイブファイルのスキャンを有効にすると、ZIP、ARJ、LZH、RAR、CAB、TAR、BZ2、GZ、JAR、TGZ形式のファイル内をスキャンすることが可能になります。

 **注:** アーカイブファイルのスキャンはファイルのコンテンツを一時的に解凍する場合があります。一時ファイルが必要とするディスク容量はアーカイブファイルのコンテンツによって異なります。

1. Elements Endpoint Protectionポータルで**プロファイル** > **Linux Protection**ページに移動します。
2. 編集するプロファイルの名前をクリックします。
3. [リアルタイムスキャン]ページで、[リアルタイムスキャン]がオンになっていることを確認します。
4. 「リアルタイムスキャンでの圧縮ファイルの処理」の下で[圧縮ファイルの内部をスキャン]をオンにします。

 **注:** また、アーカイブのスキャン設定は[Manual scanning]ページの「リアルタイムスキャンでのアーカイブファイルの処理」で個別に表示されます。

5. [暗号化されているアーカイブを危険とみなす]を選択すると、暗号化されているアーカイブをマルウェアとして自動的に扱います。
6. ネストされたアーカイブ内でスキャンする最大レベル数を設定します。
ネストされたアーカイブは他のアーカイブ内のアーカイブを意味します。
7. [ネストレベルの最大数を超えるアーカイブを危険とみなす]を選択すると、ネストレベルの最大数を超えるアーカイブをマルウェアとして自動的に扱います。

 **注:** デフォルトでは、設定した制限よりもネストレベルが高いアーカイブは安全と見なされます。

8. [保存して発行]をクリックします。
9. [OK]をクリックします。

マニュアルスキャンを設定する


手動スキャンの除外とアクションをで設定できます F-Secure Elements Endpoint Protectionポータル。

1. Elements Endpoint Protectionポータルで**プロファイル** > **Linux Protection**ページに移動します。
2. 編集するプロファイルの名前をクリックします。
3. [マニュアルスキャン]ページの[スキャンから除外されたファイルとフォルダ]で、[除外するパスを追加]をクリックして、スキャンから除外するファイルまたはフォルダを入力します。
ファイルとフォルダの除外には、ファイルまたはディレクトリの完全な絶対パス名が必要です。ワイルドカードは使用できません。
4. 「マニュアルスキャンの処理」で、[マルウェアの処理]と[不審なファイルの処理]を設定します。
 - ・ **名前の変更** 感染したファイルの名前を変更します。名前が変更されたファイルには、検出のタイプに応じて、.malwareまたは.suspectedという拡張子が付いています。
 - ・ **削除** 感染したファイルを削除します。
 - ・ **何もしない** 感染したファイルに対してアクションを実行しません。
5. [保存して発行]をクリックします。

6. [OK]をクリックします。

スケジュールスキャンの作成

スケジュールスキャンを使用して、管理対象コンピュータに対してマルウェアのスキャンを定期的に行うことができます。

注: スケジュールスキャンは、マニュアルスキャンに設定されている例外とアクション (処理)  を使用します。

1. Elements Endpoint Protectionポータルで**プロファイル** > **Linux Protection**ページに移動します。
2. 編集するプロファイルの名前をクリックします。
3. [マニュアルスキャン]ページに移動します。
4. 「**スケジュールスキャン**」の下で、スキャンを実行する時間を入力します。
ワイルドカード(*)を使用できます。たとえば、(*:***)を指定すると、スキャンを特定の時間に実行できないように設定できます。
5. スキャンを実行する曜日を選択します。
6. [保存して発行]をクリックします。
7. [OK]をクリックします。

3.7.2 F-Secure Elements Endpoint Protectionポータルでの整合性チェックの使用

整合性チェックをオンにすることができます [プロファイル] > [Linux保護]のページ F-Secure Elements Endpoint Protectionポータル。

1. Elements Endpoint Protectionポータルで**プロファイル** > **Linux Protection**ページに移動します。
2. 編集するプロファイルの名前をクリックします。
3. 「**完全性検査**」ページで[ファイルの完全性を確認する]をオンにします。
4. ベースラインに表示されている保護ファイルを確認します。
5. [保存して発行]をクリックします。
6. [OK]をクリックします。

完全性検査のベースラインにファイルを追加する

ファイルを完全性検査のリストに追加することで、ファイルを無断な変更から保護することができます。

1. Elements Endpoint Protectionポータルで**プロファイル** > **Linux Protection**ページに移動します。
2. 編集するプロファイルの名前をクリックします。
3. 「**完全性検査**」ページで[ファイルの完全性を確認する]がオンになっていることを確認します。
4. [新しいルールの追加]をクリックします。
5. 保護するファイルのパスを入力します。
6. 監視する属性を選択します。
 - ・ モード: 権限の変更
 - ・ 所有者: ファイルの所有権の変更
 - ・ グループ: ファイルグループの変更
 - ・ サイズ: ファイルサイズの変更
 - ・ 更新時間: ファイル更新時間の変更

リストに表示されているファイルのコンテンツは常に確認されます。

7. 監視対象ファイルに対する読み取りおよび書き込みアクセス権を選択します。
 - ・ [ファイルの書き込み] を [拒否] に選択すると、保護されているファイルに対する変更を阻止します。
 - ・ [ファイルの読み取り] を [拒否] に選択すると、改ざんされたファイルを開かないようにします。
8. [保存して発行] をクリックします。
9. [OK] をクリックします。


3.7.3 F-Secure Elements Endpoint Protectionポータルを使用した自動更新オプションの設定

最新のアップデートのインストール方法を制御したい場合、自動更新を設定してください。

自動更新の設定は、製品が使用する3つのF-Secure製品チャンネル linuxsecurity、fsbg、および baseguard に適用されます。自動更新の設定は、スキャンエンジンなどの他のチャンネルのアップデートが利用可能になるとすぐに適用されるため、影響はありません。自動更新の設定は、スキャンエンジンなどの他のチャンネルの更新には影響しません。自動更新をオフにすると、すべてのチャンネルの更新が無効になります。

また、製品チャンネルのアップデートは、アップデートが公開されてから1週間後に適用されます。

1. Elements Endpoint Protectionポータルで **プロファイル > Linux Protection** ページに移動します。
2. 編集するプロファイルの名前をクリックします。
3. [一般設定] ページに移動し、[自動更新を有効にする] がオンになっていることを確認します。
4. 自動更新のスケジュールを設定する
 - ・ **受信時に** この設定はデフォルトであり、製品とウイルス定義のアップデートは利用可能になるとすぐに適用されます。
 - ・ **1回** 指定した[日付]と[時刻]までアップデートの適用が保留されます。時刻はポリシーマネージャのローカルタイムゾーンで表示されます。
 - ・ **予定どおり > [毎日] [時刻]** にアップデートが適用されます。
 - ・ **予定どおり > [平日]** 選択した日の[時刻]にアップデートが適用されます。

 **注:** 各アップデートには有効期限があります。有効期限に達すると、日付と時間の設定に関係なくアップデートが適用されます。これにより、アップデートが適用されていないことでセキュリティが危うくなることに関する心配をなくします。
5. [アップデート後にアラートを送信] を選択すると、失敗したアップデートに対してアラートを生成します。
6. [保存して発行] をクリックします。
7. [OK] をクリックします。

HTTPプロキシを指定する

インターネットのアクセスが必要な機能にHTTPプロキシを指定できます。

現在、レピュテーションサービス(ORSP)のみがこの設定に依存していますが、今後製品が進化するにつれ、他のSecurity Cloud機能もそれを利用する可能性があります。

1. Elements Endpoint Protectionポータルで **プロファイル > Linux Protection** ページに移動します。
2. 編集するプロファイルの名前をクリックします。
3. [一般設定] ページに移動し、[HTTPプロキシを] をオンにします。
4. HTTPプロキシのホストアドレスとポート番号を入力します。

5. [保存して発行]をクリックします。
6. [OK]をクリックします。

3.8 lsctlユーティリティを使用して設定を構成する

lsctlコマンドラインユーティリティを使用して、製品の設定を確認および編集できます。


 **注:** 配布されたプロファイルの設定がElements Endpoint Protectionポータルでロックされている場合、その設定は強制され、lsctlコマンドラインユーティリティで編集することはできません。

lsctlユーティリティを使用するには

1. rootとしてホストにログインします。
2. 次のコマンドを使用してください。
 - 設定の現在の値を確認する: `/opt/f-secure/linuxsecurity/bin/lsctl get [OPTIONS] [SETTING]`
 - 設定に新しい値を割り当てる: `/opt/f-secure/linuxsecurity/bin/lsctl set [OPTIONS] [SETTING] [VALUE]`
 - 複数の値を保持する設定にエントリを追加する: `/opt/f-secure/linuxsecurity/bin/lsctl add [OPTIONS] [SETTING] [VALUE]`
 - 複数の値を保持する設定からエントリを削除する: `/opt/f-secure/linuxsecurity/bin/lsctl del [OPTIONS] [SETTING] [KEY]`
 - 設定の現在の値を初期化する: `/opt/f-secure/linuxsecurity/bin/lsctl reset [OPTION] [SETTING]`
 - 設定に使用できる操作のリストを表示する: `/opt/f-secure/linuxsecurity/bin/lsctl help [OPTIONS] [SETTING]`
 - バックアップから製品の設定をロードする: `/opt/f-secure/linuxsecurity/bin/lsctl load [OPTIONS] [SETTING] [VALUE]`
 - 使用できるすべての設定のリストを取得する: `/opt/f-secure/linuxsecurity/bin/lsctl -h`

get、set、add、del、loadコマンドは次のオプションをサポートします。


- json, -j** 入力をJSONとして扱い、出力をJSON形式で生成します。
- raw, -r** 現在のデフォルト形式です。Rawは、JSONに似た入出力形式であり、次の違いがあります。
 - 文字列値は受け入れられ、引用符で囲まらずに返されます。
 - ブーリアン値は、true、falseに加え、yes、noなども受け入れます。

 **注:** lsctlのデフォルト入出力形式は変更される可能性があります。lsctlに依存する場合、lsctlの実行時に--jsonまたは--rawを明示的に指定します。値の設定に加え、配列タイプの設定のキーは入力と見なされ、正しい形式で入力する必要があります。

set、add、loadコマンドは次のコマンドを受け入れます。

- prompt, -p** テキストエディタを使用して入力を求めます。
- file, -f [PATH]** ファイルから入力値を読み取ります。パスが-に設定されている場合、入力値は標準入力から読み取られます。

--promptまたは--fileオプションを使用する場合、[VALUE]をコマンドに入力しないでください。

-  **注:** set およびloadコマンドは、入力に含まれていない設定の処理方法が異なります。setコマンドは、入力に含まれていない設定の現在値を変更しませんが、loadコマンドは、これらの設定をデフォルト値にリセットします。

3.8.1 設定タイプと設定ツリー

各製品設定にはタイプと値があり、設定はツリー状に整理されています。

設定には5つの基本的なタイプがあります。

- ブール true または false
- テキスト
- 数値
- オブジェクトとは、各メンバーに名前を割り当てることで、特定の設定グループを1つのエンティティにまとめたものです。オブジェクトを検査したり、値を割り当てることができます。
- 配列は、さまざまな数のメンバーを含むコレクションです。各配列のメンバーは、一意のキーによって識別されます。配列のキーは、設定と同じ基本的な値の型で構成されています。配列内の設定を追加したり削除したりすることができます。各配列では、受け入れられる設定の種類に制限を設けることができます。たとえば、テキスト型の設定のみをメンバーとして受け入れることもできます。

本製品の設定はツリー構造で構成されており、ルートはオブジェクトタイプの設定で、他のすべての設定が含まれています。ルート設定を調べるには、get サブコマンドを使用します。例

```
# /opt/f-secure/linuxsecurity/bin/lscctl get
```

```
{
  "http_proxy": {
    "enabled": false,
    "host": "localhost",
    "port": 3128
  },
  "ic": {
    "enabled": true,
    "profiles": []
  },
  "oas": {
    "actions": {
      "malware": "rename",
      "pua": "none",
      "suspected": "none"
    },
    "archive_max_nested": 5,
    "block_archive_max_nested": false,
    "block_encrypted_archives": false,
    "detect_pua": true,
    "enabled": true,
    "exclude_paths": [],
    "include_paths": [],
    "scan_archives": false,
    "scan_only_executables": false
  },
  ...
}
```

この例には完全な出力が含まれていないことに注意してください。

出力は、ルートオブジェクトから始まり、そのすべての子設定 httpproxy、ic、oas などまで続く設定階層と、それらのサブセットが再帰的に表示します。

この例では、ルートオブジェクトに httpproxy 設定が含まれています。これは、3つのサブ設定である enabled(ブーリアン値設定)、host(テキスト設定)、port(番号設定)を含むオブジェクトタイプの設定です。

get サブコマンドを使用して、設定ツリーの一部のみを検査することもできます。オブジェクト型の設定が割り当てる名前と配列タイプの設定のキーを使用して、取得する設定を選択します。

たとえば、次のようにHTTPプロキシ設定のみを取得できます。

```
# /opt/f-secure/linuxsecurity/bin/lsctl get http_proxy
```

```
{
  "enabled": false,
  "host": "localhost",
  "port": 3128
}
```

同様に、さらにコマンドを指定して実行することで、HTTPプロキシの使用が有効になっているかどうかを確認することができます。

```
# /opt/f-secure/linuxsecurity/bin/lsctl get http_proxy enabled
```


```
false
```

タイプに加えて、設定は許可された値に追加の制約を持つことができます。たとえば、`http_proxy port` の設定では負の数は許可されません。

3.8.2 設定の操作

設定を操作するための基本的なサブコマンドは、`get`、`set`、`add`、`delete`、`reset`および`load`の6つです。

- get** `lsctl get [--json | --raw] SETTING...`
`get` サブコマンドを使用すると、設定の現在値を調べることができます。
- set** `lsctl set [--json | --raw] [--prompt | --file PATH] SETTING... VALUE`
`set` サブコマンドを使用すると、設定値を入力できます。
- add** `lsctl add [--json | --raw] [--prompt | --file PATH] SETTING... VALUE`
`add` サブコマンドを使用して、配列タイプの設定に新しいメンバーを追加します。新しい配列メンバーに使用されるキーは、値から自動的に派生します。キーがどのようにして派生されるかは、各設定に固有します。
- delete** `lsctl delete [--json | --raw] SETTING... KEY`
`delete` サブコマンドを使用して、配列型設定からメンバーを削除します。これは、配列型設定の直系の子孫である設定でのみ使用できることを意味します。
`del` を `delete` サブコマンドのエイリアスとして使用できます。
- reset** `lsctl reset [--json | --raw] [--all] SETTING...`
設定のローカル値をクリアするには、`reset` サブコマンドを使用します。これは、設定値がデフォルト値に戻されることを意味します。これにより、ポリシーマネージャーまたは Protection Service for Business を使用して設定を変更することもできます。
- load** `lsctl load [--json | --raw] [--prompt | --file PATH] SETTING... VALUE`
`load` サブコマンドを使用して、設定の値を復元します。

 **注:** `set` および `load` コマンドは、入力に含まれていない設定の処理方法が異なります。`set` コマンドは、入力に含まれていない設定の現在値を変更しませんが、`load` コマンドは、これらの設定をデフォルト値にリセットします。

設定値の提供

`set` および `add` サブコマンドは、設定値を変更するために使用されます。最後のコマンドライン引数を使用して、設定に割り当てる値 `set` サブコマンドの場合または配列型設定に追加するメンバー `add` サブコマンドの場合を入力します。

たとえば、`httpproxy host` 設定に新しい値を設定するには、次のコマンドを使用できます。

```
lsctl set http_proxy host example.com
```

これにより、`example.com` が `host` テキストタイプ設定の新しい値として割り当てられます。

`--prompt -p` フラグを使用すると、`lsctl` はテキストエディタを開き、設定の値を指定します。例

```
lsctl set --prompt http_proxy host
```

これにより、`host` 設定の新しい値を入力するために使用できるテキストエディタが開きます。

デフォルトでは、`lsctl` は、`EDITOR` 環境変数で指定されたエディタを開きます。`EDITOR` が設定されていないか、指定されたエディタが `PATH` にない場合、`lsctl` は、利用可能なエディタが見つかるまで、次のエディタを順番に見つけようとします。`nano`、`vim`、`vi`。

`--file -f` スイッチを使用すると、`lsctl` はファイルから設定値を読み取ります。例

```
lsctl set --file value.txt http_proxy host
```

これにより、`host` 設定の値が `value.txt` ファイルから読み取られます。特別な `-` 値を使用して、標準入力から値を読み取ることもできます。例

```
echo -n 'example.com' | lsctl set --file - value.txt http_proxy host
```

これは、`echo` コマンドの出力を `host` 設定の値として使用します。

関連タスク

[ファイルからの製品の設定をロードする](#) ページ30

`lsctl load` コマンドを使用して、以前に保存した JSON ファイルから設定を復元できます。

3.8.3 入力および出力フォーマット

フォーマットは、指定した設定値を解釈する方法と、`lsctl` が設定値を出力に出力する方法を定義します。

入力フォーマットは、入力された設定値をどのようにフォーマットするかを定義します。入力フォーマットはすべての入力ソースに影響を与えます。これは、設定値がコマンドライン引数を使用して入力された場合でも、ファイルから読み取られた場合でも `--file` スイッチを使用して重要ではないことを意味します。すべての入力は同じフォーマットルールに従わなければなりません。

出力フォーマットは、`lsctl` が出力するときに設定値がどのようにフォーマットされるかを定義します。

`lsctl` は現在、`json` および `raw` の2つの形式をサポートしています。これらは、`--json -j` および `--raw -r` のフラグをそれぞれ使用して選択できます。デフォルトでは、`lsctl` は現在 `raw` 形式を使用していますが、これは変更される可能性があります。`lsctl` の正確な入力または出力形式に依存している場合、これらのフラグを使用して、常に目的の形式を明示的に指定してください。

JSON

`--json` フラグを指定すると、`lsctl` は入力が有効な JSON であると想定し、有効な JSON を出力します。例

```
lsctl set --json http_proxy host 'example.com'
```

二重引用符の必要性に注意してください。JSON 文字列は常に二重引用符 `"` で囲まれます。ただし、`shell` の単語分割に対する処理方法により、`shell` が引用符を削除しないようにするため、JSON 文字列を追加の単一引用符で囲む必要があります。これは `shell` がコマンドライン引数を扱う方法に特有

のものであるため、値がファイル `--file` スイッチを使用してから読み取られる場合、またはインタラクティブエディター `--prompt` フラグから受信される場合は必要ありません。

フォーマットフラグも出力フォーマットに影響します。例

```
lsctl get --json http_proxy host
```

```
"example.com"
```

出力は有効なJSON文字列リテラルであることに注意してください。

Raw

`--raw` フラグを指定すると、`lsctl` は raw 形式を使用します。raw 形式はJSON形式をわずかに変更して、特定の操作をコマンドラインユーザーがより人間工学的に行えるようにします。json と raw 形式の違いは次のとおりです。

- 設定のテキスト値は、二重引用符で囲む必要はありません。これは、次のコマンドが有効であることを意味します。

```
lsctl set http_proxy host example.com
```

- 設定のブール値は、`true` および `false` に加えて、追加のリテラルはい、`y`、いいえ および `n` をサポートします。これにより、以下のコマンドが有効になります。

```
lsctl set http_proxy enabled yes
```

これらの追加は、テキストまたはブール値の設定を直接操作している場合にのみ適用されることに注意してください。他の設定の一部として表示される場合には適用されません。したがって、以下のコマンドは**無効**です。

```
lsctl set http proxy '{"enabled": yes, "host": example.com, "port": 9090}'
```

3.8.4 配列の操作

配列は他の設定と異なり、可変数のメンバー設定を含むことができます。

`add` および `delete` サブコマンドを使用して、メンバーを配列に追加したり、配列からメンバーを削除したりできます。

たとえば、文字列メンバーを含む配列設定である `oas include_paths` 設定に新しいメンバーを追加することで、オンアクセススキャンに含めるディレクトリを追加することができます。

```
lsctl add oas include_paths /home
```

上記のコマンドは、配列に `/home` ディレクトリを追加します。

次のコマンドを使用すると、`includepaths` から新しく追加されたディレクトリを削除できます。

```
lsctl del oas include_paths /home
```

配列には、より複雑な設定を含めることもできます。たとえば、完全性検査プロファイルは、`ic` プロファイル配列に格納されます。各完全性検査のプロファイルは、プロファイルプロパティを定義するオブジェクトです。次のコマンドを使用すると、新しい完全性検査プロファイルを追加できます。

```
lsctl add --file - ic profiles <<EOF
{
  "path": "/home",
  "verify_attributes": {
    "mode": false,
    "user": false,
```

```
"group": false,
"size": false,
"mtime": false
},
"read_action": "deny",
"write_action": "allow"
}
EOF
```

注: この例では、複数行入力を入力するためにシェルの 文書構文を使用しています。



--file スイッチと-引数を使用して、値が標準入力から読み取られることを示すことに注意してください。また、--prompt フラグを使用して、大きな入力値を簡単に入力できます。

次のコマンドを使用すると、新しく追加されたプロファイルを削除できます。

```
lsctl del ic profiles /home
```

ここでは、pathのメンバーはprofiles配列内のメンバーを識別するために使用されるキーです。配列の要素がどのように識別されるかは、それぞれの配列に固有します。

通常のサブコマンドを使用して、配列内の設定にアクセスすることもできます。たとえば、新しく追加されたプロファイル内のwriteaction設定の値を取得できます。

```
lsctl get ic profiles /home write_action
```

```
allow
```

または、verifyattributes内のmode設定の値を変更することもできます。

```
lsctl set ic profiles /home verify_attributes mode yes
```

入出力フォーマットは設定値の扱い方に影響を与えるだけでなく、配列キーの入力方法にも影響を与えます。たとえば、JSON形式を使用しているときにwrite_actionの設定値を取得したい場合は、次のようにlsctlコマンドを入力しなければなりません。

```
lsctl get --json ic profiles '/home' write_action
```

```
"allow"
```

3.8.5 ファイルからの製品の設定をロードする

lsctl loadコマンドを使用して、以前に保存したJSONファイルから設定を復元できます。

これにより、テスト済みで機能する製品構成をバックアップとして保存し、必要に応じて復元することができます。現在の製品構成を保存するには、例えば以下のコマンドを使用できます。

```
lsctl get --json > [PATH]
```

このコマンドでは、[PATH]をJSONバックアップファイルのフルパスとファイル名に置き換えます。

次のコマンドを実行して、製品の設定を復元します。

```
lsctl load --json --file [PATH]
```

これにより、[PATH]として指定されたバックアップファイルで使用可能なすべての設定が製品構成に適用されます。不足している設定はすべてデフォルト値にリセットされます。

3.8.6 製品の状態を検査する

lsctlコマンドラインユーティリティを使用して、製品の状態を検査したり、statusサブコマンドでさまざまなスキャンやパフォーマンスの統計情報を表示することができます。

ステータスの値の違いを調べる方法は、設定値を調べる方法と同様です。

- ステータス値を取得するには、lsctl status get [--json | --raw] STATUS...を実行します
- ステータスのヘルプを表示するには、lsctl status help [--json | --raw] STATUS...を実行します。


scan_service	タイプ: JSONオブジェクト スキャンサービスに関連するすべての統計情報。
scan_service connection_count	タイプ: 整数 接続されているホストの数。
scan_service last_detection	タイプ: 文字列 最後に検出された名前。
scan_service max_connection_count	タイプ: JSONオブジェクト 接続されているホストの最大数に関連するすべての統計情報。
scan_service max_connection_count daily	タイプ: 整数 過去24時間に接続されたホストの最大数。
scan_service max_connection_count weekly	タイプ: 整数 先週接続されたホストの最大数。
scan_service max_connection_count monthly	タイプ: 整数 先月の接続されたホストの最大数。
scan_service transaction_count	タイプ: JSONオブジェクト スキャントランザクションの数値に関連するすべての統計情報。
scan_service transaction_count total	タイプ: 整数 トランザクションの総数。
scan_service transaction_count daily	タイプ: 整数 過去24時間のトランザクション数。
scan_service transaction_count weekly	タイプ: 整数 先週のトランザクション数。
scan_service transaction_count monthly	タイプ: 整数 先月のトランザクション数。
scan_service detection_count	タイプ: JSONオブジェクト 検出値に関連するすべての統計情報。
scan_service detection_count total	タイプ: 整数 検出数。
scan_service detection_count daily	タイプ: 整数 過去24時間の検出数。
scan_service detection_count weekly	タイプ: 整数

	先週の検出数。
scan_service detection_count monthly	タイプ: 整数 先月の検出数。
scan_service min_transaction_latency	タイプ: JSONオブジェクト スキャントランザクションの最小遅延に関連するすべての統計情報。
scan_service min_transaction_latency daily	タイプ: 整数 過去24時間の最小トランザクション待ち時間 ミリ秒。
scan_service min_transaction_latency weekly	タイプ: 整数 先週の最小トランザクション待ち時間 ミリ秒。
scan_service min_transaction_latency monthly	タイプ: 整数 先月の最小トランザクション待ち時間 ミリ秒。
scan_service max_transaction_latency	タイプ: JSONオブジェクト スキャントランザクションの最大遅延に関連するすべての統計情報。
scan_service max_transaction_latency daily	タイプ: 整数 過去24時間の最大トランザクション待ち時間 ミリ秒。
scan_service max_transaction_latency weekly	タイプ: 整数 先週の最大トランザクション待ち時間 ミリ秒。
scan_service max_transaction_latency monthly	タイプ: 整数 先月の最大トランザクション待ち時間 ミリ秒。
scan_service avg_transaction_latency	タイプ: JSONオブジェクト スキャントランザクションの平均遅延に関連するすべての統計情報。
scan_service avg_transaction_latency daily	タイプ: 整数 過去24時間の平均トランザクション待ち時間 ミリ秒。
scan_service avg_transaction_latency weekly	タイプ: 整数 先週の平均トランザクション待ち時間 ミリ秒。
scan_service avg_transaction_latency monthly	タイプ: 整数 先月の平均トランザクション待ち時間 ミリ秒。
product_version	タイプ: JSONオブジェクト 製品バージョンに関連するすべての情報。
product_version name	タイプ: 文字列 固定された製品バージョンの名前。固定された製品バージョンが指定されていない場合、空の文字列になる可能性があります。
product_version status	タイプ: 文字列 固定された製品バージョンのステータスを表す文字列。値は、unset、error、in-progress、またはappliedになります。

3.8.7 リアルタイムスキャンのコマンドライン設定

lsctlで使用可能なリアルタイムスキャンに関連する設定。

ヒント: 設定の構造例を表示するには、次のコマンドの出力を確認できます:

```
 /opt/f-secure/linuxsecurity/bin/lsctl get [--json|--raw] [SETTING]
```


oas	タイプ: JSONオブジェクト リアルタイムスキャンに関連するすべての設定。
oas enabled	タイプ: ブーリアン リアルタイムスキャンのオン/オフを制御します。
oas include_paths	タイプ: JSON配列 リアルタイムスキャンに含まれるパス。
oas include_paths [PATH]	タイプ: 文字列 リアルタイムスキャンに含まれるパス。addおよびdelコマンドに使用されます。[PATH]は完全な絶対パス名である必要があります。
oas exclude_paths	タイプ: JSON配列 リアルタイムスキャンから除外されているパス。
oas exclude_paths [PATH]	タイプ: 文字列 リアルタイムスキャンから除外されたパス。addおよびdelコマンドに使用されます。[PATH]は完全な絶対パス名である必要があります。
oas actions	タイプ: JSONオブジェクト リアルタイムスキャンのアクション構成。
oas actions malware	タイプ: 文字列 マルウェアとして検出されたファイルに対するアクション。設定できる値: <ul style="list-style-type: none"> remove - ファイルを削除 名前の変更 - ファイル名に .malware の拡張子を追加します なし - ファイルに対して何もしません
oas actions suspected	タイプ: 文字列 安全ではないと疑いがあるファイルに対するアクション。設定できる値: <ul style="list-style-type: none"> remove - ファイルを削除 rename - ファイル名に .suspected の拡張子を追加します なし - ファイルに対して何もしません
oas detect_pua	タイプ: ブーリアン 不要と思われるアプリケーション (PUA) のスキャンをオンにするかオフにするかを制御します。
oas actions pua	タイプ: 文字列 不要と思われるアプリケーションとして検出されたファイルに対するアクション。設定できる値: <ul style="list-style-type: none"> remove - ファイルを削除 rename - ファイル名に .pua の拡張子を追加します なし - ファイルに対して何もしません
oas scan_archives	タイプ: ブーリアン

	アーカイブファイル内のスキャンをオンにするかオフにするかを制御します。
oas block_archive_max_nested	タイプ: ブーリアン ネストが深すぎるアーカイブファイルを安全でないとみなすかどうかを制御します。
oas archive_max_nested	タイプ: 整数 アーカイブファイルの入れ子の最大許容レベル。 <code>oas block_archive_max_nested</code> が <code>true</code> に設定されている場合、このレベルのネストを超えるアーカイブファイルは安全ではないとみなされます。
oas block_encrypted_archives	タイプ: ブーリアン 暗号化されたアーカイブファイルを安全でないとみなすかどうかを制御します。
oas scan_only_executables	タイプ: ブーリアン スキャンを実行権限を持つファイルに制限するかどうかを制御します。

マニュアル/スケジュールスキャンの設定

`lsctl` で使用可能なマニュアル/スケジュールスキャンに関連する設定。

ヒント: 設定の構造例を表示するには、次のコマンドの出力を確認できます:

```
 /opt/f-secure/linuxsecurity/bin/lsctl get [--json|--raw] [SETTING]
```


ods	タイプ: JSONオブジェクト マニュアル/スケジュールスキャンに関連するすべての設定。
ods exclude_paths	タイプ: JSON配列 スケジュールスキャンから除外されているパス。
ods exclude_paths [PATH]	タイプ: 文字列 スケジュールスキャンから除外されたパス。 <code>add</code> および <code>del</code> コマンドに使用されます。 <code>[PATH]</code> は完全な絶対パス名である必要があります。
ods actions	タイプ: JSONオブジェクト マニュアル/スケジュールスキャンのアクション構成。
ods actions malware	タイプ: 文字列 マルウェアとして検出されたファイルに対するアクション。設定できる値: <ul style="list-style-type: none"> • <code>remove</code> - ファイルを削除 • 名前の変更 - ファイル名に <code>.malware</code> の拡張子を追加します • なし - ファイルに対して何もしません
ods actions suspected	タイプ: 文字列 安全ではないと疑いがあるファイルに対するアクション。設定できる値: <ul style="list-style-type: none"> • <code>remove</code> - ファイルを削除 • <code>rename</code> - ファイル名に <code>.suspected</code> の拡張子を追加します • なし - ファイルに対して何もしません
ods detect_pua	タイプ: ブーリアン

	不要と思われるアプリケーション(PUA)のスキャンをオンにするかオフにするかを制御します。
ods actions pua	<p>タイプ: 文字列</p> <p>不要と思われるアプリケーションとして検出されたファイルに対するアクション。設定できる値:</p> <ul style="list-style-type: none"> • remove - ファイルを削除 • rename - ファイル名に .pua の拡張子を追加します • なし - ファイルに対して何もしません
ods scan_archives	<p>タイプ: ブーリアン</p> <p>アーカイブファイル内のスキャンをオンにするかオフにするかを制御します。</p>
ods block_archive_max_nested	<p>タイプ: ブーリアン</p> <p>ネストが深すぎるアーカイブファイルを安全でないとみなすかどうかを制御します。</p>
ods archive_max_nested	<p>タイプ: 整数</p> <p>アーカイブファイルの入れ子の最大許容レベル。ods block_archive_max_nested が true に設定されている場合、このレベルのネストを超えるアーカイブファイルは安全ではないとみなされます。</p>
ods block_encrypted_archives	<p>タイプ: ブーリアン</p> <p>暗号化されたアーカイブファイルを安全でないとみなすかどうかを制御します。</p>
ods schedule	<p>タイプ: JSONオブジェクト</p> <p>スケジュールスキャンに関連するすべての設定。</p>
ods schedule schedule_type	<p>タイプ: 文字列</p> <p>使用中のスケジュールスキャンの種類。現在、weekly のみ設定できます。</p>
ods schedule weekly_schedule	<p>タイプ: JSONオブジェクト</p> <p>毎週のスケジュールでスキャンするための設定。</p>
ods schedule weekly_schedule [monday tuesday wednesday thursday friday saturday sunday]	<p>タイプ: ブーリアン</p> <p>指定された曜日にスケジュールスキャンを実行するかどうかを制御する設定。</p>
ods schedule weekly_schedule time_of_day	<p>タイプ: 文字列</p> <p>設定された日にスケジュールスキャンを開始する時間。値は、エンドポイントのタイムゾーンの24時間の時間値 HH:MM です。</p>

ORSP 設定

lsctl で使用可能なオブジェクトレピュテーションサービスプラットフォーム (ORSP) に関連する設定。

ヒント: 設定の構造例を表示するには、次のコマンドの出力を確認できます:

```
 /opt/f-secure/linuxsecurity/bin/lsctl get [--json|--raw] [SETTING]
```


orsp	<p>タイプ: JSONオブジェクト</p> <p>クラウドベースのオブジェクト評価サービスプラットフォーム ORSP のスキャンでの使用に関連する設定。</p>
-------------	---

updates enabled	<p>タイプ: ブーリアン</p> <p>自動更新をオンまたはオフに切り替えるかどうかを制御します。</p>
updates product_version	<p>タイプ: 文字列</p> <p>注: この設定は使用しないでください。この設定は将来の機能リリースのために予約されています。</p>
updates schedule	<p>タイプ: JSONオブジェクト</p> <p>自動更新のスケジュール。</p>
updates schedule regime	<p>タイプ: 文字列</p> <p>自動更新スケジュールのタイプ。次の値が許可されています。</p> <ul style="list-style-type: none"> • on_arrival - アップデートは利用可能になった時点ですぐに適用されます • at_date - アップデートの適用は at_date_schedule の設定で指定された時刻まで保留となります • with_repetition - アップデートは repetition_schedule の設定にしたがって定期的に適用されます
updates schedule at_date_schedule	<p>タイプ: 整数</p> <p>regime が at_date に設定されたときにアップデートを適用する時間。値はエポック時間形式のタイムスタンプである必要があります。</p>
updates schedule repetition_schedule	<p>タイプ: JSONオブジェクト</p> <p>定期的に繰り返される自動更新スケジュールに関連する設定。</p>
updates schedule repetition_schedule day	<p>タイプ: 文字列</p> <p>繰り返し更新される予定日。指定可能な値は、daily(毎日特定の時間にアップデートを確認する)、monday、tuesday、wednesday、thursday、friday、saturday、sunday(毎週特定の曜日にアップデートを確認する)。</p>
updates schedule repetition_schedule time_of_day	<p>タイプ: 文字列</p> <p>アップデートを適用するためにスケジュールされた時刻。値は24時間値(HH□MM)です。</p>

HTTPプロキシ設定

lsctl で使用可能なHTTPプロキシの使用に関連する設定。

ヒント: 設定の構造例を表示するには、次のコマンドの出力を確認できます:

```
 /opt/f-secure/linuxsecurity/bin/lsctl get [--json|--raw] [SETTING]
```

http_proxy	<p>タイプ: JSONオブジェクト</p> <p>すべてのHTTPプロキシ設定</p>
http_proxy enabled	<p>タイプ: ブーリアン</p> <p>ネットワークのアクセスが必要な機能にHTTPプロキシを使用するかどうかを制御します。</p>
http_proxy host	<p>タイプ: 文字列</p> <p>HTTPプロキシのホスト名。</p>
http_proxy port	<p>タイプ: 整数</p>

HTTPプロキシのポート。値は1~65535の間の数値である必要があります。

3.9 コマンドラインを使用する

ここでは、本製品のコマンドラインから実行できるコマンドについて説明します。

3.9.1 サービスの開始と停止

本製品には、`/opt/f-secure/fsbg/bin/master-switch`コマンドを使用して制御できるいくつかのバックグラウンドサービスが含まれています。

`master-switch` コマンドを使用して、本製品に関連するすべてのサービスを制御することができます。コマンドの構文は以下の通りです。

```
/opt/f-secure/fsbg/bin/master-switch ( on | off | status | enable | disable | is-enabled )
```

すべてのF-Secureサービスを一時的に停止するには、`off`サブコマンドを使用します。

```
/opt/f-secure/fsbg/bin/master-switch off
```

`on`サブコマンドは、停止したサービスを再開します。

```
/opt/f-secure/fsbg/bin/master-switch on
```

F-Secureサービスのステータスを表示するには、`status`サブコマンドを使用します。

```
/opt/f-secure/fsbg/bin/master-switch status
```

`disable` サブコマンドは、すべてのF-Secureサービスをオフにします。コンピュータを再起動した後も、サービスはオフのままになります。`enable`サブコマンドは、すべてのF-Secureサービスを再びオンにします。`is-enabled` サブコマンドを使用して、F-Secureサービスがオンになっているかどうかを確認できます。

3.9.2 コマンドラインからコンピュータを手動でスキャンする

製品がインストールされているLinuxホストのコマンドラインから手動でコンピュータに対するマルウェアスキャンを行えます。

`fsanalyze`プログラムは、実行するユーザの権限でファイルをスキャンします。ユーザは、ファイルへの読み取りアクセス権と、スキャン対象のすべてのディレクトリへの読み取りおよび実行アクセス権が必要とします。感染したファイルの名前を変更または削除を行うには、ファイルが置かれているディレクトリへの書き込みアクセス権が必要です。

`fsanalyze` コマンドは特定のターゲット(ファイルまたはディレクトリ)をスキャンし、検出した悪質なコードを報告します。

コマンドラインでファイルが指定されていない場合、`fsanalyze`は分析するコンテンツの標準入力を読み取ります。それ以外の場合、指定されたすべてのファイルが分析され、指定されたすべてのディレクトリが次の例外で再帰的にスキャンされます。

- 特殊なファイル(ソケット、FIFO、デバイスファイルなど)、および `/proc`と `/sys`の特殊ファイルシステム上のファイルはスキップされます。
- 再帰的スキャンは、`--follow`コマンドラインオプションで明示的に要求されない限り、シンボリックリンクを追従しません。

- 再帰的スキャンでは、ファイルシステムの境界を自動的に越えることはありません。他のファイルシステムにマウントされたファイルをスキャンするには、コマンドラインでマウントポイントを明示的に指定します。

注: マニュアルスキャンの現在のポリシーで設定されている例外は、コマンドラインからマニュアルスキャンを使用する場合に適用されます。

マニュアルスキャンを開始するために次のコマンドをシェルから実行します:

```
/opt/f-secure/linuxsecurity/bin/fsanalyze [OPTIONS] [FILE...]
```

使用方法: `fsanalyze [OPTIONS] [FILE...]`

潜在的な悪意のあるコンテンツがないか、各FILEを分析します。ファイルが指定されていない場合は、標準入力を分析します。この場合、サポートされているアクションはnoneだけになります。

-h, --help	ヘルプを表示します。
--malware=ACTION	ファイルがマルウェアとして検出されたときに実行するアクション(処理)を示します。アクションが <code>remove</code> (削除) の場合、ファイルは削除されます。アクションが <code>rename</code> (名前の変更) の場合、ファイルの拡張子に検出タイプが追加され、ファイル名が変更されません。アクションが <code>none</code> (なし) の場合、感染は標準出力にのみ報告されます。 デフォルト <code>rename</code>
--pua=ACTION	ファイルが不要と思われるアプリケーションとして検出されたときに実行するアクションを示します。アクションのオプションはマルウェアの場合と同じです。 デフォルト: <code>none</code> (なし)
--suspected=ACTION	ファイルが不審として検出されたときに実行するアクションを示します。アクションのオプションはマルウェアの場合と同じです。 デフォルト: <code>none</code> (なし)
-L, --follow	シンボリックリンクに従います。
-l, --list	スキャンしたすべてのファイルを一覧表示します。
--preserve-atime=VALUE	スキャンされたファイルのアクセス時間を保持します。値は <code>yes</code> または <code>no</code> になります。 デフォルト: <code>no</code> (いいえ)
-q, --quiet	スキャンの結果のみ出力します。
--scan-archives=VALUE	アーカイブスキャンを有効にします。値は <code>yes</code> または <code>no</code> です。 デフォルト: <code>no</code> (いいえ)
--max-nested=NUMBER	スキャンされたアーカイブの最大許容ネストレベルを設定します。 デフォルト 5
--detect-max-nested=VALUE	有効にした場合、最大ネスト数を超えるアーカイブをマルウェアとして扱います。値は <code>yes</code> または <code>no</code> になります。 デフォルト: <code>no</code> (いいえ)
--detect-encrypted-archives=VALUE	暗号化されたアーカイブをマルウェアとして扱います。値には <code>yes</code> か <code>no</code> を指定します。 デフォルト: <code>no</code> (いいえ)
--detect-pua=VALUE	無効にすると、不要と思われるアプリケーションは無視されます。値は <code>yes</code> または <code>no</code> です。

	デフォルト <code>yes</code>
<code>--use-orsp=VALUE</code>	有効になっている場合は、Security Cloudのレピュテーションサービス ORSP を使用します。値は <code>yes</code> または <code>no</code> になります。 デフォルト <code>yes</code>
<code>-v, --version</code>	プログラムのバージョンを表示して終了します。
<code>--quoting-style=STYLE</code>	出力の引用方法を設定します。STYLE が <code>url</code> の場合、出力は URL エンコードされます。STYLE が <code>escape</code> の場合、出力の ASCII 制御文字は <code><NAME></code> としてエスケープされ、出力がファイルにリダイレクトされない場合は、それらも強調表示されます。STYLE が <code>literal</code> の場合、出力はそのまま表示されます。 デフォルト <code>url</code>
<code>--ignore-exclude-paths</code>	マニュアルスキャンの現在のポリシーから設定された除外を使用しません。

スキャンの終了コードは次のとおりです。

- 0 = スキャンはエラーなしで完了し、マルウェア、PUA、または不審なコンテンツは検出されていません
- 1 = 1つ以上のファイルのスキャンに失敗しました
- 2 = スキャンはエラーなしで完了し、マルウェア、PUA、または不審なコンテンツが検出されました

3.9.3 コマンドラインから手動で製品を更新する

製品、マルウェア定義データベース、スキャンエンジンは、`lsctl` プログラムを使用して手動で最新バージョンに更新することができます。

root ユーザとして次のコマンドを実行することにより、コマンドラインから製品を更新できます。

```
/opt/f-secure/linuxsecurity/bin/lsctl update
```

このコマンドは、ネットワーク経由で新しいアップデートの有無を確認し、システムにダウンロードしてインストールすることで、すべてを最新の状態にします。

`--timeout` オプションを使用すると、更新が完了するまでの待ち時間を秒単位で指定し、バックグラウンドで終了させることができます。このオプションが指定されていない場合、デフォルトとして使用される値 `0` は、タイムアウトを無効化します。

注: 更新前および更新中に考慮すべきことがいくつかあります。



- コマンドを実行する前に、製品サービスが実行されていることを確認してください。
- アップデートが開始されると、アップデートのダウンロードとインストールをキャンセルすることはできません。`--timeout` オプションを使用する場合、タイムアウト後、操作はバックグラウンドで完了します。

関連概念

[サービスの開始と停止](#) ページ 38

本製品には、`/opt/f-secure/fsbg/bin/master-switch`コマンドを使用して制御できるいくつかのバックグラウンドサービスが含まれています。

3.9.4 アンチウイルス保護の動作確認

本製品が正常に機能しているか確認するために、ウイルスとして検出される専用のテストファイルを使用できます。

テストファイルはEICAR (European Institute of Computer Anti-virus Research) Standard Anti-Virus Testファイルといい、他のアンチウイルスプログラムでも検出されます。EICARの情報は以下のURLでご覧になれます。
<https://www.f-secure.com/v-descs/eicar.shtml>

1. EICARのテストファイルをダウンロードまたは作成します。
 - EICAR テスト ファイルを次の URL からダウンロードします。
http://www.europe.f-secure.com/virus-info/eicar_test_file.shtmlまたは、
 - テキストエディタを使用して、次の1行を持つeicar.comファイルを作成します。
`X5O!P%@AP[4\PZX54(P^)7CC)7}$EICAR-STANDARD-ANTIVIRUS-TEST-FILE!$H+H*`
2. 次のコマンドを実行します:`/opt/f-secure/linuxsecurity/bin/fsanalyze eicar.com`
 ウイルスがファイル内に検出されます。

3.9.5 コマンドラインから完全性検査を実行する

完全性検査の機能の一部はコマンドラインから利用できます。

Linux ホストのコマンドラインから完全性検査を実行するには

1. `root` としてホストにログインします。
2. 次のコマンドをシェルから実行します:`/opt/f-secure/linuxsecurity/bin/fsic`
 説明:
 - `fsic verify`:ポリシーに構成されているベースライン内の各ファイルをチェックします。ファイルが改ざんされていないかチェックし、ファイルに対してマルウェアのスキャンを行います。このコマンドは、検証に失敗したすべてのファイルを標準に出力します。
 - `fsic unprotect`:ベースラインファイルの変更を許可します。
 - `fsic protect`:ベースラインインデックスが再生成され、ベースラインファイルへの変更が阻止されます。
3. コマンドの終了コードを確認してください。
 - 0 = エラーや検証の失敗の結果は発生していません
 - 1 = エラーが発生したため、コマンドを正常に実行できません
 - 2 = 1つ以上のファイルがマルウェア、不要と思われるコンテンツ、または疑わしいコンテンツで改ざんされている

付録 A


警告の深刻度


警告には次の深刻度があります。

深刻度	Syslog priority	説明
情報	info	ホストからの通常の操作情報。
警告	warning	ホストからの警告。 ファイルの読み込みにエラーが発生した場合などに発生します。
エラー	err	ホスト上の回復可能なエラー。 ウイルス定義ファイルの更新が以前に受信したものよりも古いときなどに発生します。
重大なエラー	emerg	ホスト上の回復不能なエラー。 処理の実行エラー、カーネルモジュールのエラーなどに発生します。
セキュリティ警告	alert	ホスト上のセキュリティ警告。 感染および実行された処理の情報を含みます。

製品と一緒にインストールされるサービス

本製品には、さまざまな製品機能を処理するために使用されるサービスをインストールします。

 **注:** 記載されているサービスは、予告なしに変更される場合があります。

 **重要:** ここにリストされているサービスを手動で開始または停止しないでください。

アクティブなサービス

f-secure-baseguard-accd.service	ファイルアクセスの通知を処理するリアルタイムスキャナーサービス。
f-secure-baseguard-as.service	メールのスパムスキャンのためのBaseGuardの機能です。 注 linuxsecurityチャンネルバージョン12.0.286では、このサービスは実行されなくなりました。
f-secure-baseguard-av.service	下位互換性のために必要なサービススタブ。
f-secure-baseguard-cleanup.service	ディスク容量を節約するために、古いチャンネルのアップデートがクリーンアップされるようにします。
f-secure-baseguard-doormand.service	サブスクリプション、登録、およびクラウドサービスルックアップの処理を容易にします。
f-secure-baseguard-icap.service	リアルタイム、スケジュールおよびマニュアルスキャンに使用されるマルウェア分析サービス。
f-secure-baseguard-orspgw.service	F-Secureのオンラインレピュテーションサービスのローカルプロキシ。
f-secure-baseguard-tokenverify.service	HTTPAPIで使用されるOAuth2認証マネージャー。

	このサービスは実行していますが、製品では使用されていません。
f-secure-baseguard-update.service	F-SecureのGUTS2サービスでチャンネルの更新を監視し、fsbg-updated.serviceに通知を送信します。 注意 BaseGuardチャンネルのバージョン1.0.574では、このサービスは実行されなくなりました。
f-secure-linuxsecurity-fsacd.service	完全性検査のベースラインを維持します。
f-secure-linuxsecurity-scand.service	マニュアルスキャンとスケジュールスキャンを管理します。
f-secure-linuxsecurity-lspmd.service	集中管理された設定を製品サービスにローカルに配布します。
f-secure-linuxsecurity-statusd.service	製品サービスからステータスと統計情報を収集し、それらを集中管理エージェント fsma2 に中継します。
f-secure-linuxsecurity-webserver.service	一元管理された設定に内部インターフェースを提供します。
fsbg-updated.service	オンラインチャンネルアップデートのインストールをスケジュールします。
fsbg-statusd.service	BaseGuardサービスからステータスと統計情報を収集し、それらを集中管理エージェント fsma2 に中継します。
fsbg-pmd.service	集中管理された設定をBaseGuardサービスにローカルに配布します。
fsma2.service	集中管理された設定と通信を処理します 例 ポリシーマネージャまたはF-Secure Elements Endpoint Protection

非アクティブなサービス

これらのサービスはインストール中に含まれますが、製品によってロードされることはありません。


f-secure-baseguard-authorize.service	HTTP APIで使用されるOAuth 2承認サーバ。
---	-----------------------------

f-secure-baseguard-sensor.service

F-Secure Elements Endpoint Detection and Responseの機能を処理します。使用中のサブスクリプションキーでEndpoint Detection and Response機能が有効になっている場合、このサービスはアクティブになりますが、デフォルトでは非アクティブになっています。

クラウドサービス

この製品は、マルウェア定義の更新を確認するなど、ネットワークを介してさまざまなサービスに接続します。

 **注:** 次の表に記載されている詳細は変更される場合があります。

サービスアドレス	プロトコル	ポート	アクセス済み
*.fsapi.com	HTTPS	443	時折、さまざまなサービスのために
aspam.sp.f-secure.com	HTTPS	443	スパム対策機能に関連して、サービスの再起動時に、およびスパムスキャン中に非常に頻繁に使用されます
guts2.sp.f-secure.com	HTTP	80	毎時、アップデートの処理に使用されます
[*.]orsp.f-secure.com	HTTP	80	非常に頻繁に、レピュテーションリクエストによってトリガーされます
