# Red Hat Enterprise Linux 6

## セキュリティーガイド

セキュア化するためのガイド Red Hat Enterprise Linux

Last Updated: 2020-10-22

# Red Hat Enterprise Linux 6 セキュリティーガイド

セキュア化するためのガイド Red Hat Enterprise Linux

Mirek Jahoda
Red Hat Customer Content Services
mjahoda@redhat.com

Robert Krátký
Red Hat Customer Content Services

Martin Prpič
Red Hat Customer Content Services

Tomáš Čapek
Red Hat Customer Content Services

Stephen Wadeley
Red Hat Customer Content Services

Yoana Ruseva
Red Hat Customer Content Services

Miroslav Svoboda
Red Hat Customer Content Services

## 法律上の通知



## 概要

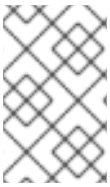
本書は、ユーザーおよび管理者が、ローカルおよびリモートの侵入、悪用、悪意のある行為に対してワークステーションおよびサーバーを保護するプロセスおよびプラクティスを学ぶのに役に立ちます。 本ガイド Red Hat Enterprise Linux は、すべての Linux システムで有効な概念および技術に重点を置き、データセンター、ワークプレース、およびホーム用にセキュアなコンピューティング環境を構築するのに必要な計画およびツールについて詳しく説明します。 管理上の適切な知識、警戒体制、およびツールを備えることで、Linux を実行しているシステムの機能をフルに活用して、大概の一般的な侵入や悪用の手法からシステムを保護できます。

# 目次

# 第1章 セキュリティーの概要

ビジネスの運営や個人情報の把握ではネットワーク化された強力なコンピューターへの依存度が高まっていることから、各種業界ではネットワークとコンピューターのセキュリティーの実践に関心が向けられています。企業は、システム監査を適正に行い、ソリューションが組織の運営要件を満たすようにするために、セキュリティーの専門家の知識と技能を求めてきました。多くの組織はますます動的になってきていることから、従業員は、会社の重要な IT リソースに、ローカルまたはリモートからアクセスするようになっています。このため、セキュアなコンピューティング環境に対するニーズはより顕著になっています。

ただし、多くの組織（個々のユーザーも含む）は、パフォーマンス、生産性、便利さ、使いやすさ、および予算面の懸念事項に反し、セキュリティーをよりよく考慮しています。適切なセキュリティー実装は、無許可の侵入が発生し て はじめて後続的なものになることがよくあります。多くの侵入の試みを阻止する効果的な方法は、インターネットなどの信頼できないネットワークにサイトを接続する前に、適切な措置を講じることです。



**注記**

本ガイドでは、**/lib** ディレクトリー内のファイルへの参照が複数作成されます。64 ビットシステムを使用する場合は、上述のファイルの一部がにある可能性があり **/lib64**ます。

## 1.1. セキュリティーの概要

### 1.1.1. コンピューターセキュリティーとは

コンピューターセキュリティーは、コンピューティングと情報処理の幅広い分野で使用される一般的な用語です。コンピューターシステムとネットワークを使用して日々の業務を行い、重要な情報へアクセスしている業界では、企業データを総体的資産の重要な部分であると見なしています。総保有コスト (Total Cost of Ownership: TCO)、投資利益率 (Return on Investment: ROI)、サービスの品質 (Quality of Service: QoS) などの用語や評価指標は日常的なビジネス用語として用いられるようになっています。各種の業界が、計画およびプロセス管理コストの一環として、これらの評価指標を用いてデータ保全性や可用性などを算出しています。電子商取引などの業界では、データの可用性と信頼性は、成功と失敗の違いを意味します。

#### 1.1.1.1. コンピューターセキュリティーのサポート状況

情報セキュリティーは、パブリックネットワークへの依存度が高くなり、個人、商取引などの制限された情報を開示しないことから、情報セキュリティーが進化しました。Mitnick などのインスタンスが多数あります。[1] and the Vladimir Levin [2] すべての業界の組織に対し、送信や開示など、情報の処理方法を再調査するよう促したケースです。インターネットは最も重要な開発開発の 1 つで、データセキュリティーの集約にあふれる努力を促しました。

個人コンピューターを使用して、インターネットが提供する必要のあるリソースへのアクセスを取得する場合が常に多数あります。電子商取引や商取引のトランザクションに関する調査および情報から、インターネットは 20 つ目の最も重要な開発の 1 つとして取り上げられました。

ただし、インターネットとその以前のプロトコルは *信頼ベース* のシステムとして開発されました。これは、インターネットプロトコル(IP)自体を保護するように設計されていませんでした。承認されたセキュリティー標準は TCP/IP 通信スタックに組み込まれず、ネットワーク全体で悪意のあるユーザーやプロセスに開放されます。最近の開発では、インターネット通信がより安全になりましたが、いくつかのインシデントに注意を向け、完全に安全でないという事実に警告します。

#### 1.1.1.2. Security Today

2000 年 2 月、分散型サービス拒否(DDoS)攻撃がインターネットで最も高速なサイトの一部に取り消されました。この攻撃でレンダリングされた yahoo.com、amazon.com、Amazon.com、および他のいくつかのサイトは、通常のユーザーに完全に到達できません。これは、高速バイトの ICMP パケット転送（*ping フラッド* とも呼ばれる）がルーターを関連付けられているためです。この攻撃は、特別に作成された、脆弱なネットワークサーバーをスキャンする幅広く利用可能なプログラムを使用する不明な攻撃者によって引き起こされました。このプログラムは、サーバー *上* にトールの木車と呼ばれるクライアントアプリケーションをインストールします。その後、すべての大幅なサーバーで攻撃を阻止し、それらを利用不可能にしました。使用するルーターやプロトコルの方法において、基本的な欠陥に対する攻撃の多くが、パケットの目的や送信先の場所に関係なく、すべての受信データを受け入れられるように構造化されます。*は削除される可能性があります。*

2009 年に、Wired Equivalent Privacy(WEP)ワイヤレス暗号化プロトコルの広く知られた特性を悪用すると、グローバルの盗難に、45万以上のクレジットカード番号の盗難が発生しました。

ただし、システムおよびネットワークのセキュリティーは難しくなる可能性があり、組織が情報をどのように検討するか、使用、操作、および送信を行う方法について厳密な知識が必要です。適切なセキュリティープランの実装には、組織（および組織を構成している人）がどのようにビジネスを行うかを理解することができます。

#### 1.1.1.3. セキュリティーの標準化

企業はどの業界でも、米国医師会 (AMA: American Medical Association)、米国電気電子学会 (IEEE: Institute of Electrical and Electronics Engineers) などの標準化推進団体が作成する規制やルールに従っています。情報セキュリティーにも同じことが当てはまります。多くのセキュリティーコンサルタントやベンダーが *機密性 (Confidentiality)*、*保全性 (Integrity)*、*可用性 (Availability)* の頭文字をとった CIA として知られる標準セキュリティーモデルを採用しています。この 3 階層モデルは、機密情報のリスク評価やセキュリティー方針の確立において、一般的に採用されているモデルです。以下でこの CIA モデルを説明します。

- 機密性 - 機密情報は、事前に定義された個人だけが利用できるようにする必要があります。許可されていない情報の送信や使用は、制限する必要があります。たとえば、情報の機密性により、権限のない個人が顧客情報や規制情報が悪意のある目的（ID 盗難やクレジットカードなど）で取得されないようにします。

- 保全性 - 情報は、改ざんして不完全または不正確なものにすべきではありません。承認されていないユーザーが、機密情報を変更したり破壊したりする機能を使用できないように制限する必要があります。

- 可用性 - 情報は、認証されたユーザーが必要な時にいつでもアクセスできるようにする必要があります。可用性は、合意した頻度とタイミングで情報を入手できることを保証します。これは、パーセンテージで表されることが多く、ネットワークサービスプロバイダーやその企業顧客が使用するサービスレベルアグリーメント (SLA) で正式に合意となります。

### 1.1.2. SELinux

Red Hat Enterprise Linux には SELinux と呼ばれる Linux カーネルの拡張機能が含まれており、システム内のファイル、プロセス、ユーザー、アプリケーションに対する詳細な制御レベルを提供する Mandatory Access Control(MAC)アーキテクチャーを実装しています。SELinux の詳細は、本書の範囲外になりますが、SELinux と Red Hat Enterprise Linux でのその使用の詳細については、『『Red Hat Enterprise Linux SELinux ユーザーガイド』を参照してください。』SELinux で保護されているサービスの設定および実行の詳細は、『SELinux 『Managing Confined Services Guide』を参照してください』。SELinux で利用可能なその他のリソースは、に記載されてい 11章 *リファレンス* ます。

### 1.1.3. セキュリティーコントロール

多くの場合、コンピューターセキュリティーは、一般的に以下の 3 つのマスターカテゴリーに分類されます。 *Controls（制御）*:

- 物理的
- 技術的
- 管理的

この 3 つのカテゴリーは、セキュリティーの適切な実施における主な目的を定義するものです。このコントロールには、コントロールと、その実装方法を詳細化するサブカテゴリーがあります。

#### 1.1.3.1. 物理的コントロール

物理的コントロールは、機密資料への非認証アクセスの抑止または防止のために、明確な構造でセキュリティー対策を実施します。物理的コントロールの例は以下のとおりです。

- 有線監視カメラ
- 動作または温度の感知アラームシステム
- 警備員
- 写真付き身分証明書
- 施錠された、デッドボルト付きのスチールドア
- バイオメトリクス (本人確認を行うための指紋、声、顔、虹彩、筆跡などの自動認識方法が含まれます)

#### 1.1.3.2. 技術的コントロール

技術的コントロールでは、物理的な構造物やネットワークにおける機密データのアクセスや使用を制御する基盤となる技術を使用します。技術的コントロールは広範囲に及び、以下のような技術も含まれます。

- 暗号化
- スマートカード
- ネットワーク認証
- アクセス制御リスト (ACL)
- ファイルの完全性監査ソフトウェア

#### 1.1.3.3. 管理的コントロール

管理的コントロールは、セキュリティーの人的要素を定義します。これは組織内のあらゆるレベルの職員や社員に関連するもので、誰がどのリソースや情報にアクセスするかを、次のような手段で決定します。

- トレーニングおよび認識の向上
- 災害準備および復旧計画

- 人員採用と分離の戦略
- 人員登録とアカウンティング

### 1.1.4. まとめ

これで、セキュリティーの起点、理由、および機能について理解したので、Red Hat Enterprise Linux に関する適切なアクションコースを簡単に判断できます。適切なストラテジーを計画および実装するには、どのような要素や条件がセキュリティーを構成するかを理解することが重要です。この情報は公式化でき、セキュリティープロセスの詳細に明確化されるため、パスは明確になります。

## 1.2. 脆弱性の評価

時間やリソースがあり、その気になれば、攻撃者はほとんどすべてのシステムに侵入できます。現在利用できるセキュリティーの手順と技術をすべて駆使しても、すべてのシステムを侵入から完全に保護できる訳ではありません。ルーターは、インターネットへのセキュアなゲートウェイを提供します。ファイアウォールは、ネットワークの境界を保護します。仮想プライベートネットワーク (VPN) では、データが、暗号化されているストリームで安全に通過できます。侵入検知システムは、悪意のある活動を警告します。しかし、これらの技術が成功するかどうかは、以下のような数多くの要因によって決まります。

- 技術の設定、監視、および保守を行うスタッフの専門知識
- サービスとカーネルのパッチ、および更新を迅速かつ効率的に行う能力
- ネットワーク上での警戒を常に怠らない担当者の能力

データシステムと各種技術が動的であることを考えると、企業リソースを保護するタスクは極めて複雑になる可能性もあります。この複雑さゆえに、使用するすべてのシステムの専門家を見つけることは、多くの場合困難になります。情報セキュリティーの多くの分野によく精通している人材を確保することはできても、多くの分野を専門とするスタッフを確保することは容易ではありません。これは、情報セキュリティーの各専門分野で、継続的な注意と重点が必要となるためです。情報セキュリティーは、常に変化しています。

### 1.2.1. 重要です。

エンタープライズネットワークを管理するとします。このようなネットワークは、一般的にオペレーティングシステム、アプリケーション、サーバー、ネットワークモニター、ファイアウォール、侵入検出システムなどで構成されます。ここで、それぞれの点について最新のことを考えてみましょう。現在のソフトウェアおよびネットワーク環境の複雑性を考慮して、不正使用とバグは一定です。ネットワーク全体のパッチや更新を最新の状態に保つと、異種システムを持つ大規模な組織では深刻なタスクになります。

現在のタスクと専門知識要件を組み合わせます。また、インシデントが発生したり、システムが侵害されたり、データが破損し、サービスが中断されることは防ぎます。

セキュリティー技術を強化し、システム、ネットワーク、およびデータの保護を支援するためには、ネゴシエーションを確認して、攻撃者のように見なされ、システムのセキュリティーを測定する必要があります。お客様のシステムおよびネットワークリソースに対する予防的な脆弱性アセスメントは、攻撃者が悪用する前に対処できる可能性がある問題を引き起こす可能性があります。

脆弱性アセスメントは、お使いのネットワークとシステムのセキュリティーに関する内部監査です。このアセスメントの結果により、ネットワークの機密性、完全性、および可用性が分かります（詳細は「」を参照 「セキュリティーの標準化」）。通常、脆弱性アセスメントは、対象システムとリソースに関する重要なデータを収集する調査フェーズから開始します。その後システム準備フェーズとなりま

す。基本的にこのフェーズでは、対象を絞り、すべての既知の脆弱性を調べます。準備フェーズが終わると報告フェーズになります。ここでは、調査結果が高中低のカテゴリーに分類され、ターゲットのセキュリティーを改善する（または脆弱性のリスクを軽減する）方法が説明されています。

たとえば、自宅の脆弱性アセスメントを実施することを想定してみましょう。まずは自宅のドアを点検し、各ドアが閉まっていて、かつ施錠されていることを確認します。また、すべての窓が完全に閉まっていて鍵が閉まっていることも確認します。これと同じ概念が、システム、ネットワーク、および電子データにも適用されます。悪意のあるユーザーはデータを盗んで、破壊します。悪意のあるユーザーが使用するツール、思考、動機に注目すると、彼らの行動にすばやく反応することが可能になります。

### 1.2.2. 評価とテストの定義

脆弱性アセスメントは、*外部からの視点* と *内部からの視点* の 2 種類に分類できます。

外部からの視点で脆弱性アセスメントを実施する場合は、外部からシステムに攻撃を試みます。所属企業の外部にあることで、攻撃者の視点が提供されます。一般にルーティング可能な IP アドレス、*DMZ* のシステム、ファイアウォールの外部インターフェースなど、攻撃者が目を付けるものを確認します。DMZ は「非武装地帯 (demilitarized zone)」を表し、企業のプライベート LAN などの信頼できる内部ネットワークと、公的なインターネットなどの信頼できない外部ネットワークの間にあるコンピューターまたは小さなサブネットワークに相当します。通常、DMZ には Web(HTTP)サーバー、FTP サーバー、SMTP(e-mail)サーバー、DNS サーバーなどのインターネットトラフィックにアクセスできるデバイスが含まれます。

内部からの視点で脆弱性アセスメントを実施する場合、実行者は内部関係者であり、信頼されるステータスにあることから、有利な立場になります。これは、ユーザーと、同じワーカーがシステムに一度ログインした観点です。プリントサーバー、ファイルサーバー、データベースなどのリソースを見ることができます。

これら 2 種類の脆弱性アセスメントには大きな違いがあります。社内のユーザーには、部外者が得られない多くの特権が付与されています。多くの組織では、侵入者を締め出すようにセキュリティーが構成されています。しかし、組織内の細かい部分 (部門内ファイアウォール、ユーザーレベルのアクセス制御および内部リソースに対する認証手順など) には、セキュリティー対策がほとんど行われていません。また、一般的にほとんどのシステムは社内にあるため、内部からの方がより多くのリソースを確認できます。いったん社外に移動すると、ステータスは信頼されない状態になります。通常、外部から利用できるシステムやリソースは、非常に限られたものになります。

脆弱性アセスメントと *侵入テスト* の違いを考えてみましょう。脆弱性アセスメントを、侵入テストの第一歩と捉えてください。このアセスメントで得られる情報は、その後のテストで使用します。アセスメントは抜け穴や潜在的な脆弱性を検査する目的で行われるのに対し、侵入テストでは調査結果を実際に使用する試みがなされます。

ネットワークインフラストラクチャーのアセスメントは動的なプロセスです。セキュリティー (情報セキュリティーおよび物理的なセキュリティー) は動的なものです。システムでアセスメントを実行すると、概要が示されており、誤検出と誤検出が示唆されます。誤検出は、実際には存在しない脆弱性をツールが検出することを指します。検出漏れは、実際の脆弱性が検出されないことを指します。

セキュリティー管理者の力量は、使用するツールとその管理者が有する知識で決まります。現在使用できるアセスメントツールのいずれかを選び、それらをシステムに対して実行すると、ほぼ間違いなく誤検出がいくつか見つかります。プログラム障害でもユーザーエラーでも、結果は同じです。ツールは、誤検出することもあれば、さらに悪い場合は、検出漏れをすることもあります。

脆弱性アセスメントと侵入テストの違いが定義されたところで、新たなベストプラクティスの一環として侵入テストを実施する前に、アセスメントの結果を注意深く確認し、検討してみましょう。

**警告**

実稼働システムで脆弱性を悪用する試みを行わないでください。システムおよびネットワークの生産性ならびに効率に悪影響を与える可能性があります。

以下の一覧で、脆弱性アセスメントを実施する利点をいくつか説明します。

- 情報セキュリティーに事前にフォーカスできる
- 攻撃者が発見する前に潜在的な不正使用を探します。
- システムを最新の状態に維持し、パッチを適用できる
- スタッフの成長と専門知識の開発を促す
- 経済的な損失や否定的な評判を減らす

#### 1.2.2.1. 方法論の確立

脆弱性アセスメントの方法論が確立されれば、脆弱性アセスメント用のツール選択に役立ちます。現時点では、事前定義の方法論や業界で承認された方法論はありませんが、一般常識やベストプラクティスを適切なガイドとして活用できます。

**「ターゲット」とは何を指していますか? 1台のサーバー、またはネットワーク全体およびネットワーク内にあるすべてのサーバーを確認しますか? 会社外ですか? それとも内部ですか?** この質問に対する回答は、選択したツールだけでなく、そのツールの使用方法を決定する際に重要です。

方法論の確立の詳細は、以下の Web サイトを参照してください。

- http://www.owasp.org/ 『The Open Web Application Security Project 』

### 1.2.3. ツールの評価

アセスメントは、情報収集ツールを使用することから始まります。ネットワーク全体を評価する際は、最初にレイアウトを描いて、稼働しているホストを把握します。ホストの場所を確認したら、それぞれのホストを個別に検査します。各ホストにフォーカスするには別のツールセットが必要になります。どのツールを使用すべきかを知っておくことは、脆弱性の発見において最も重要なステップになる可能性があります。

ライフタイムのあらゆる側面と同様に、同じジョブを実行するツールが多数あります。この概念は、脆弱性アセスメントの実行にも適用されます。オペレーティングシステム、アプリケーション、およびネットワーク（使用されるプロトコルに基づく）に固有のツールがあります。一部のツールは無料で、その他のツールではありません。一部のツールは直感的で使いやすいツールですが、他のツールには暗号化されず、文書化されていませんが、他のツールでは提供されない機能があります。

適切なツールを見つけることは困難なタスクであり、最後には経験が低い数になります。可能な場合は、テストラボを設定してできる限り多くのツールを試してください。各ツールの強みと強度を書き留めます。ツールに同梱されるドキュメントを確認してください（README ファイルや man ページなど）。詳細は、検索記事、ステップごとのガイド、またはインターネットのツールに固有するメーリングリストも併せて参照してください。

以下に説明しているツールは、利用可能なツールの小規模なサンプリングです。

#### 1.2.3.1. Nmap を使用したホストのスキャン

Nmap は、ネットワークのレイアウトを決定するために使用できる一般的なツールです。Nmap は長期間利用でき、情報を収集する際に最もよく使用されるツールです。オプションと使用方法についての詳細情報を記載した優れた man ページが含まれています。管理者はネットワーク上で Nmap を使用して、ホストシステムを見つけ、そのシステムでポートを開くことができます。

Nmap は、脆弱性アセスメントにおける最初のステップです。ネットワーク内のすべてのホストをマッピングして、Nmap が特定のホスト上で実行されているオペレーティングシステムの特定を試みるオプションを渡すこともできます。nmap は、セキュアなサービスを使用して未使用のサービスを制限するポリシーを確立するための適切な基盤です。

Nmap をインストールするには、root で **yum install nmap** コマンドを実行します。

##### 1.2.3.1.1. Nmap の使用

nmap は、シェルプロンプトから実行するには、**nmap** コマンドの後にスキャンするマシンのホスト名または IP アドレスを指定します。

```
nmap <host name>
```

たとえば、ホスト名 **foo.example.com** のマシンをスキャンするには、シェルプロンプトで以下を入力します。

```
~]$ nmap foo.example.com
```

基本的なスキャンの結果（ホストの場所およびその他のネットワーク条件により数分かかる場合があります）は、以下のようになります。

```
Interesting ports on foo.example.com:
Not shown: 1710 filtered ports
PORT    STATE  SERVICE
22/tcp  open   ssh
53/tcp  open   domain
80/tcp  open   http
113/tcp closed auth
```

nmap は、サービスのリッスンまたは待機中の最も一般的なネットワーク通信ポートをテストします。この知識は、不要なサービスや未使用のサービスを閉じる必要がある管理者に役立ちます。

Nmap の使用に関する詳細は、以下の URL の公式ホームページを参照してください。

http://www.insecure.org/

#### 1.2.3.2. Nessus

Nessus は完全なサービスセキュリティースキャナーです。Nessus のプラグインアーキテクチャーにより、ユーザーはシステムおよびネットワーク用にカスタマイズできます。スキャナーと同様に、Nessus は、それが依存する署名データベースと同様にのみ適しています。また、Nessus は頻繁に更新され、完全なレポート、ホストのスキャン、およびリアルタイムの脆弱性検索機能を利用できます。強力なツールで、Nessus として頻繁に更新されるツールであっても、誤検出と誤検出が生じる可能性があることに注意してください。

**注記**

Nessus クライアントおよびサーバーソフトウェアを使用するには、サブスクリプションが必要です。これは、この一般的なアプリケーションの使用に関心のあるユーザーへの参照として本書に含まれています。

Nessus の詳細は、以下の URL の公式 Web サイトを参照してください。

http://www.nessus.org/

#### 1.2.3.3. grikto

最も一般的なゲートウェイインターフェース(CGI)スクリプトスキャナーです。Fujikto は CGI の脆弱性をチェックするだけでなく、侵入検出システムを回避するため、悪用的な方法で実行されます。このドキュメントには詳細なドキュメントが含まれています。これは、プログラムの実行前に注意して確認する必要があります。CGI スクリプトを提供する Web サーバーがある場合は、これらのサーバーのセキュリティーを確認するのに便利なリソースを使用できます。

Frankkto の詳細は、以下の URL を参照してください。

http://cirt.net/nikto2

#### 1.2.3.4. 将来のニーズを予測

ターゲットとリソースに応じて、利用可能なツールが多数あります。ワイヤレスネットワーク、Noveell ネットワーク、Windows システム、Linux システムなど用のツールがあります。アセスメントの実行におけるもう 1 つの重要な部分には、物理的なセキュリティーの確認、人員スクリーニング、音声/PBX ネットワーク評価のレビューなどがあります。企業の物理構造の境界を *スキャン* してワイヤレスネットワークの脆弱性をスキャンするという概念は、必要であれば調査し、必要であれば評価に組み込む必要があるという概念です。脆弱性アセスメントを計画および実施する唯一の制限です。

## 1.3. セキュリティーに関する影響

優れたセキュリティー戦略を計画および実装するには、最初に、システムに危険を及ぼす可能性がある攻撃者が攻撃を招く問題の一部を最初に認識してください。

### 1.3.1. ネットワークセキュリティーへの脅威

ネットワークの以下の要素を設定する際に不適当なプラクティスが行われると、攻撃のリスクが増大する可能性があります。

#### 1.3.1.1. 安全ではないアーキテクチャー

間違った構成のネットワークは、未承認ユーザーの主要なエントリーポイントになります。信頼ベースでオープンなローカルネットワークを、安全性が非常に低いインターネットに対して無防備な状態にしておくことは、いかなる時間でも発生する **こと** はありませんが、この機会を悪用するような問題があります。

##### 1.3.1.1.1. ブロードキャストネットワーク

システム管理者は、セキュリティー計画においてネットワーキングハードウェアの重要性を見落としがちです。ハブやルーターなどの単純なハードウェアは、ブロードキャストまたはスイッチ以外のプリンシパルに依存します。つまり、あるノードがネットワーク経由で受信ノードにデータを送信するたびに、受信ノードがデータを受信して処理するまで、ハブまたはルーターがデータパケットのブロード

キャストを送信します。この方式は、外部侵入者やローカルホストの未認証ユーザーが仕掛けるアドレス解決プロトコル (*ARP*) およびメディアアクセスコントロール (*MAC*) アドレスの偽装に対して最も脆弱です。

#### 1.3.1.1.2. 集中化サーバー

ネットワーキングのもうひとつの落とし穴は、集中化されたコンピューティングの使用にあります。多くの企業では、一般的なコスト削減手段として、すべてのサービスを 1 台の強力なマシンに統合しています。集中化は、複数サーバーを設定するよりも管理が簡単で、コストを大幅に削減できるので便利です。ただし、集中化されたサーバーはネットワークにおける単一障害点となります。中央のサーバーが攻撃されると、ネットワークが完全に使用できなくなるか、データの不正操作や盗難が起きやすくなる可能性があります。このような状況では、中央サーバーは、ネットワーク全体へのアクセスを可能にすることになります。

### 1.3.2. サーバーセキュリティーへの脅威

サーバーには組織の重要情報が数多く保持されるため、サーバーのセキュリティーは、ネットワークのセキュリティーと同様に重要です。サーバーが危険にさらされると、攻撃者がいつでも攻撃や操作を行うことができる可能性があります。以下のセクションでは、主要な問題の一部を詳述します。

#### 1.3.2.1. 未使用のサービスとオープンポート

Red Hat Enterprise Linux 7 のフルインストールには、アプリケーションパッケージとライブラリーパッケージが 1000 個以上含まれています。ただし、サーバー管理者が、ディストリビューションに含まれるすべての個別パッケージをインストールすることはほとんどありません。代わりに、複数のサーバーアプリケーションを含むパッケージのベースインストールを行います。

システム管理者は、インストールに含まれるプログラムに注意を向けずにオペレーティングシステムをインストールしてしまうことがよくあります。これにより、不要なサービスがインストールされ、デフォルト設定でオンになっていることで、問題が発生する場合があります。これにより、管理者が認識せずに Telnet、DHCP、DNS などの不要なサービスがサーバーやワークステーションで実行し、その結果、サーバーへの不要なトラフィックが発生したり、攻撃者がシステムにパスする可能性があります。ポートのクローズおよび未使用 「サーバーセキュリティー」 のサービスの無効化に関する情報は、を参照してください。

#### 1.3.2.2. 管理の不注意

管理者がシステムにパッチを当てないことが、サーバーのセキュリティーに対する最大の脅威の 1 つになります。*SysAdmin*、*Audit, Audit, Network, Security Institute* (*SANS*)によると、コンピューターのセキュリティー脆弱性の主な原因は「セキュリティーを維持するための未実施の人数を割り当て、ジョブを実行できるようにトレーニングも時間も提供しないことです。これは、管理者の経験の少なさだけでなく、管理者の過信やモチベーションの低さなども原因となります。

管理者が、サーバーやワークステーションにパッチを当てることを忘れたり、システムのカーネルやネットワーク通信のログメッセージを見落とす場合もあります。その他にも、よく起こるケースとして、サービスのデフォルトパスワードや鍵を変更しないまま放置しておくことが挙げられます。たとえば、データベースにはデフォルトの管理パスワードが設定されているものがありますが、ここでは、システム管理者がインストール後すぐにデフォルトパスワードを変更することを、データベース開発者は想定しています。データベース管理者がこのパスワードを変更できないと、経験が低い攻撃者は、一般的に知られているデフォルトパスワードを使用して、データベースの管理者権限を得ることができます。この他に、管理者の不注意によりサーバーが危険にさらされる場合もあります。

#### 1.3.2.3. 本質的に安全ではないサービス

どんなに注意深い組織であっても、選択するネットワークサービスが本質的に安全でない限り、攻撃を

受けやすくなります。たとえば、多くのサービスは、信頼できるネットワークでの使用を想定して開発されますが、このサービスが (本質的に信頼できない) インターネットで利用可能になる時点で、この仮定は成立しなくなります。

安全ではないネットワークサービスの例として、暗号化されていないユーザー名とパスワードを認証時に要求するサービスが挙げられます。具体例としては、Telnet や FTP の 2 つがあげられます。パケット盗聴ソフトウェアがリモートユーザーとこのようなサービスの間のトラフィックを監視していれば、ユーザー名とパスワードは簡単に傍受される可能性があります。

また、基本的にこのようなサービスはセキュリティー業界で *中間者* 攻撃と呼ばれる攻撃の被害者になりやすくなります。この種の攻撃では、攻撃者はネットワーク上でクラッキングされたネームサーバーをトリックし、目的のサーバーではなくクラッカーのマシンを指定してネットワークトラフィックをリダイレクトします。サーバーへのリモートセッションを開くと、攻撃者のマシンはリモートサービスと無防備なユーザーとの間に秘密のパイプとして機能し、悪意のあるユーザーが情報をキャプチャーします。このようにして、攻撃者はサーバーやユーザーによる認識なしで管理パスワードや生データを収集できます。

安全ではないサービスの例としては、他にも NFS、NIS などのネットワークファイルシステムおよび情報サービスが挙げられます。このサービスは、LAN 利用を目的として開発されましたが、(リモートユーザー用の) WAN も対象に含まれるように拡張されました。NFS では、攻撃者が NFS 共有をマウントして含まれるものへアクセスしないように、デフォルトでは認証またはセキュリティーメカニズムが設定されていません。NIS も、プレーンテキストの ASCII または DBM (ASCII から派生) データベースに、パスワードやファイルパーミッションなど、ネットワーク上の全コンピューターへの周知が必要となる重要な情報を保持しています。このデータベースへのアクセスを取得する攻撃者は、管理者のアカウントなど、ネットワークのすべてのユーザーアカウントにアクセスできます。

デフォルトでは、Red Hat Enterprise Linux は、上記のサービスがすべて無効になっています。ただし、管理者は、このようなサービスを使用しないといけない場合があるため、注意して設定することが重要となります。安全な方法でサービスを設定する方法 「サーバーセキュリティー」 は、を参照してください。

### 1.3.3. ワークステーションおよびホーム PC セキュリティーに対する脅威

ワークステーションやホーム用 PC は、ネットワークやサーバーほど攻撃の対象ではありませんが、クレジットカード情報などの機密データが含まれるため、システム攻撃の対象となります。ワークステーションは、ユーザーの知識なしに選択でき、一連の攻撃で攻撃者が「スレーブ」マシンとして使用することもできます。このため、ユーザーはワークステーションの脆弱性を理解しておくと、オペレーティングシステムの再インストールや、深刻な場合はデータ盗難からの回復といった問題から免れることができます。

#### 1.3.3.1. 不適切なパスワード

攻撃者が最も簡単にシステムへのアクセスを得る方法の 1 つとして、パスワードが適切でないことが挙げられます。パスワードの作成時によくあるパイプが発生しないようにする方法は、を参照してください 「パスワードセキュリティー」。

#### 1.3.3.2. 脆弱なクライアントアプリケーション

管理者がサーバーに十分な安全対策を施し、パッチを当てている場合でも、リモートユーザーによるアクセスが安全であるわけではありません。たとえば、サーバーがパブリックネットワーク上で Telnet または FTP のサービスを提供している場合、攻撃者はネットワークを通過する際にプレーンテキストのユーザー名とパスワードを取得して、アカウント情報を使用してリモートユーザーのワークステーションにアクセスすることが可能です。

SSH などのセキュアなプロトコルを使用している場合であっても、クライアントアプリケーションを定期的に更新していないと、リモートユーザーは特定の攻撃を受けやすくなる可能性があります。たとえ

ば、v.1 SSH クライアントは、悪意のある SSH サーバーからの X 転送攻撃に対して脆弱です。クライアントがサーバーに接続すると、攻撃者はネットワーク上でクライアントによるキー入力やマウス操作をひそかに収集できます。この問題は v.2 SSH プロトコルで修正されましたが、ユーザーはどのアプリケーションにこのような脆弱性があるかを追跡し、必要に応じてアプリケーションを更新する必要があります。

「ワークステーションのセキュリティー」 コンピューターワークステーションの脆弱性を制限するために、管理者およびホームユーザーが行うべき手順を詳細に説明します。

## 1.4. 一般的な展開および A FIXESCKS

表1.1「一般的な展開」では、侵入者が組織のネットワークリソースにアクセスするために使用する最も一般的な不正使用とエントリーポイントの例を挙げて詳しく説明します。この一般的な不正使用では、それがどのように実行され、管理者がその攻撃からネットワークをどのように適切に保護できるかを理解していることが重要になります。

**表1.1 一般的な展開**

| 不正使用 | 説明 | 備考 |
|---|---|---|
| 空またはデフォルトのパスワード | 管理パスワードを空白のままにしたり、製品ベンダーが設定したデフォルトのパスワードをそのまま使用します。これは、ルーターやファイアウォールなどのハードウェアで最もよく見られますが、Linux で実行されるサービスにはデフォルトの管理者パスワードも含まれています（ただし、Red Hat Enterprise Linux には含まれていません）。 | 一般的に、ルーター、ファイアウォール、VPN、ネットワーク接続ストレージ (NAS) の機器など、ネットワークハードウェアに関連するものです。<br><br>多数のレガシーオペレーティングシステム、特にサービスをバンドルしたオペレーティングシステム (UNIX や Windows など) でよく見られます。<br>管理者が急いで特権ユーザーアカウントを作成したためにパスワードが空白のままになっていることがありますが、このような空白のパスワードは、このアカウントを発見した悪意のあるユーザーが利用できる絶好のエントリーポイントとなります。 |
| デフォルトの共有キー | セキュアなサービスでは、開発や評価テスト向けにデフォルトのセキュリティー鍵がパッケージ化されていることがあります。この鍵を変更せずにインターネットの実稼働環境に置いた場合は、同じデフォルトの鍵を持つ **すべての** ユーザーがその共有鍵のリソースや、そこにあるすべての機密情報にアクセスできるようになります。 | 無線アクセスポイントや、事前設定済みでセキュアなサーバー機器に最も多く見られます。 |

| 不正使用 | 説明 | 備考 |
| --- | --- | --- |
| IP スプーフィング | リモートマシンがローカルネットワークのノードのように動作し、サーバーに脆弱性を見つけるとバックドアプログラムまたはトロイの木馬をインストールして、ネットワークリソース全体へのコントロールを得ようとします。 | スプーフィングは、攻撃者が対象システムへの接続を調整するのに TCP/IP シーケンス番号を予測する必要があるため、かなり難しくなりますが、攻撃者がこのような脆弱性を実行できるツールがいくつかあります。<br><br>は **rsh**、*ソースベース* の認証技術を使用するサービス（**telnet**FTP など）を実行するターゲットシステムに依存します。これは、または **ssh** SSL/TLS で使用される PKI またはその他の形式の暗号化認証と比較すると推奨されません。 |
| 盗聴 | 2 つのノード間の接続を盗聴することにより、ネットワーク上のアクティブなノード間を行き交うデータを収集します。 | この種類の攻撃には大抵、Telnet、FTP、HTTP 転送などのプレーンテキストの転送プロトコルが使用されます。<br><br>リモートの攻撃者はこのような攻撃を実行するために LAN で攻撃されるシステムにアクセスできなければなりません。通常は、攻撃者が LAN 上のシステムを危険にさらすためにアクティブな攻撃（IP スプーフィングや中間者攻撃など）を使用しています。<br><br>パスワードのなりすましに対する防護策としては、暗号化鍵交換、ワンタイムパスワード、または暗号化された認証によるサービス使用が挙げられます。通信中は強力な暗号化を実施することをお勧めします。 |

| 不正使用 | 説明 | 備考 |
| --- | --- | --- |
| サービスの脆弱性 | 攻撃者はインターネットで実行しているサービスの欠陥や抜け穴を見つけます。攻撃者がこの脆弱性を利用する場合は、システム全体と格納されているデータを攻撃するだけでなく、ネットワーク上の他のシステムも攻撃する可能性があります。 | CGI などの HTTP ベースのサービスは、リモートのコマンド実行やインタラクティブなシェルアクセスに対しても脆弱です。HTTP サービスが「nobody」などの権限のないユーザーとして実行している場合でも、設定ファイルやネットワークマップなどの情報が読み取られる可能性があります。または、攻撃者がサービス拒否攻撃を開始して、システムのリソースを浪費させたり、他のユーザーが利用できないようにする可能性もあります。<br><br>開発時およびテスト時には気が付かない脆弱性がサービスに含まれることがあります。(アプリケーションのメモリーバッファー領域をあふれさせ、任意のコマンドを実行できるようなインタラクティブなコマンドプロンプトを攻撃者に提供するように、攻撃者が任意の値を使用してサービスをクラッシュさせる *バッファーオーバーフロー*などの) 脆弱性は、完全な管理コントロールを攻撃者に与えるものとなる可能性があります。<br><br>管理者は、root 権限でサービスが実行されないようにし、ベンダー、または CERT、CVE などのセキュリティー組織がアプリケーション用のパッチやエラータ更新を提供していないかを常に注意する必要があります。 |
| アプリケーションの脆弱性 | 攻撃者は、デスクトップやワークステーションのアプリケーション（電子メールクライアントなど）に欠陥を見つけ出し、任意のコードを実行したり、将来のシステム侵害のためにトリックの木形を移植したり、システムをクラッシュしたりする可能性があります。攻撃を受けたワークステーションがネットワークの残りの部分に対して管理特権を持っている場合は、さらなる不正使用が起こる可能性があります。 | ワークステーションとデスクトップは、ユーザーが侵害を防いだり検知するための専門知識や経験を持たないため、不正使用の対象になりやすくなります。認証されていないソフトウェアをインストールしたり、要求していないメールの添付ファイルを開く際には、それに伴うリスクについて個々に通知することが必須です。<br><br>電子メールクライアントソフトウェアが添付ファイルを自動的に開いたり、実行したりしないようにするといった、予防手段を取ることが可能です。さらに、Red Hat Network またはその他のシステム管理サービスを使用したワークステーションソフトウェアの自動更新により、マルチシートのセキュリティーデプロイメントの負担を軽減できます。 |

| 不正使用 | 説明 | 備考 |
| --- | --- | --- |
| サービス拒否 (DoS)攻撃 | 攻撃者のまたは攻撃者のグループは、権限のないパケットをターゲットホスト（サーバー、ルーター、ワークステーションのいずれか）に送信することで、組織のネットワークまたはサーバーのリソースに対して攻撃を攻撃します。これにより、正当なユーザーがリソースを使用できなくなります。 | 通常ソースパケットは、真の攻撃元を調査するのが難しくなるよう、偽装 (または再ブロードキャスト）されています。<br><br>を使用した Ingress フィルタリング(IETF rfc2267) **iptables** およびネットワーク侵入検出システム **snort** における進捗 |

## 1.5. セキュリティー更新

セキュリティー上の脆弱性が検出されると、セキュリティーリスクを制限するために影響を受けるソフトウェアを更新する必要があります。現在対応している Red Hat Enterprise Linux ディストリビューション内のパッケージにソフトウェアの一部である場合、Red Hat は、できるだけ早く脆弱性を修正する更新パッケージをリリースするよう努めています。多くの場合、特定のセキュリティーの悪用に関するアナウンスにはパッチ（または問題を修正するソースコード）が含まれます。このパッチは Red Hat Enterprise Linux パッケージに適用され、エラータ更新としてテストおよびリリースされます。ただし、通知にパッチが含まれていない場合には、開発者が最初にソフトウェアの保守管理者が機能し、問題を修正します。問題が修正されると、パッケージはエラータ更新としてテストされ、リリースされます。

エラータの更新がシステムで使用されるソフトウェア用にリリースされている場合は、影響を受けるパッケージをできるだけ早く更新して、システムが脆弱である可能性がある期間を最小限に抑えるように強く推奨します。

### 1.5.1. パッケージの更新

システムでソフトウェアを更新する場合は、信頼できるソースから更新をダウンロードすることが重要です。攻撃者は、問題の修正予定と同じバージョン番号を使用してパッケージを簡単に再構築できますが、別のセキュリティー悪用を使用してパッケージをインターネットに解放できます。その場合、元の RPM に対してファイルを検証するなどのセキュリティー対策を使用しても、悪用が検出されません。そのため、Red Hat からなど、信頼できるソースから RPM のみをダウンロードすることが非常に重要です。パッケージの署名を確認してその整合性を検証することが非常に重要です。

**注記**

Red Hat Enterprise Linux には、更新が利用可能な場合に表示されるアラートを表示する便利なパネルアイコンが含まれています。

### 1.5.2. 署名済みパッケージの確認

Red Hat Enterprise Linux パッケージはすべて、Red Hat *GPG* キーで署名されています。GPG は GNU Privacy Guard(GnuPG)を表し、配布ファイルの信頼性を確保するために使用されます。たとえば、秘密鍵（秘密鍵）は、公開鍵がロックされ、パッケージを検証する際にパッケージがロックされます。RPM の検証中に Red Hat Enterprise Linux によって配布される公開鍵が秘密鍵と一致しない場合は、パッケージが変更されているため、信頼できない可能性があります。

Red Hat Enterprise Linux 6 の RPM ユーティリティーは、インストールする前に、RPM パッケージの GPG 署名を自動的に検証しようとします。Red Hat GPG キーがインストールされていない場合は、Red Hat インストール CD-ROM や DVD などの安全な静的な場所からインストールします。

ディスクがにマウントされていることを前提として **/mnt/cdrom**、root ユーザーとして次のコマンドを使用して、キー *リング*（システム上の信頼済み鍵のデータベース）にインポートします。

```
~]# rpm --import /mnt/cdrom/RPM-GPG-KEY
```

Red Hat GPG キーは **/etc/pki/rpm-gpg/** ディレクトリーに置かれています。

RPM 検証用にインストールされた鍵の一覧を表示するには、以下のコマンドを実行します。

```
~]# rpm -qa gpg-pubkey*
gpg-pubkey-db42a60e-37ea5438
```

特定の鍵の詳細を表示するには、以下の例のように、**rpm -qi** コマンドのその後に直前のコマンドの出力を表示します。

```
~]# rpm -qi gpg-pubkey-db42a60e-37ea5438
Name        : gpg-pubkey                   Relocations: (not relocatable)
Version     : 2fa658e0                        Vendor: (none)
Release     : 45700c69                     Build Date: Fri 07 Oct 2011 02:04:51 PM CEST
Install Date: Fri 07 Oct 2011 02:04:51 PM CEST      Build Host: localhost
Group       : Public Keys                  Source RPM: (none)
[output truncated]
```

RPM ファイルの署名を確認してから元のパッケージから変更されていないことを確認することが非常に重要です。ダウンロードしたパッケージをすべて一度に確認するには、以下のコマンドを実行します。

```
~]# rpm -K /root/updates/*.rpm
alsa-lib-1.0.22-3.el6.x86_64.rpm: rsa sha1 (md5) pgp md5 OK
alsa-utils-1.0.21-3.el6.x86_64.rpm: rsa sha1 (md5) pgp md5 OK
aspell-0.60.6-12.el6.x86_64.rpm: rsa sha1 (md5) pgp md5 OK
```

GPG キーが正常に検証されると、各パッケージでコマンドが返され **gpg OK**ます。そうでない場合は、正しい Red Hat 公開鍵を使用していることを確認し、コンテンツのソースを確認してください。GPG 検証に合格しないパッケージは、サードパーティーによって変更されている可能性があるため、インストールしないでください。

GPG キーを確認し、エラータレポートに関連するパッケージをすべてダウンロードしたら、シェルプロンプトで root としてパッケージをインストールします。

Yum ユーティリティーを使用して署名済みパッケージを検証できます。Yum は、GPG 署名パッケージで GPG 署名検証をすべてのパッケージリポジトリー（パッケージソース）、または個々のリポジトリーに対して有効にすることで、セキュアなパッケージ管理を提供します。署名の検証が有効になっていると、Yum は、そのリポジトリーの正しいキーで GPG 署名されていないパッケージのインストールを拒否します。つまり、使用中のシステムにダウンロードしてインストールする RPM パッケージが Red Hat などの信頼できるソースからのものであり、転送中に変更されていないことを保証できます。

Yum でパッケージのインストールまたは更新時に GPG 署名の自動検証を有効にするには、**/etc/yum.conf** ファイルの **[main]** セクションで以下のオプションが定義されていることを確認します。

```
gpgcheck=1
```

### 1.5.3. 署名済みパッケージのインストール

ほとんどのパッケージのインストールは、root で以下のコマンドを実行すると、（カーネルパッケージを除く）安全に実行できます。

```
rpm -Uvh <package>…
```

たとえば、ディレクトリー下のという名前の新しいディレクトリーにすべてのパッケージをインストールするには **updates**/、以下のコマンドを **/tmp** 実行します。

```
~]# rpm -Uvh /tmp/updates/*.rpm
Preparing...                ########################################### [100%]
   1:alsa-lib               ########################################### [ 33%]
   2:alsa-utils             ########################################### [ 67%]
   3:aspell                 ########################################### [100%]
```

カーネルパッケージの場合、root で以下の形式でコマンドを使用します。

```
rpm -ivh <kernel-package>
```

たとえば、をインストールするには kernel-2.6.32-220.el6.x86_64.rpm、シェルプロンプトで以下を入力します。

```
~]# rpm -ivh /tmp/updates/kernel-2.6.32-220.el6.x86_64.rpm
Preparing...                ########################################### [100%]
   1:kernel                 ########################################### [100%]
```

新しいカーネルを使用してマシンを安全に再起動すると、以下のコマンドを使用して古いカーネルを削除できます。

```
rpm -e <old-kernel-package>
```

たとえば、を削除するには、以下のコマンドを kernel-2.6.32-206.el6.x86_64入力します。

```
~]# rpm -e kernel-2.6.32-206.el6.x86_64
```

Yum でパッケージをインストールするには、root で以下のコマンドを実行します。

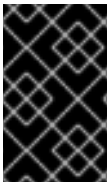
```
~]# yum install kernel-2.6.32-220.el6.x86_64.rpm
```

Yum でローカルパッケージをインストールするには、root で以下のコマンドを実行します。

```
~]# yum localinstall /root/updates/emacs-23.1-21.el6_2.3.x86_64.rpm
```

**注記**

古いカーネルを削除する必要はありません。デフォルトのブートローダー GRUB では、複数のカーネルをインストールし、ブート時にメニューから選択できます。



**重要**

セキュリティーエラータをインストールする前に、エラータに含まれる特別な指示を読み込んで、適切に実行してください。エラータ更新による変更 「変更の適用」 の適用に関する一般的な手順は、を参照してください。

### 1.5.4. 変更の適用

セキュリティーエラータおよび更新をダウンロードしてインストールしたら、古いソフトウェアの使用を停止して、新しいソフトウェアの使用を開始することが重要です。この方法は、更新されたソフトウェアの種類によって異なります。以下の一覧は、ソフトウェアの一般的なカテゴリーを項目化し、パッケージのアップグレード後に更新されたバージョンを使用する手順を説明します。

**注記**

通常、システムを再起動することは、最新バージョンのソフトウェアパッケージが使用されるようにする最も簡単な方法です。ただし、このオプションは常に必要な訳ではなく、システム管理者が利用できるようになります。

**アプリケーション**

ユーザー空間のアプリケーションは、システムユーザーが開始できるプログラムです。通常、このようなアプリケーションは、ユーザー、スクリプト、または自動化タスクユーティリティーが起動する場合にのみ使用され、長期間保持されません。

このようなユーザー空間アプリケーションが更新されたら、システム上のアプリケーションのインスタンスをすべて停止し、更新されたバージョンを使用するようにプログラムを再度起動します。

**カーネル**

カーネルは、Red Hat Enterprise Linux オペレーティングシステムの中核となるソフトウェアコンポーネントです。メモリー、プロセッサー、およびPeripherals へのアクセスを管理し、すべてのタスクをスケジュールします。

その一元的な役割が原因で、コンピューターを停止せずにカーネルを再起動することはできません。したがって、システムを再起動するまで、カーネルの更新バージョンは使用できません。

**Shared Libraries**

共有ライブラリーは、などのコードの単位です。これは **glibc**、多数のアプリケーションやサービスにより使用されます。共有ライブラリーを使用するアプリケーションは、通常、アプリケーションを初期化するときに共有コードを読み込みます。そのため、更新されたライブラリーを使用するアプリケーションは停止して再起動する必要があります。

特定のライブラリーにリンクしている実行中のアプリケーションを確認するには、**lsof** コマンドを使用します。

```
lsof <path>
```

たとえば、実行中のアプリケーションリンクを **libwrap.so** ライブラリーにリンクするには、以下を入力します。

```
~]# lsof /lib64/libwrap.so*
COMMAND   PID     USER  FD   TYPE DEVICE SIZE/OFF   NODE NAME
sshd    13600 root mem    REG  253,0    43256 400501 /lib64/libwrap.so.0.7.6
sshd    13603 juan mem    REG  253,0    43256 400501 /lib64/libwrap.so.0.7.6
```

```
gnome-set 14898 juan mem    REG  253,0    43256 400501 /lib64/libwrap.so.0.7.6
metacity  14925 juan mem    REG  253,0    43256 400501 /lib64/libwrap.so.0.7.6
[output truncated]
```

このコマンドは、ホストアクセス制御に TCP ラッパーを使用する実行中のプログラムの一覧を表示します。したがって、**tcp_wrappers** パッケージが更新された場合、一覧表示されるプログラムは停止し、再起動する必要があります。

### SysV サービス

SysV サービスは、ブートプロセス中に起動される永続サーバープログラムです。SysV サービスの例に **sshd**は、、**vsftpd**、およびがあり **xinetd**ます。

これらのプログラムは通常、マシンを起動している限りメモリーに保持されるため、パッケージのアップグレード後に、更新された各 SysV サービスを停止し、再起動する必要があります。これは、**Services Configuration Tool** を使用して行うか、または root シェルプロンプトにログインして **/sbin/service** コマンドを実行します。

```
/sbin/service <service-name> restart
```

*<service-name>* をなどのサービスの名前に置き換え **sshd**ます。

### xinetd services

**xinetd** スーパーサービスによって制御されるサービスは、アクティブな接続がある場合にのみ実行されます。で制御されるサービスの例には、Telnet、IMAP、および POP3 が **xinetd** 含まれます。

これらのサービスの新規インスタンスは、新しいリクエストを受信する **xinetd** たびに起動されるため、アップグレード後に発生する接続は更新されたソフトウェアによって処理されます。ただし、制御されたサービスのアップグレード時に **xinetd** アクティブな接続がある場合は、古いバージョンのソフトウェアによって処理されます。

特定の **xinetd** 制御されたサービスの古いインスタンスを強制終了するには、サービスのパッケージをアップグレードしてから、現在実行中のすべてのプロセスを停止します。プロセスが実行中かどうかを確認するには、**ps** または **pgrep** コマンドを使用して、サービスの現在のインスタンス **kill killall** を停止します。

たとえば、セキュリティーエラータパッケージがリリースされて **imap** いる場合は、パッケージをアップグレードし、root で以下のコマンドをシェルプロンプトに入力します。

```
~]# pgrep -l imap
1439 imapd
1788 imapd
1793 imapd
```

このコマンドは、アクティブな IMAP セッションをすべて返します。個々のセッションを終了するには、root で以下のコマンドを実行します。

```
kill <PID>
```

このセッションの終了に失敗した場合は、代わりに以下のコマンドを実行します。

```
kill -9 <PID>
```

この例では、*<PID>* を IMAP セッションのプロセス ID 番号（ **pgrep -l** コマンドの 2 番目のコラムにあるもの）に置き換えます。

アクティブな IMAP セッションをすべて強制終了するには、以下のコマンドを実行します。

```
~]# killall imapd
```

[1] http://law.jrank.org/pages/3791/Kevin-Mitnick-Case-1999.html

[2] http://www.livinginternet.com/i/ia_hackers_levin.htm

# 第2章 ネットワークのセキュリティー保護

## 2.1. ワークステーションのセキュリティー

Linux 環境のセキュリティー保護はワークステーションで始まります。個人マシンのロックダウンまたはエンタープライズシステムのセキュリティーを保護するかどうかは、個々のコンピューターからサウンドセキュリティーポリシーが開始されます。コンピューターネットワークは、最も弱いノードのセキュリティーレベルと同等です。

### 2.1.1. ワークステーションセキュリティーの評価

Red Hat Enterprise Linux ワークステーションのセキュリティーを評価する場合は、以下を検討してください。

- **BIOS and Boot Loader Security**- 承認されていないユーザーはマシンに物理的にアクセスして、パスワードなしで単一ユーザーまたはレスキューモードで起動できますか。
- **Password Security** - How secure the user account password on the machine?
- **管理** 的コントロール - システムに関するアカウントと管理制御の量
- **利用可能なネットワークサービス：** ネットワークからの要求をリッスンしているサービスと、それらが全く実行しているか？
- **個人ファイアウォール** - 必要な場合はどの種類のファイアウォールが必要ですか？
- **セキュリティー強化通信ツール** - ワークステーション間の通信にどのツールを使用するか、どのツールもワークステーション間の通信に使用すべきか。

### 2.1.2. BIOS およびブートローダーのセキュリティー

BIOS (もしくは BIOS に相当するもの) およびブートローダーをパスワードで保護することで、システムに物理的にアクセス可能な未承認ユーザーがリムーバブルメディアを使用して起動したり、シングルユーザーモードで root 権限を取得することを防ぐことができます。このような攻撃に対するセキュリティー対策は、ワークステーションの情報の機密性とマシンの場所によって異なります。

たとえば、見本市で使用されていて機密情報を含んでいないマシンでは、このような攻撃を防ぐことが重要ではないかもしれません。ただし、同じ見本市で、企業ネットワークに暗号化されていない SSH 秘密鍵を持つ従業員のノートパソコンが同じ見本市で無人ままになっている場合は、組織全体に対して大きなセキュリティー侵害が生じる可能性があります。

一方で、ワークステーションが権限のあるユーザーもしくは信頼できるユーザーのみがアクセスできる場所に置かれてるのであれば、BIOS もしくはブートローダーの安全確保は必要ない可能性もあります。

#### 2.1.2.1. BIOS パスワード

コンピューターの BIOS をパスワードで保護する主な 2 つの理由を以下に示します。[3]:

1. **BIOS 設定への変更を防止** する - 侵入者が BIOS にアクセスできる場合は、CD-ROM またはフラッシュドライブから起動するように設定できます。これにより、侵入者がレスキューモードまたはシングルユーザーモードに入り、システムで任意のプロセスを開始したり、機密データをコピーできるようになります。

2. **システムの起動** を防止する - BIOS の中には起動プロセスをパスワードで保護できるものもあります。これを有効にすると、攻撃者は BIOS がブートローダーを開始する前にパスワード入力を求められます。

BIOS パスワードの設定方法はコンピューターメーカーによって異なるため、具体的な手順はコンピューターのマニュアルを参照してください。

BIOS パスワードを忘れた場合は、マザーボードのジャンパーでリセットするか、CMOS バッテリーを外します。このため、可能な場合はコンピューターのケースをロックすることが推奨されます。ただし、CMOS バッテリーを外す前にコンピューターもしくはマザーボードのマニュアルを参照してください。

#### 2.1.2.1.1. x86 以外のプラットフォームのセキュリティー保護

その他のアーキテクチャーは、異なるプログラムを使用して、x86 システムの BIOS とほぼ同等の低レベルのタスクを実行します。たとえば、Intel® Itanium™ コンピューターは*EFI (Extendsible Firmware Interface* )シェルを使用します。

他のアーキテクチャーで BIOS のようなプログラムをパスワードで保護する方法は、製造元の手順を参照してください。

### 2.1.2.2. ブートローダーのパスワード

Linux ブートローダーをパスワードで保護する主な理由を以下に示します。

1. **シングルユーザーモードへのアクセスを防止する -** 攻撃者がシングルユーザーモードでシステムを起動できる場合は、root パスワードを求められることなく root として自動的にログインされます。

**警告**

**/etc/sysconfig/init** ファイルで ***SINGLE*** パラメーターを編集してパスワードを使用してシングルユーザーモードへのアクセスを保護することは推奨されません。攻撃者は、GRUB のカーネルコマンドラインでカスタム初期コマンド（ ***init=*** パラメーターを使用して）を指定することで、パスワードをバイパスできます。で指定されている GRUB ブートローダーのパスワード保護が推奨され 「GRUB のパスワード保護」 ます。

2. **GRUB コンソールへのアクセスを防止** する - マシンが GRUB をブートローダーとして使用する場合、攻撃者は GRUB エディターインターフェースを使用して設定を変更したり、**cat** コマンドを使用して情報を収集したりできます。

3. 非**セキュアなオペレーティングシステムへのアクセスを防止する -** デュアルブートシステムの場合、攻撃者は起動時にオペレーティングシステムを選択でき、アクセス制御やファイルのパーミッションを無視します。

Red Hat Enterprise Linux 6 には、x86 プラットフォームに GRUB ブートローダーが含まれています。GRUB の詳細は、『 『Red Hat Enterprise Linux インストールガイド』を参照してください 』。

#### 2.1.2.2.1. GRUB のパスワード保護

に記載されている最初の 2 つの問題に対応するように GRUB を設定 「ブートローダーのパスワード」するには、設定ファイルに password ディレクティブを追加します。これを行うには、まず強固なパスワードを選択し、シェルを開き、root でログインしてから以下のコマンドを入力します。

```
/sbin/grub-md5-crypt
```

プロンプトが表示されたら、GRUB パスワードを入力し、を押し **Enter**ます。これは、パスワードの MD5 ハッシュを返します。

次に、GRUB 設定ファイルを編集し **/boot/grub/grub.conf**ます。ドキュメントのメインセクションにあるファイル **timeout** を開き、以下の行を追加します。

```
password --md5 <password-hash>
```

*<password-hash>* をによって返された値に置き換えます。 **/sbin/grub-md5-crypt**[4].

次にシステムを起動すると、GRUB メニューは、最初に **p** GRUB パスワードを押さずにエディターまたはコマンドラインインターフェースにアクセスできなくなります。

ただし、このソリューションにより、攻撃者はデュアルブート環境のセキュアでないオペレーティングシステムで起動することを防ぐことはできません。そのためには、**/boot/grub/grub.conf** ファイルの異なる部分を編集する必要があります。

セキュアにするオペレーティングシステムの **title** 行を探し、その下の **lock** ディレクティブの行を追加します。

DOS システムの場合、スタンザは以下のようになります。

```
title DOS
 lock
```

**警告**

この方法が適切に機能するには、**/boot/grub/grub.conf** ファイルのメインセクションに **password** 行が存在する必要があります。それ以外の場合は、攻撃者は GRUB エディターインターフェースにアクセスし、ロック行を削除できます。

特定のカーネルまたはオペレーティングシステムに別のパスワードを作成するには、スタンザに **lock** 行を追加し、パスワード行を追加します。

一意のパスワードで保護される各スタンザは、以下の例のような行で開始する必要があります。

```
title DOS
 lock
 password --md5 <password-hash>
```

#### 2.1.2.2.2. 対話的な起動の無効化

ブートシーケンスの開始時に **I** キーを押して、システムを対話的に起動できます。対話式の起動中に、

システムは各サービスを起動するように求められます。ただし、これにより、システムに物理的なアクセスを取得する攻撃者は、セキュリティー関連のサービスを無効にし、システムへのアクセスを取得する可能性があります。

ユーザーがシステムを対話的に起動しないようにするには、root で **/etc/sysconfig/init** ファイルの ***PROMPT*** パラメーターを無効にします。

```
PROMPT=no
```

### 2.1.3. パスワードセキュリティー

パスワードは、Red Hat Enterprise Linux がユーザーのアイデンティティーを検証するために使用する主な方法です。このため、パスワードセキュリティーが、ユーザー、ワークステーション、およびネットワークを保護するのに非常に重要です。

セキュリティー上の理由から、インストールプログラムは、システムが *Secure Hash Algorithm 512(***SHA512** )とシャドウパスワードを使用するように設定します。これらの設定は変更しないことが強く推奨されます。

インストール時にシャドウパスワードを選択すると、すべてのパスワードが全読み取り **/etc/passwd** ファイルの一方向ハッシュとして保存されるため、システムがオフラインパスワードクラッキング攻撃に対して脆弱になります。侵入者が通常ユーザーとしてマシンにアクセスすることができる場合は、**/etc/passwd** ファイルを自分のマシンにコピーして、任意の数のパスワードクラッキングプログラムを実行することができます。ファイルにセキュリティー保護されていないパスワードがある場合は、パスワード攻撃者が検出する前に時間がかかります。

シャドウパスワードは **/etc/shadow**、ファイルにパスワードハッシュを保存することで、この種の攻撃を防ぎます。これは、root ユーザーのみが読み取り可能です。

これにより、攻撃者は SSH や FTP などのマシン上のネットワークサービスにログインして、パスワードのクラッキングをリモートで試行することが強制されます。このブルートフォース攻撃の種別はかなり遅く、数百の失敗したログイン試行がシステムファイルに書き込まれるため、明らかな証跡が残されます。当然ながら、攻撃者が弱いパスワードを持つシステム上で攻撃を開始すると、クラッカーはログファイルを盗んで編集し、追跡内容をカバーするためにログファイルを編集する可能性があります。

フォーマットやストレージの考慮事項に加えて、コンテンツの問題も挙げられます。パスワードクラッキング攻撃に対してユーザーがアカウントを保護することが強固なパスワードを生じさせることです。

#### 2.1.3.1. 強固なパスワードの作成

セキュアなパスワードを作成する場合、ユーザーは長いパスワードが短いパスワードや複雑なパスワードよりも強力なことを認識する必要があります。数字、特殊文字、大文字を含む場合でも、8 文字のパスワードのみを作成することは良いでしょう。john The Ripper などのパスワードクラッキングツールは、このようなパスワードを破損するように最適化されています。これは、人によって記憶が困難です。

情報交換では、エントロピーはランダムな変数と関連付けられ、ビットで示されます。エントロピーの値が大きいほど、パスワードのセキュリティーレベルは高くなります。NIST SP 800-63-1 によると、一般的に選択されるパスワードは、少なくとも 10 ビットのエントロピーを持つ必要があるディクショナリーにないパスワードです。このため、4 つのランダムな単語で構成されるパスワードには、約 40 ビットのエントロピーが含まれています。セキュリティーを強化する複数の単語で構成される長いパスワードは *パスフレーズ*とも呼ばれます。以下に例を示します。

```
randomword1 randomword2 randomword3 randomword4
```

システムが大文字、数字、または特殊文字を使用する場合は、たとえば最初の文字を大文字に変更して「**1!**」を追加することで、上記の推奨事項に続くパスフレーズを簡単に変更できます。このような変更により、パスフレーズのセキュリティーが大幅に強化される **わけ** ではないことに注意してください。

セキュアなパスワードを作成する方法は複数ありますが、常に以下の不適切なプラクティスを回避してください。

- 1 つの辞書単語、言語の単語、反転単語、または数字のみを使用します。
- パスワードまたはパスフレーズに 10 文字未満を使用します。
- キーボードレイアウトのキーシーケンスの使用
- パスワードを書き留めます。
- 写真日、匿名、ファミリーメンバー名、ペット名などのパスワードで個人情報を使用します。
- 複数のマシンで同じパスフレーズまたはパスワードを使用する。

セキュアなパスワードの作成は必須ですが、特に組織内のシステム管理者には適切に管理することが重要です。次のセクションでは、組織内でユーザーパスワードを作成して管理するのに適したプラクティスを説明します。

### 2.1.4. 組織内におけるユーザーパスワードの作成

組織にユーザーが多数ある場合、システム管理者には、適切なパスワードの使用を強制するために 2 つの基本的なオプションを利用できます。ユーザーのパスワードを作成したり、ユーザーが自身のパスワードを作成したり、パスワードが妥当な品質であることを検証したりできます。

ユーザーのパスワードを作成すると、パスワードは良好ですが、組織が大きくなると深刻なタスクになります。また、パスワードを書き込むリスクが高まります。

このため、システム管理者は、ユーザーによるパスワード作成を希望しますが、パスワードが適切であることをアクティブに確認します。場合によっては、パスワードエージングによってパスワードを定期的に変更します。

#### 2.1.4.1. 強固なパスワードの強制

ネットワークを侵入から保護するには、システム管理者が、組織内で使用されているパスワードが強固であることを検証することを推奨します。パスワードの作成または変更が求められた場合、コマンドラインアプリケーションを使用できます。これは*PAM( Pluggable Authentication Modules* )認識しているので、パスワードが短すぎるか **passwd**、または簡単に解除できるかどうかを確認します。このチェックは、**pam_cracklib.so** PAM モジュールを使用して実行します。Red Hat Enterprise Linux では、**pam_cracklib** PAM モジュールを使用して、一連のルールに対してパスワードの強度を確認することができます。ユーザーのログイン用にカスタムルールセットを設定するために、**/etc/pam.d/passwd** ファイルの **password** コンポーネントにある他の PAM モジュールとともにスタックできます。のルーチン **pam_cracklib**は 2 つの部分から構成されます。提供されたパスワードがディクショナリーで見つかったかどうかをチェックします。これがディクショナリーでない場合は、多くのチェックで続行されます。これらのチェックの一覧は、**pam_cracklib(8)** man ページを参照してください。

> **例2.1パスワードの強度チェックの設定 pam_cracklib**
>
> 最小長 8 文字のパスワード（すべて 4 文字を含む）を要求するには、**/etc/pam.d/passwd** ファイルの **password** セクションに以下の行を追加します。
>
> ```
> password   required     pam_cracklib.so retry=3 minlen=8 minclass=4
> ```

パスワードの strength-check で連続する文字または繰り返しの文字を設定するには、**/etc/pam.d/passwd** ファイルの **password** セクションに以下の行を追加します。

```
password   required     pam_cracklib.so retry=3 maxsequence=3 maxrepeat=3
```

この例では、「abcd」や「1234」などの連続する 3 文字を超えるパスワードを含めることはできません。また、同じ連続する文字数は 3 に制限されます。

**注記**

これらのチェックは root ユーザーに対して実行されないため、警告メッセージが表示されても、通常のユーザーのパスワードを設定できます。

PAM はカスタマイズできるため、パスワード整合性チェッカー（http://www.openwall.com/passwdqc/ から利用可能）や新しいモジュールの作成 **pam_passwdqc** などを追加できます。利用可能な PAM モジュールの一覧は、http://uw714doc.sco.com/en/SEC_pam/pam-6.html を参照してください。PAM の詳細は、『『シングルサインオンおよびスマートカードの管理』を参照して』ください。

パスワードチェックでは、作成時に実行されるパスワードを確認しても、パスワードクラッキングプログラムを実行するのと同様に、不適切なパスワードが検出されません。

多くのパスワードクラッキングプログラムは、Red Hat Enterprise Linux で実行できますが、オペレーティングシステムには同梱されていません。以下は、パスワードクラッキングプログラムの最も一般的な一覧です。

- **john The Ripper** - 高速で柔軟なパスワードクラッキングプログラムです。複数の単語一覧を使用でき、パスワードクラッキングをブルートフォースできます。これは http://www.openwall.com/john/ でオンラインで利用できます。
- crack- 最もよく知られているパスワードクラッキングソフトウェアである **Crack** も非常に高速ですが、Ripper として簡単に使用**できません**。
- **Slurpie** - Slurpie **は、Ripper と Crack と 似** ていますが、複数のコンピューターを同時に実行するように設計されており、分散パスワードクラッキング攻撃が発生します。このツールと、http://www.ussrback.com/distributed.htm でオンラインの分散型攻撃セキュリティー評価ツールが多数あります。

**警告**

組織内のパスワードをクラッキングする前に、必ず書き込みを承認してください。

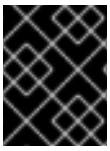
#### 2.1.4.2. passphrases

パスフレーズとパスワードは、現在のシステムの多くでセキュリティーの基盤となります。ただし、多くのシステムでは、biometrics や 2 要素認証などの技術がまだメインストリームになっていません。パスワードを使用してシステムのセキュリティーを保護する場合は、パスフレーズの使用を考慮する必要があります。パスフレーズはパスワードよりも長く、数字やシンボルなどの標準文字で実装されても、パスワードよりも優れた保護を提供します。

#### 2.1.4.3. パスワードの集約

パスワードエージングは、システム管理者が組織内の不適切なパスワードに対して行うのに使用するもう 1 つの手法です。パスワードエージングとは、（通常は 90 日）指定した期間後に、新しいパスワードの作成を求めるプロンプトが付けられることを意味します。このため、ユーザーが定期的にパスワードを変更することが求められても、クラッカーしたパスワードは限られた時間だけ侵入者が役に立ちます。ただし、パスワードエージングのマイナス面は、ユーザーがパスワードをダウンする可能性が高くなります。

Red Hat Enterprise Linux でパスワードエージングを指定するために使用される主要なプログラムは、**chage** コマンドまたは **グラフィカルユーザーマネージャー( system-config-users**)アプリケーションです。



**重要**

**chage** コマンドを使用するには、シャドウパスワードを有効にする必要があります。詳細は、『『Red Hat Enterprise Linux 6 デプロイメントガイド』を参照してください 』。

**chage** コマンドの **-M** オプションは、パスワードが有効である日数を指定します。たとえば、ユーザーのパスワードが 90 日で期限切れになるようにするには、以下のコマンドを使用します。

```
chage -M 90 <username>
```

上記のコマンドで、*<username>* をユーザーの名前に置き換えます。パスワードの有効期限を無効にするには、**-M** オプションの **99999** 後にの値を使用します（これは 273 年より少し類似しています）。

**chage** コマンドで使用可能なオプションの詳細は、以下の表を参照してください。

**表2.1 chage コマンドラインオプション**

| オプション | 説明 |
|---|---|
| **-d** *days* | パスワードが変更された 2017 年 1 月 1 日からの日数を指定します。 |
| **-E** *日付* | アカウントがロックされる日付を YYYY-MM-DD の形式で指定します。日付の代わりに、2017 年 1 月 1 日以降の日数を使用することもできます。 |
| **-I** *days* | パスワードの有効期限からアカウントをロックするまでの非アクティブ日数を指定します。値がの場合 **0**、パスワードが失効してもアカウントはロックされません。 |
| **-l** | 現在のアカウントエージング設定を一覧表示します。 |
| **-m** *days* | ユーザーがパスワードを変更する必要のある最小日数を指定します。値がの場合 **0**、パスワードは期限切れではありません。 |
| **-M** *days* | パスワードが有効である日数の最大数を指定します。このオプションで指定された日数と、オプションで指定した日数が現在の日よりも小さい場合 **-d** は、ユーザーはアカウントを使用する前にパスワードを変更する必要があります。 |
| **-W** *days* | パスワードの有効期限の日数を指定して、ユーザーを警告します。 |

インタラクティブモードで **chage** コマンドを使用して、複数のパスワードエージングおよびアカウントの詳細を変更することもできます。インタラクティブモードに入るには、以下のコマンドを使用します。

```
chage <username>
```

インタラクティブセッションの例を以下に示します。

```
~]# chage juan
Changing the aging information for juan
Enter the new value, or press ENTER for the default
Minimum Password Age [0]: 10
Maximum Password Age [99999]: 90
Last Password Change (YYYY-MM-DD) [2006-08-18]:
Password Expiration Warning [7]:
Password Inactive [-1]:
Account Expiration Date (YYYY-MM-DD) [1969-12-31]:
```

パスワードを設定して、ユーザーが初回ログインしたときに期限切れになるように設定できます。これにより、ユーザーはパスワードをすぐに変更できるようになります。

1. 初期パスワードを設定します。この手順には、デフォルトのパスワードを割り当てるか、null パスワードを使用できます。

   デフォルトのパスワードを割り当てるには、**root** で次のコマンドを実行します。

   ```
   passwd username
   ```

   代わりに null パスワードを割り当てるには、以下のコマンドを使用します。

   ```
   passwd -d username
   ```

   **可能な限り NULL パスワードを使用しないでください。**

   任意のサードパーティーが最初にログインしてセキュアではないユーザー名を使用してシステムにアクセスするため、便利なパスワードを使用するのに便利です。必ずユーザーがログインできる状態であることを確認してから、null パスワードでアカウントをロックを解除してください。

2. **root** で以下のコマンドを実行して、パスワードの有効期限を強制的に実行します。

   ```
   chage -d 0 username
   ```

   このコマンドは、パスワードが最後にエポック（1970 年 5 月 1 日）に変更された日付の値を設定します。この値により、パスワードエージングポリシー（ある場合）に関係なく、すぐにパスワードの有効期限が強制されます。

最初のログイン時に、ユーザーに新しいパスワードの入力が求められます。

また、グラフィカルの User Manager アプリケーションを使用して、以下のようにパスワードエージングポリシーを作成することもできます。注記： この手順を実行するには、管理者権限が必要です。

1. パネルの System メニューをクリックして、以下を参照します。 **管理** をクリック Users and Groups し、ユーザーマネージャーを表示します。シェルプロンプト **system-config-users** でコマンドを入力します。

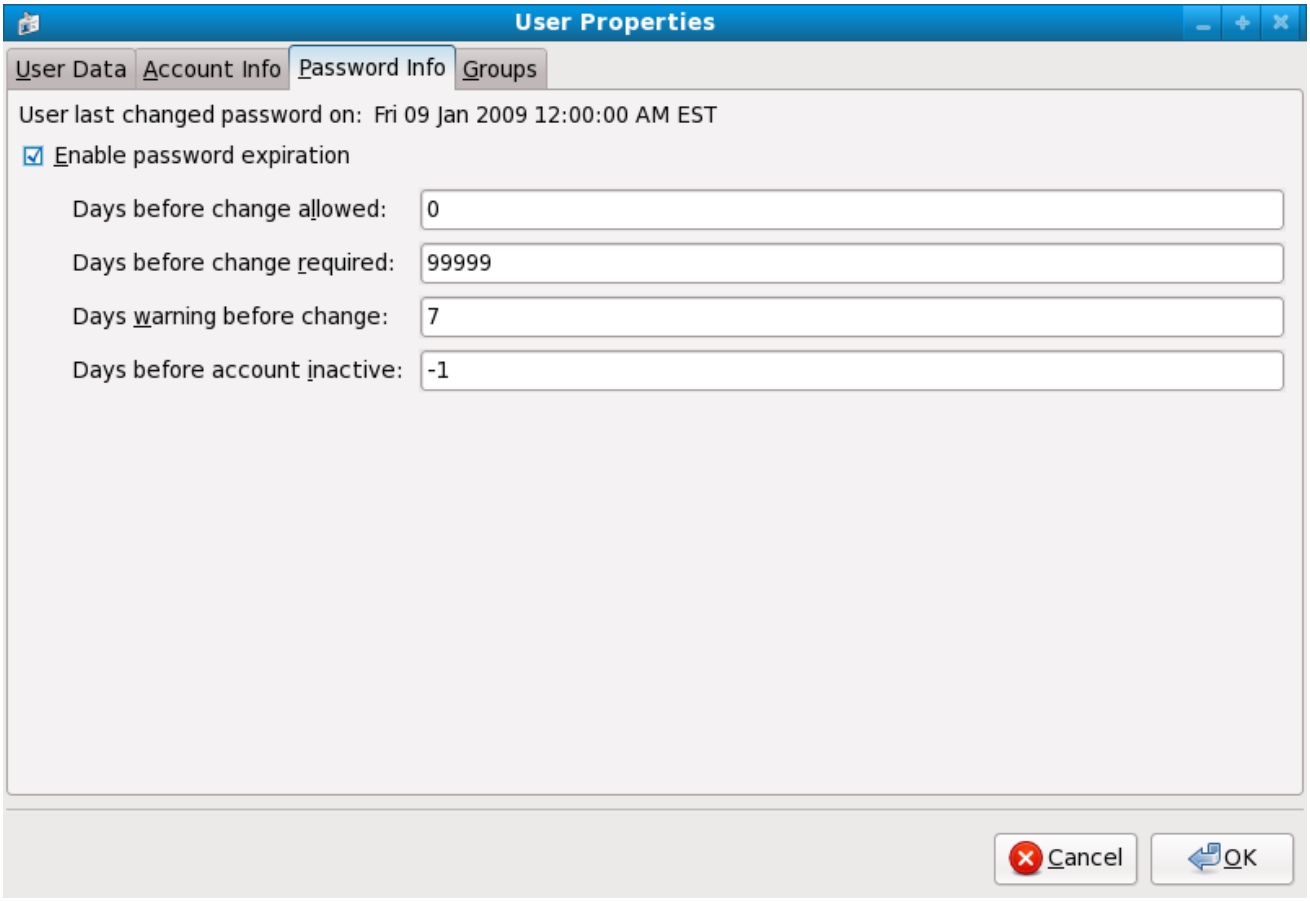
2. **Users** タブをクリックし、ユーザーリストで必要なユーザーを選択します。

3. ツールバー **Properties** をクリックして、ユーザープロパティーダイアログボックスを表示します（または File メニュー Properties で選択します）。

4. **Password Info** タブをクリックし、**パスワード有効期限の有効化** のチェックボックスを選択します。

5. 必須 **フィールドを変更する前に必要な値を Days に** 入力し、をクリックし **OK**ます。

**図2.1 パスワードエージングオプションの指定**




[D]

*スクリーンショットを更新する必要があります。*

### 2.1.5. 非アクティブアカウントのロック

**pam_lastlog** PAM モジュールは、最近ログインしていないユーザーのロックアウトや、ユーザーの最後のログイン試行に関する情報を表示するために使用されます。モジュールは root アカウントでチェックを実行しないため、ロックアウトされません。

**lastlog** コマンドは、コマンドではなく、ユーザーの最後のログインセッションを表示します。これにより **last**、現在のログインセッションと以前のログインセッションがすべて表示されます。コマンド

は、データがバイナリー形式で保存される **/var/log/lastlog** および **/var/log/wtmp** ファイルからそれぞれ読み込まれます。

- ユーザーのログインに最後に成功した前に失敗したログイン試行回数を表示するには、root で以下の行を **/etc/pam.d/login** ファイルの **session** セクションに追加します。

```
session     optional      pam_lastlog.so silent noupdate showfailed
```

非アクティブ化によるアカウントのロックは、コンソール、GUI、またはその両方で機能するように設定できます。

- 非アクティブが 10 日後にアカウントをロックアウトするには、root で以下の行を **/etc/pam.d/login** ファイルの **auth** セクションに追加します。

```
auth  required  pam_lastlog.so inactive=10
```

- GNOME デスクトップ環境のアカウントをロックアウトするには、root で **/etc/pam.d/gdm** ファイルの **auth** セクションに以下の行を追加します。

```
auth  required  pam_lastlog.so inactive=10
```

**注記**

他のデスクトップ環境では、これらの環境のそれぞれのファイルを編集する必要があることに注意してください。

### 2.1.6. アクセス制御のカスタマイズ

**pam_access** PAM モジュールを使用すると、管理者はログイン名、ホスト名、または IP アドレスに基づいてアクセス制御をカスタマイズできます。デフォルトでは、モジュールは、指定がない場合は、**/etc/security/access.conf** ファイルからアクセスルールを読み取ります。これらのルールの形式の詳細な説明は、man ページの **access.conf(5)** を参照してください。Red Hat Enterprise Linux では、デフォルトで **pam_access** は **/etc/pam.d/crond** および **/etc/pam.d/atd** ファイルに含まれています。

コンソールおよびグラフィックデスクトップ環境からユーザー john がシステムにアクセスできないようにするには、以下の手順に従います。

1. **/etc/pam.d/login** および **/etc/pam.d/gdm-*** ファイルの両方の **account** セクションに以下の行を追加します。

```
account     required      pam_access.so
```

2. **/etc/security/access.conf** ファイルに以下のルールを指定します。

```
- : john : ALL
```

このルールは、任意の場所からユーザー john からのログインをすべて禁止します。

1.2.3.4 IP アドレスのユーザー john を除く SSH を使用したログインを試みるすべてのユーザーへのアクセスを許可するには、以下の手順に従います。

1. の **account** セクションに以下の行を追加し **/etc/pam.d/sshd**ます。

```
account     required     pam_access.so
```

2. /etc/security/access.conf ファイルに以下のルールを指定します。

```
+ : ALL EXCEPT john : 1.2.3.4
```

他のサービスからのアクセスを制限するには、**/etc/pam.d** ディレクトリーの各ファイルで **pam_access** モジュールが必要になります。

以下のコマンドを使用して、システム全体の PAM 設定***-auth** ファイル（ **/etc/pam.d** ディレクトリー）を呼び出すすべてのサービスの **pam_access** モジュールを呼び出すことができます。

```
authconfig --enablepamaccess --update
```

この **pam_access** モジュールは認証設定ユーティリティーを使用して有効にできます。このユーティリティーを起動するには、トップメニュー **System → Administration → Authentication** からを選択します。**Advanced Options** タブから、「ローカルアクセス制御オプションの有効化」を確認します。これにより、**pam_access** モジュールがシステム全体の PAM 設定に追加されます。

### 2.1.7. 時間ベースのアクセス制限

**pam_time** PAM モジュールは、1 日のある特定の時間内にアクセスを制限するために使用されます。週、ユーザー名、システムサービスの使用状況などに基づいてアクセスを制御するように設定することもできます。デフォルトでは、モジュールは **/etc/security/time.conf** ファイルからアクセスルールを読み取ります。これらのルールの形式の詳細な説明は、**time.conf(5)** man ページを参照してください。

root ユーザーを除くすべてのユーザーを 05:30 PM 05:30 PM から午前 08:00 時に制限するには、以下の手順を実行します。

1. **/etc/pam.d/login** ファイルの account セクションに以下の行を追加します。

```
account     required     pam_time.so
```

2. **/etc/security/time.conf** ファイルに以下のルールを指定します。

```
login ; tty* ; ALL ; !root ; !Wk1730-0800
```

ユーザー john が作業時間および稼働日（月曜日から開始）時に SSH サービスを使用できるようにするには、以下の手順に従います。

1. に以下の行を追加します。 **/etc/pam.d/sshd file:**

```
account     required     pam_time.so
```

2. **/etc/security/time.conf** ファイルに以下のルールを指定します。

```
sshd ; tty* ; john ; Wk0800-1730
```

**注記**

これらの設定をデスクトップ環境に適用するには、**pam_time** モジュールを **/etc/pam.d** ディレクトリー内の対応するファイルに追加する必要があります。

### 2.1.8. アカウント制限の適用

**pam_limits** PAM モジュールは、以下の目的で使用されます。

- ユーザーごとに同時ログインセッションの最大数などの、ユーザーログインセッションに制限を適用します。
- **ulimit** ユーティリティーで設定する制限を指定します。
- and は、**nice** ユーティリティーで設定する優先度を指定します。

デフォルトでは、ルールは**/etc/security/limits.conf** ファイルから読み込まれます。これらのルールの形式の詳細な説明は、man ページの **limits.conf(5)** を参照してください。さらに、特定のアプリケーションやサービス用に個別の設定ファイルを **/etc/security/limits.d** ディレクトリーに作成できます。デフォルトでは、**pam_limits** モジュールは**/etc/pam.d/** ディレクトリー内の複数のファイルに含まれます。ユーザープロセスのデフォルトの制限は **/etc/security/limits.d/90-nproc.conf** ファイルで定義され、フォークなど、サービス攻撃の悪意のある拒否を防ぐことができます。ユーザープロセスのデフォルトの上限を 50 に変更するには、ファイルでの形式を **/etc/security/limits.d/90-nproc.conf**以下のように変更します。

```
* soft nproc 50
```

**例2.2 ユーザーあたりの最大ログイン数の指定**

1. グループ内の各ユーザーに対して同時ログインの最大数を設定するには **office**、**/etc/security/limits.conf** ファイルに以下のルールを指定します。

   ```
   @office - maxlogins 4
   ```

2. にデフォルトで以下の行を追加する必要があり **/etc/pam.d/system-auth**ます。そうでない場合は、手動で追加します。

   ```
   session  required  pam_limits.so
   ```

### 2.1.9. 管理的コントロール

ホームマシンを管理する場合は、一部のタスクを root ユーザーで実行するか、**sudo** またはなどの *setuid* プログラムを介して有効な root 権限を付与する必要があり **su**ます。setuid プログラムは、プログラムを実行するユーザーではなく、プログラムの所有者のユーザー ID(**UID**)を操作するプログラムです。このようなプログラムは、以下の例のように、長い形式のリストの **s** 所有者セクションので示されます。

```
~]$ ls -l /bin/su
-rwsr-xr-x. 1 root root 34904 Mar 10  2011 /bin/su
```

**注記**

**s** は大文字または小文字になります。大文字として表示される場合は、基礎となるパーミッションビットが設定されていないことを意味します。

ただし、組織のシステム管理者には、組織内のユーザーがマシンにどの程度必要かを決める必要があります。と呼ばれる PAM モジュールを介して通常 **pam_console.so**、一部のアクティビティーは、物理

コンソールでログインする最初のユーザーに対しては、再起動やリムーバブルメディアのマウントなど、root ユーザーのみに予約されています（ **pam_console.so** モジュールの詳細は、「『シングルサインオンの 管理」および「スマートカード』 の管理」を参照してください）。 ただし、ネットワーク設定の変更、新しいマウスの設定、ネットワークデバイスのマウントなど、その他の重要なシステム管理タスクは、管理者権限なしではできません。したがって、システム管理者は、ネットワーク上のユーザーアクセスの量を決定する必要があります。

#### 2.1.9.1. Root アクセスの許可

組織内のユーザーが信頼され、コンピューターに命令されていれば、root アクセスを許可することは問題ではない可能性があります。ユーザーが root アクセスを許可すると、デバイスの追加やネットワークインターフェースの設定などのマイナーアクティビティーは、個々のユーザーが処理できるため、システム管理者はネットワークセキュリティーやその他の重要な問題に対処することができます。

一方、個々のユーザーに root アクセスを付与すると、以下の問題が発生する可能性があります。

- マシン**の移行： root アクセスを持つユーザーはマシンを誤っ** て設定し、問題の解決に支援を必要とする可能性があります。悪意あふれるとしても、それに気付いてもセキュリティーの欠陥が開く可能性があります。

- セキュア**でないサービス** - root アクセスを持つユーザーは、FTP、Telnet などのセキュアでないサーバーを実行すると、ユーザー名とパスワードが危険にさらされる可能性があります。これらのサービスは、プレーンテキストでネットワーク経由でこの情報を送信します。

- **Email Attachments As Root の実行**: Linux に影響するまれな電子メールウイルスが存在します。ただし、これらが脅威になる唯一のタイミングは、root ユーザーが実行している時です。

- **監査証跡をそのまま保持** する - ルートアカウントは複数のユーザーによって共有されることが多いため、複数のシステム管理者がシステムを管理できるため、指定した時点でどのユーザーが root であったかを特定することはできません。別のログインを使用する場合、ユーザーがでログインするアカウントと、セッション追跡目的で一意の番号がタスク構造に置かれます。これは、ユーザーが開始するすべてのプロセスが継承されます。同時ログインを使用する場合、一意の番号を使用してアクションを特定のログインを追跡できます。アクションが監査イベントを生成すると、ログインアカウントと、その一意の番号に関連付けられたセッションで記録されます。これらのログインおよびセッションを表示するには、**aulast** コマンドを使用します。**aulast** コマンドの **--proof** オプションを使用すると、特定のセッションで生成された監査可能なイベントを分離するための特定の **ausearch** クエリーが提案されます。

#### 2.1.9.2. Root アクセスの拒否

管理者がこのような理由またはその他の理由で root としてログインできない場合は、root パスワードを秘密にし、1 つまたは単一ユーザーモードへのアクセスはブートローダーのパスワード保護では拒否する必要があります（本トピックの 「ブートローダーのパスワード」 詳細はを参照してください）。

以下は、管理者がルートログインを拒否する 4 つの方法になります。

**root シェルの変更**

ユーザーが root として直接ログインできないようにするには、システム管理者が **/etc/passwd** ファイル **/sbin/nologin** で root アカウントのシェルをに設定します。

**表2.2 Root Shell の無効化**

| 影響 | 影響なし |
| --- | --- |
| root シェルへのアクセスを阻止し、このような試行をログに記録します。以下のプログラムは、root アカウントにアクセスできません。<br>• **login**<br>• **gdm**<br>• **kdm**<br>• **xdm**<br>• **su**<br>• **ssh**<br>• **scp**<br>• **sftp** | FTP クライアント、メールクライアント、および setuid プログラムなど、シェルを必要としないプログラム。以下のプログラムは、root アカウントにアクセスでき**ません**。<br>• **sudo**<br>• FTP クライアント<br>• メールクライアント |

**コンソールデバイス(tty)による root アクセスの無効化**

root アカウントへのアクセスをさらに制限するために、管理者は **/etc/securetty** ファイルを編集して、コンソールでの root ログインを無効にすることができます。このファイルは、root ユーザーがログインできるすべてのデバイスを一覧表示します。ファイルが存在しない場合は、root ユーザーは、コンソールまたは raw ネットワークインターフェースを使用して、システム上の通信デバイスを介してログインできます。ユーザーが Telnet を介して root としてマシンにログインできるため、ネットワーク上でパスワードをプレーンテキストで送信できるため、危険です。

デフォルトでは、Red Hat Enterprise Linux の **/etc/securetty** ファイルでは、root ユーザーはマシンに物理的に接続されているコンソールでのみログインできます。root ユーザーがログインしないようにするには、root で次のコマンドを実行します。

```
echo > /etc/securetty
```

KDM、GDM、および XDM ログインマネージャーで **securetty** サポートを有効にするには、以下の行を追加します。

```
auth [user_unknown=ignore success=ok ignore=ignore default=bad] pam_securetty.so
```

以下に記載されているファイル。

- **/etc/pam.d/gdm**
- **/etc/pam.d/gdm-autologin**
- **/etc/pam.d/gdm-fingerprint**
- **/etc/pam.d/gdm-password**
- **/etc/pam.d/gdm-smartcard**
- **/etc/pam.d/kdm**

- **/etc/pam.d/kdm-np**
- **/etc/pam.d/xdm**

**警告**

空の **/etc/securetty** ファイルは、認証後までコンソールを開くことができ **ない** ため、root ユーザーがツールの OpenSSH スイートを使用してリモートでログインできないようにする訳ではありません。

**表2.3 root ログインの無効化**

| 影響 | 影響なし |
| --- | --- |
| コンソールまたはネットワークを使用して root アカウントにアクセスできないようにします。以下のプログラムは、root アカウントにアクセスできません。<br>• **login**<br>• **gdm**<br>• **kdm**<br>• **xdm**<br>• tty を開くその他のネットワークサービス | root としてログインせず、setuid またはその他のメカニズムを使用して管理タスクを実行します。以下のプログラムは、root アカウントにアクセスでき **ません**。<br>• **su**<br>• **sudo**<br>• **ssh**<br>• **scp**<br>• **sftp** |

### root SSH ログインの無効化

SSH プロトコルを使用した root ログインを防ぐには、SSH デーモンの設定ファイルを編集して **/etc/ssh/sshd_config**、の行を変更します。

```
#PermitRootLogin yes
```

以下で読み込むには、以下のコマンドを実行します。

```
PermitRootLogin no
```

**表2.4 root SSH ログインの無効化**

| 影響 | 影響なし |
| --- | --- |
| OpenSSH スイートを使用して root アクセスを阻止します。以下のプログラムは、root アカウントにアクセスできません。<br>• **ssh**<br>• **scp**<br>• **sftp** | OpenSSH スイートに含まれないプログラム。 |

### PAM を使用したサービスへの root アクセスを制限する

**/lib/security/pam_listfile.so** モジュールを介して PAM により、特定のアカウントを柔軟に拒否できます。管理者は、このモジュールを使用して、ログインが許可されないユーザーの一覧を参照できます。システムサービスへの root アクセスを制限するには、**/etc/pam.d/** ディレクトリー内のターゲットサービスのファイルを編集し、**pam_listfile.so** モジュールが認証に必要であることを確認します。

以下は、**/etc/pam.d/vsftpd** PAM 設定ファイルの **vsftpd** FTP サーバーでモジュールがどのように使用されているかの例になります（ディレクティブが 1 行目にある場合は、最初の行の最後にある \ 文字は必要あり **ません** ）。

```
auth    required    /lib/security/pam_listfile.so    item=user \
 sense=deny file=/etc/vsftpd.ftpusers onerr=succeed
```

これにより、PAM が **/etc/vsftpd.ftpusers** ファイルを確認し、一覧表示されたユーザーのサービスへのアクセスを拒否するよう指示します。管理者はこのファイルの名前を変更でき、サービスごとに個別の一覧を保持することや、1 つの中央リストを使用して複数のサービスへのアクセスを拒否することができます。

管理者が複数のサービスへのアクセスを拒否する場合は、メールクライアント **/etc/pam.d/pop** や SSH クライアントなど、PAM 設定ファイルに同様 **/etc/pam.d/imap** の行 **/etc/pam.d/ssh** を追加できます。

PAM の詳細は、『『Red Hat Enterprise Linux Managing Single Sign-On and Smart Cards 』』の「『Using Pluggable Authentication Modules(PAM) 』」の章を参照してください。

**表2.5 PAM を使用した root の無効化**

| 影響 | 影響なし |
| --- | --- |
| PAM 対応ネットワークサービスへの root アクセスを防ぎます。以下のサービスは、root アカウントにアクセスできません。<br>• **login**<br>• **gdm**<br>• **kdm**<br>• **xdm**<br>• **ssh**<br>• **scp**<br>• **sftp**<br>• FTP クライアント<br>• メールクライアント<br>• PAM 対応サービス | PAM に対応していないプログラムやサービス |

#### 2.1.9.3. 自動ログアウトの有効化

**root** としてユーザーがログインすると、無人ログインセッションによりセキュリティー上のリスクが大幅に低下する可能性があります。このリスクを軽減するには、一定期間後にアイドルユーザーを自動的にログアウトするようにシステムを設定できます。

1. screen パッケージがインストールされていることを確認します。これを行うには、**root** で以下のコマンドを実行します。

   ```
   ~]# yum install screen
   ```

   Red Hat Enterprise Linux でパッケージをインストールする方法は、『『Red Hat Enterprise Linux 6 デプロイメントガイド』の「パッケージのインストール」セクションを参照してください』。

2. **root** で以下の行をファイルの先頭に追加し、この **/etc/profile** ファイルの処理を中断しないようにします。

   ```
   trap "" 1 2 3 15
   ```

3. ユーザーが仮想コンソールまたはリモートでログインするたびに **screen** セッションを開始するには、**/etc/profile** ファイルの末尾に以下の行を追加します。

   ```
   SCREENEXEC="screen"
   if [ -w $(tty) ]; then
     trap "exec $SCREENEXEC" 1 2 3 15
     echo -n 'Starting session in 10 seconds'
   ```

```
  sleep 10
  exec $SCREENEXEC
fi
```

新しいセッションを開始するたびに、メッセージが表示され、ユーザーは 10 秒待機する必要があることに注意してください。セッションの開始前に待機する時間を調整するには、**sleep** コマンドの後に値を変更します。

4. **/etc/screenrc** 設定ファイルに以下の行を追加して、特定のアクティブでない期間の後に **screen** セッションを閉じます。

```
idle 120 quit
autodetach off
```

これにより、時間制限が 120 秒に設定されます。この制限を調整するには、**idle** ディレクティブの後に値を変更します。

代わりに、次の行を使用して、セッションのみをロックするようにシステムを設定できます。

```
idle 120 lockscreen
autodetach off
```

こうすることで、セッションのロックを解除するにはパスワードが必要になります。

この変更は、ユーザーが次回システムにログインしたときに有効になります。

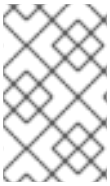
#### 2.1.9.4. ルートアクセスの制限

root ユーザーへのアクセスを完全に拒否するのではなく、管理者は、**su** やなどの setuid プログラムによるアクセスのみを許可でき **sudo**ます。**su** およびの詳細は、『Red Hat Enterprise Linux 6 デプロイメントガイド』 **su(1)** および **sudo(8)** man ページを **sudo**参照してください。

#### 2.1.9.5. アカウントのロック

Red Hat Enterprise Linux 6 では、PAM モジュール **pam_faillock** により、システム管理者は、指定した回数失敗した試行後にユーザーアカウントをロックできます。ユーザーのログイン試行を制限することは主に、ユーザーのアカウントパスワードの取得先となるブルートフォース攻撃の防止を目的とするセキュリティー対策として機能します。

**pam_faillock** モジュールを使用すると、失敗したログイン試行は **/var/run/faillock** ディレクトリーの各ユーザーに対して別のファイルに保存されます。



**注記**

ログファイルの試行に失敗した行の順序は重要です。この順序の変更により、**even_deny_root** オプションが使用されると、root ユーザーアカウントを含むすべてのユーザーアカウントがロックされます。

以下の手順に従って、アカウントのロックを設定します。

1. root 以外のユーザーを 3 回ロックして、そのユーザーが 10 分後にロックを解除した後にロックアウトするには、**/etc/pam.d/system-auth** および **/etc/pam.d/password-auth** ファイルの **auth** セクションに以下の行を追加します。

```
auth        required      pam_faillock.so preauth silent audit deny=3 unlock_time=600
auth        sufficient    pam_unix.so nullok try_first_pass
auth        [default=die]  pam_faillock.so authfail audit deny=3 unlock_time=600
```

2. 以下の行を、直前の手順で指定されている両方のファイルの **account** セクションに追加します。

```
account     required      pam_faillock.so
```

3. root ユーザーにもアカウントロックを適用するには、**/etc/pam.d/system-auth** および **/etc/pam.d/password-auth** ファイルの **pam_faillock** エントリーに **even_deny_root** オプションを追加します。

```
auth        required      pam_faillock.so preauth silent audit deny=3 even_deny_root
unlock_time=600
auth        sufficient    pam_unix.so nullok try_first_pass
auth        [default=die] pam_faillock.so authfail audit deny=3 even_deny_root
unlock_time=600

account     required      pam_faillock.so
```

ユーザーが 3 回ログインに失敗した後に 4 回ログインを **john** 試みると、4 番目の試行時にアカウントがロックされます。

```
[user@localhost ~]$ su - john
Account locked due to 3 failed logins
su: incorrect password
```

システムが複数のログインに失敗しても、システムがユーザーのロックアウトを防ぐには、とで初めて呼び出さ **pam_faillock** れる行の上に次の行を追加 **/etc/pam.d/system-auth** し **/etc/pam.d/password-auth**ます。また **user1**、**user2**、**user3** を実際のユーザー名に置き換えます。

```
auth [success=1 default=ignore] pam_succeed_if.so user in user1:user2:user3
```

ユーザーごとの失敗した試行回数を表示するには、root で以下のコマンドを実行します。

```
[root@localhost ~]# faillock
john:
When                Type  Source                                           Valid
2013-03-05 11:44:14 TTY   pts/0                                               V
```

ユーザーのアカウントをアンロックするには、root で以下のコマンドを実行します。

```
faillock --user <username> --reset
```

**authconfig** ユーティリティーを使用して認証設定を変更する場合、**system-auth** および **password-auth** ファイルは、**authconfig** ユーティリティーの設定で上書きされます。これは、設定ファイルの代わりにシンボリックリンクを作成すると回避できます。これは、**authconfig** が認識し、上書きしません。設定ファイルでカスタム設定と **authconfig** を同時に使用するには、以下の手順に従ってアカウントのロックを設定します。

1. 設定ファイルの名前を変更します。

```
~]# mv /etc/pam.d/system-auth /etc/pam.d/system-auth-local
~]# mv /etc/pam.d/password-auth /etc/pam.d/password-auth-local
```

2. 以下のシンボリックリンクを作成します。

```
~]# ln -s /etc/pam.d/system-auth-local /etc/pam.d/system-auth
~]# ln -s /etc/pam.d/password-auth-local /etc/pam.d/password-auth
```

3. この **/etc/pam.d/system-auth-local** ファイルには以下の行が含まれている必要があります。

```
auth        required      pam_faillock.so preauth silent audit deny=3 unlock_time=600
auth        include       system-auth-ac
auth        [default=die] pam_faillock.so authfail silent audit deny=3 unlock_time=600

account     required      pam_faillock.so
account     include       system-auth-ac

password    include       system-auth-ac

session     include       system-auth-ac
```

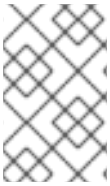
4. この **/etc/pam.d/password-auth-local** ファイルには以下の行が含まれている必要があります。

```
auth        required      pam_faillock.so preauth silent audit deny=3 unlock_time=600
auth        include       password-auth-ac
auth        [default=die] pam_faillock.so authfail silent audit deny=3 unlock_time=600

account     required      pam_faillock.so
account     include       password-auth-ac

password    include       system-auth-ac

session     include       system-auth-ac
```

さまざまな **pam_faillock** 設定オプションの詳細は、man ページの **pam_faillock(8)** を参照してください。

### 2.1.10. セッションのロック

運用中には、多くの理由でワークステーションを無人のままにしないといけない場合があります。これにより、攻撃者は特に物理的なセキュリティー対策が不十分な環境でマシンに物理的にアクセスできる可能性があります（を参照 「物理的コントロール」）。ラップトップは特にモビリティーが物理的なセキュリティーに干渉するため、特に公開されています。セッションロック機能を使用することで、このようなリスクを軽減できます。これにより、正しいパスワードの入力が完了するまでシステムへのアクセスが阻止されます。



**注記**

ログアウトせずに画面をロックする主な利点は、ロックにより、ユーザーのプロセス（ファイル転送など）の実行を継続できることです。ログアウトはこれらのプロセスを停止します。

#### 2.1.10.1. gnome-screensaver-command を使用した GNOME のロック

Red Hat Enterprise Linux 6 のデフォルトのデスクトップ環境には、GNOME が含まれています。この機能により、ユーザーはいつでも画面をロックできます。ロックをアクティベートする方法は複数あります。

- で指定されているキーの組み合わせを押し **System → Preferences → Keyboard Shortcuts → Desktop → Lock screen**ます。デフォルトの組み合わせはです **Ctrl**+**Alt**+**L**。
- パネル **System → Lock screen** でを選択します。
- コマンドラインインターフェースから以下のコマンドを実行します。

```
gnome-screensaver-command -l
```

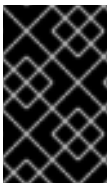
説明されている手法はすべて同じ結果を持ちます。スクリーンセーバーはアクティブになり、画面はロックされます。その後、任意のキーを押してスクリーン保存を非アクティブにし、パスワードを入力して作業を継続できます。

この関数には **gnome-screensaver** プロセスを実行する必要があります。プロセスに関する情報を提供するコマンドを使用して、これが確認できるかどうかを確認できます。たとえば、ターミナルから以下のコマンドを実行します。

```
pidof gnome-screensaver
```

**gnome-screensaver** プロセスが実行中の場合は、コマンドを実行した後に画面に識別番号(PID)を示す数字が表示されます。プロセスが現在実行していない場合は、コマンドで出力は表示されません。

追加情報は、の **gnome-screensaver-command(1)** man ページを参照してください。



**重要**

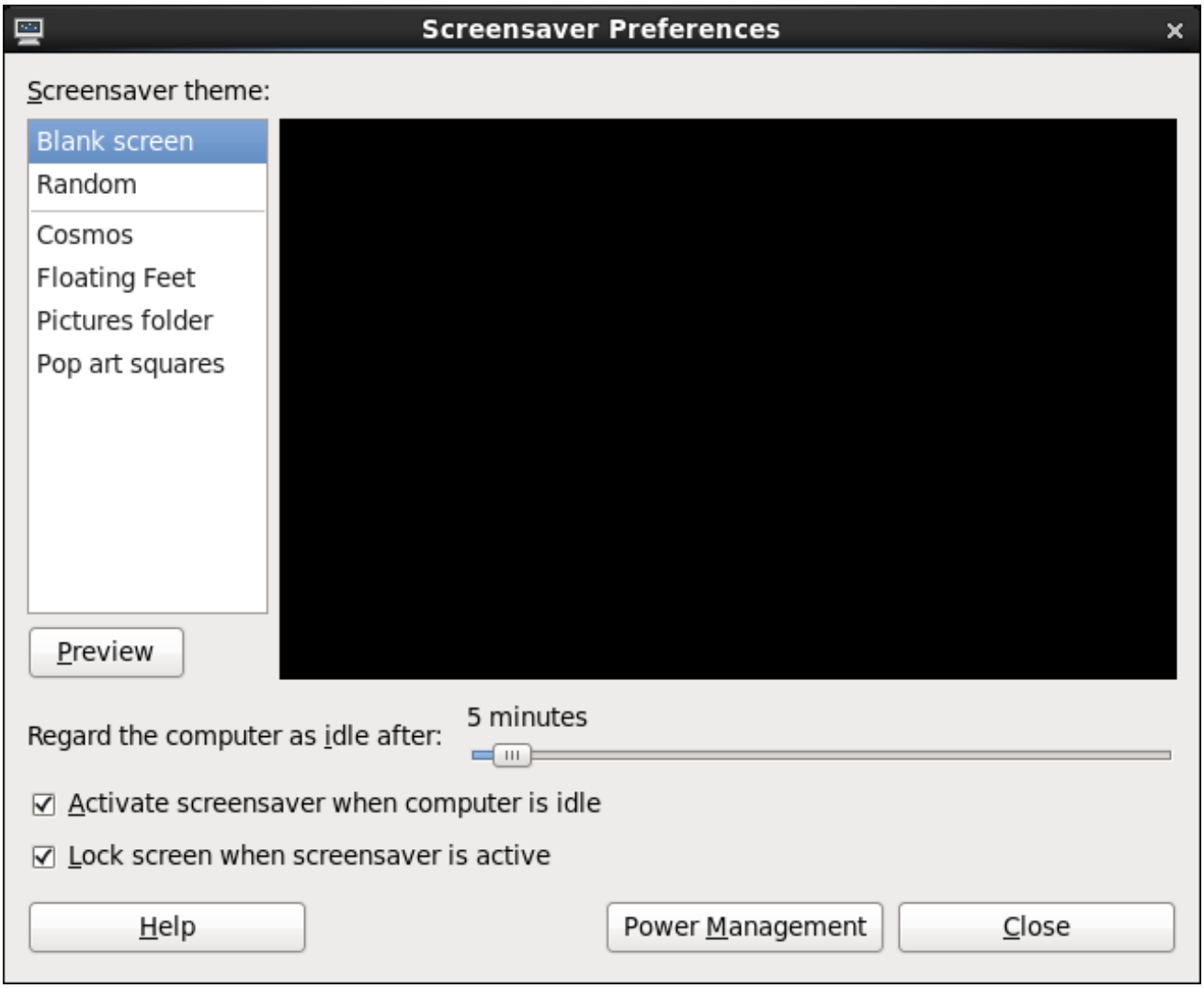
上記の画面をロックするには、手動によるアクティベーションが必要です。このため、管理者は短期間だけであっても、無人状態のままになるたびにコンピューターをロックするよう推奨する必要があります。

#### 2.1.10.1.1. スクリーンセーバーのアクティベーションの自動ロック

この名前が **gnome-screensaver-command** 示すように、ロック機能は GNOME のスクリーンセーバーに関連付けられます。スクリーンセーバーのアクティベーションにロックを結び付け、一定期間にわたり無人状態になるたびにワークステーションをロックできます。この機能は、デフォルトで 5 分タイムアウトでアクティベートされます。

自動ロック設定を変更する場合は、メインパネル **System → Preferences → Screensaver** でを選択します。これにより、タイムアウトの時間（スケーラーの **後にコンピューターをアイドル状態** にする）を設定し、自動ロックのアクティブ化または非アクティブ化（ **スクリーンセーバーがアクティブな場合にロック画面）をアクティブまたは非アクティブ** にするウィンドウが開きます。

**図2.2 スクリーン保存者設定の変更**




[D]

**注記**

**Screensaver Preferences** ダイアログで、**コンピューターがアイドル状態のときに Activate スクリーンセーバー** を無効にすると、スクリーンセーバーが自動的に起動できなくなります。このため、自動ロック機能も無効にされますが、に記載の手法を使用してワークステーションを手動でロックすることは可能です 「gnome-screensaver-command を使用した GNOME のロック」。

#### 2.1.10.1.2. リモートセッションのロック

ターゲットワークステーションがこのプロトコルの接続を受け入れる **ssh** 限り、GNOME セッションをリモートでロックすることもできます。アクセスできるマシンで画面をリモートでロックするには、以下のコマンドを実行します。

```
ssh -X <username>@<server> "export DISPLAY=:0; gnome-screensaver-command -l"
```

*<username>* をユーザー名に置き換え、*<server>* をロックするワークステーションの IP アドレスに置き換えます。

詳細はを 「セキュアなシェル」 参照してください **ssh**。

#### 2.1.10.2. vlock を使用した仮想コンソールのロック

また、ユーザーは仮想コンソールをロックする必要があります。これは、と呼ばれるユーティリティーを使用して実行でき **vlock**ます。このユーティリティーをインストールするには、root で以下のコマンドを実行します。

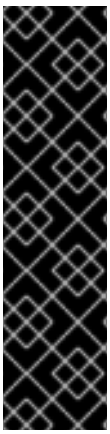
```
~]# yum install vlock
```

インストール後には、**vlock** コマンドでパラメーターを付けずにコンソールセッションをロックできます。これにより、現在アクティブな仮想コンソールセッションがロックされ、他のセッションへのアクセスは許可されます。ワークステーションのすべての仮想コンソールへのアクセスを防ぐには、次のコマンドを実行します。

```
vlock -a
```

この場合は、現在アクティブなコンソールを **vlock** ロックし、**-a** オプションにより、他の仮想コンソールへの切り替えができなくなります。

追加情報は、の **vlock(1)** man ページを参照してください。



**重要**

**vlock** 現在利用できる Red Hat Enterprise Linux 6 のバージョンには、以下の既知の問題がいくつかあります。

- 現在、プログラムは root パスワードを使用してコンソールのロックを解除することはできません。詳細は、BZ# 895066 を参照してください。
- コンソールのロックでは、画面とスクロールバックバッファーはクリアされず、ワークステーションに物理的にアクセスできるすべてのユーザーが、以前に実行したコマンドや、コンソールに表示された出力を表示することができます。詳細は、BZ# 807369を 参照してください。

### 2.1.11. 利用可能なネットワークサービス

管理的コントロールへのユーザーアクセスは組織内のシステム管理者にとって重要ですが、Linux システムを管理および運用するユーザーにとって、どのネットワークサービスもアクティブなネットワークサービスを監視することが重要です。

Red Hat Enterprise Linux 6 のサービスの多くは、ネットワークサーバーとして動作します。ネットワークサービスがマシンで実行している場合、サーバーアプリケーション（ *デーモン*と呼ばれます）は、1 つ以上のネットワークポートでの接続をリッスンしています。これらのサーバーは、潜在的な攻撃として扱う必要があります。

#### 2.1.11.1. サービスへのリスク

ネットワークサービスは、Linux システムに多くのリスクを課す可能性があります。以下は、主な問題の一部です。

- **サービス拒否(DoS): リクエストによりサービス拒否(DoS)により、各リクエストのログと応答を試みるため、サービス拒否攻撃** によりシステムが使用できなくなる可能性があります。
- **分散型サービス拒否(DDoS)- DoS 攻撃の**1 つで、複数の不正なマシン（数千個以上のマシン）を使用して、サービス上で調整された攻撃を指示し、要求で改ざんして使用不可能にします。

- Script Vulnerability Ackscks - サーバーがスクリプトを使用してサーバー側のアクション（Web サーバー）を一般的に実行すると、攻撃者は誤って書き込まれたスクリプトを攻撃できます。このスクリプトの脆弱性攻撃により、バッファーオーバーフロー状態が発生したり、攻撃者がシステム上のファイルを変更したりする可能性があります。

- bufferOverflow Ackscks - 番号 が付けられたポート 0 から 1023 までのポートに接続するサービスは、管理ユーザーとして実行する必要があります。アプリケーションに悪用可能なバッファーオーバーフローがある場合、攻撃者はデーモンを実行しているユーザーとしてシステムにアクセスできる可能性があります。悪用可能なバッファーオーバーフローが存在するため、攻撃者は自動ツールを使用して脆弱性のあるシステムを特定し、アクセス権限が大きいと、自動ルートキットを使用してシステムへのアクセスを維持します。

**注記**

Red Hat Enterprise Linux では、x86 互換カーネルおよびマルチプロセッサーカーネルがサポートする実行可能なメモリーセグメンテーションと保護テクノロジーである 、 Red Hat Enterprise Linux では、バッファーオーバーフローの脆弱性の脅威が軽減されます。Execstructeld は、仮想メモリーを実行可能なセグメントと実行不可能なセグメントに分割することで、バッファーオーバーフローのリスクを軽減します。実行可能なセグメント（バッファーオーバーフローの悪用からインジェクトされた悪意のあるコードなど）以外を実行しようとするプログラムコードは、セグメンテーションフォールトをトリガーし、終了します。

Execleeeld には *No eXecute* のサポートも含まれています（NXAMD64 プラットフォームおよび *eXecute Disable* （）の技術XD）Itanium およびの技術 Intel® 64 システムこれらの技術は Execleeeld と連動し、悪意のあるコードが仮想メモリーの実行可能な部分で、4KB の実行可能ファイルで実行されないようにし、攻撃のリスクをバッファーオーバーフローの悪用からのリスクを軽減します。

**重要**

ネットワーク上での攻撃の公開を制限するために、未使用のサービスをすべて無効にします。

#### 2.1.11.2. サービスの特定と設定

セキュリティーを強化するため、Red Hat Enterprise Linux でインストールしたほとんどのネットワークサービスは、デフォルトで無効になっています。ただし、主な例外があります。

- **cupsd** : Red Hat Enterprise Linux のデフォルトプリントサーバー

- **lpd** : 代替のプリントサーバーです。

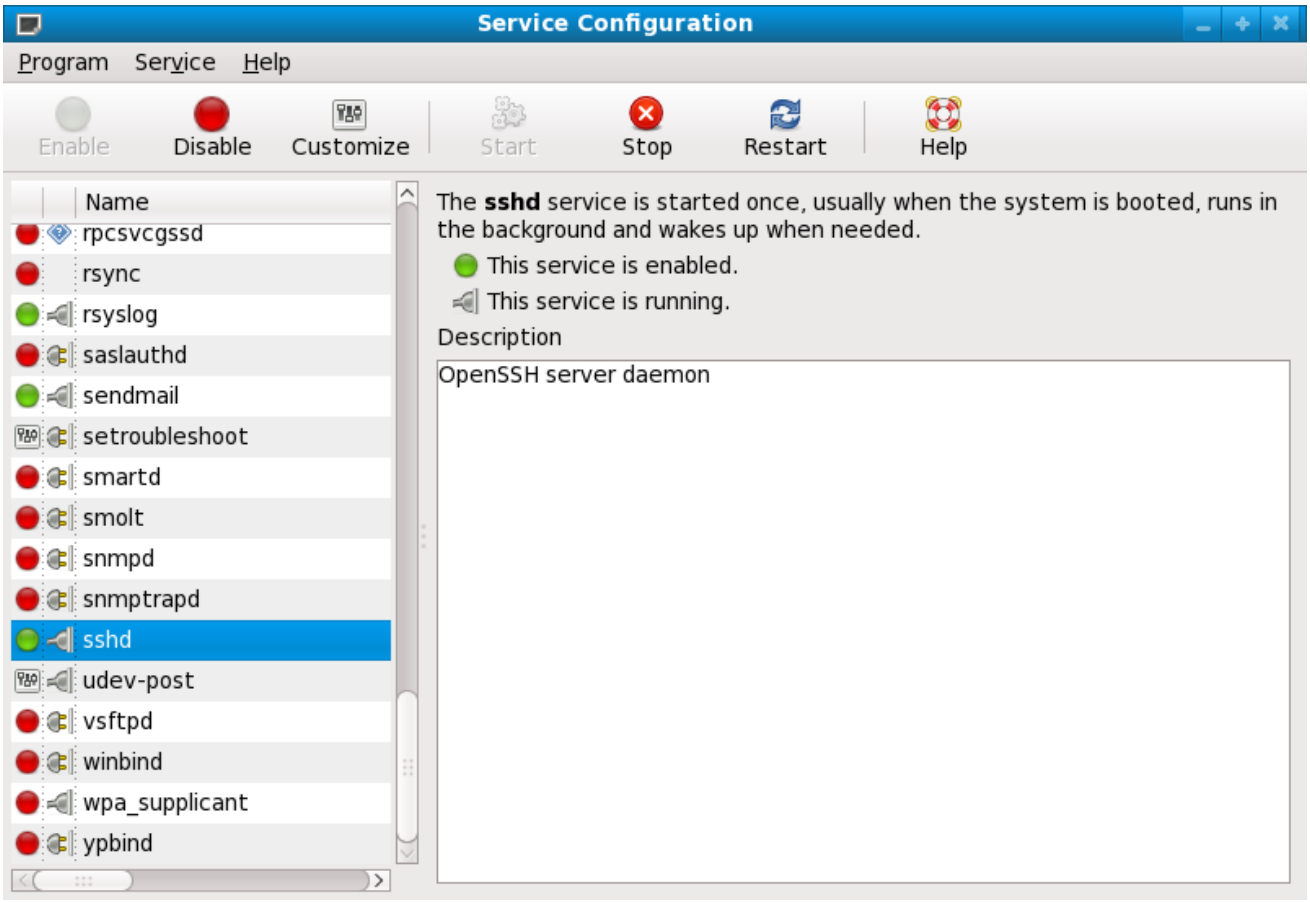
- **xinetd** : やなど、さまざまな下位サーバーへの接続を制御するスーパーサーバーです **gssftp telnet**。

- **sendmail** : Sendmail *メール転送エージェント* （）MTA）はデフォルトで有効になっていますが、からの接続のみをリッスンします。 localhost.

- **sshd** : OpenSSH サーバー。これは、Telnet の安全な代替です。

これらのサービスを実行中のままにするかどうかを決定する際には、一般的な意味を使用し、リスクを避けることが推奨されます。たとえば、プリンターが利用できない場合は、**cupsd** 実行しなくなります。も同様です **portmap**。NFSv3 ボリュームをマウントしたり、NIS（ **ypbind** サービス）を使用する場合、無効にする **portmap** 必要があります。

図2.3 Services Configuration Tool




[D]

特定のサービスの目的が分からない場合は、Service **Configuration Tool** に説明されている説明フィールドがあり 図2.3「Services Configuration Tool」、追加情報を提供します。

システムの起動時に起動することのできるネットワークサービスだけでは不十分です。開いているポートとリッスンするポートも確認することを推奨します。詳細は「ポートが一覧表示されるかどうかの確認」を参照してください。

### 2.1.11.3. 安全ではないサービス

可能性として、すべてのネットワークサービスが安全ではない可能性があります。このため、未使用のサービスをオフにすることが非常に重要です。サービスの悪用は定期的に作成およびパッチが適用され、ネットワークサービスに関連するパッケージを定期的に更新することが非常に重要です。詳細は「セキュリティー更新」を参照してください。

一部のネットワークプロトコルは、基本的に他のプロトコルよりも安全ではないものもあります。これには、以下のサービスが含まれます。

- 暗号化**されていないネットワーク上でのユーザー名およびパスワードの送信** - Telnet や FTP などの古いプロトコルは認証セッションを暗号化せず、可能な限り使用しないようにしてください。

- 暗号化**されていないネットワーク上で機密データを送信** する - 多くのプロトコルは、暗号化されていないネットワーク上でデータを転送します。これらのプロトコルには、Telnet、FTP、HTTP、SMTP などがあります。NFS や SMB などの多くのネットワークファイルシステムも、暗号化されていないネットワーク上で情報を送信します。送信されるデータのタイプを制限するために、これらのプロトコルを使用する場合のユーザーの責任です。

リモートメモリーダンプサービスは、ネットワーク経由で暗号化されていない状態でメモリーの内容を **netdump**送信します。メモリーダンプには、パスワードや、さらにはデータベースエントリー、およびその他の機密情報が含まれることがあります。

システムのユーザーに関する情報 **rwhod** など **finger**、その他のサービスに通知します。

本質的にセキュリティー保護されていないサービスの例に **rlogin**は、**rsh**、**telnet**、およびがあり **vsftpd**ます。

リモートログインおよびシェルプログラム（、**rlogin**、**telnet**）はすべて **rsh**、SSH を利用するために使用しないでください。の詳細は、「セキュリティーの強化通信ツール」 を参照してください **sshd**。

FTP は、基本的にリモートシェルとしてシステムのセキュリティーに危険が及ぶことはありませんが、問題を回避するために FTP サーバーを慎重に設定し、監視する必要があります。FTP サーバーのセキュリティー保護 「FTP のセキュリティー保護」 に関する詳細は、を参照してください。

注意してファイアウォールの背後で実装すべきサービスには、以下が含まれます。

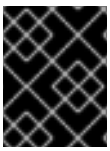
- **finger**
- **authd** （これは、以前の Red Hat Enterprise Linux リリース **identd** で知られていました。）
- **netdump**
- **netdump-server**
- **nfs**
- **rwhod**
- **sendmail**
- **smb** (Samba)
- **yppasswdd**
- **ypserv**
- **ypxfrd**

ネットワークサービスのセキュリティー保護に関する詳細は、を参照してください 「サーバーセキュリティー」。

次のセクションでは、簡単なファイアウォールを設定するのに使用できるツールを説明します。

### 2.1.12. 個人ファイアウォール

**必要** なネットワークサービスを設定したら、ファイアウォールを実装することが重要です。



**重要**

必要なサービスを設定し、インターネットに接続する **前** に、または信頼していない他のネットワークに接続する前にファイアウォールを実装します。

ファイアウォールは、ネットワークパケットがシステムのネットワークインターフェースにアクセスするのを防ぎます。ファイアウォールによってブロックされているポートに対する要求が行われると、要

求は無視されます。サービスがブロックされたポートのいずれかをリッスンしている場合は、パケットを受信しず、効果的に無効になります。このため、ファイアウォールの設定時に使用されていないポートへのアクセスをブロックし、設定されたサービスが使用するポートへのアクセスをブロックしないようにしてください。

多くのユーザーでは、単純なファイアウォールを設定するのに最適なツールは、Red Hat Enterprise Linux の **Firewall Configuration Tool** (**system-config-firewall**)を含むグラフィカルなファイアウォール設定ツールです。このツールは、コントロールパネルインターフェースを使用して汎用ファイアウォールの幅広い **iptables** ルールを作成します。

このアプリケーションとその利用可能なオプション 「ファイアウォールの基本設定」 の使用方法は、を参照してください。

上級ユーザーおよびサーバー管理者は、でファイアウォールを手動で設定すること **iptables** が推奨されます。詳細は「ファイアウォール」を参照してください。**iptables** コマンドに関する包括的 「iptables」 なガイドについては、を参照してください。

### 2.1.13. セキュリティーの強化通信ツール

インターネットのサイズと好ましいことから、通信の傍受の脅威があります。今後、通信がネットワーク経由で転送される際に、通信を暗号化するツールが開発されています。

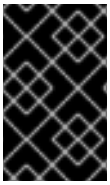
Red Hat Enterprise Linux 6 には、ネットワークを通過する際に情報を保護するために、高レベルかつ公開鍵暗号ベースの暗号化アルゴリズムを使用する 2 つの基本ツールが含まれています。

- **OpenSSH** - ネットワーク通信を暗号化する SSH プロトコルを自由に実装します。
- **GNU Privacy Guard(GPG)** データを暗号化する PGP（Pretty プライバシー）暗号化アプリケーションのフリー実装。

OpenSSH は、リモートマシンにアクセスして、やなどの古い暗号化されていないサービスに代わるより安全な方法です **telnet rsh**。OpenSSH には、と呼ばれるネットワークサービス **sshd** と 3 つのコマンドラインクライアントアプリケーションが含まれます。

- **ssh** : セキュアなリモートアクセスアクセスクライアント
- **scp** : セキュアなリモートコピーコマンド
- **sftp** : 対話式のファイル転送セッションを可能にするセキュアな擬似ソフトp クライアント。

OpenSSH 「セキュアなシェル」 の詳細は、を参照してください。



**重要**

**sshd** サービスは本質的に安全ですが、セキュリティーの脅威を防ぐためにサービスを最新の状態に維持 **する必要** があります。詳細は「セキュリティー更新」を参照してください。

GPG は、プライベートメール通信を確実にする 1 つの方法です。これは、パブリックネットワークを介して機密データを電子メールで送信し、ハードドライブの機密データを保護するために使用できます。

### 2.1.14. リムーバブルメディアの読み取り専用マウントの強制

（USB フラッシュディスクなど）リムーバブルメディアの読み取り専用マウントを強制するために、管

理者は **udev** ルールを使用してリムーバブルメディアを検出し、**blockdev** ユーティリティーを使用して読み取り専用にマウントするよう設定できます。Red Hat Enterprise Linux 6.7 以降、ファイルシステムの読み取り専用マウントを強制するために、特別なパラメーターを **udisks** ディスクマネージャーに渡すこともできます。

blockdev ユーティリティーをトリガーする **udev** ルールは、物理メディアの読み取り専用マウントに十分なものですが、**udisks** パラメーターを使用して、読み書きされたメディアにファイルシステムの読み取り専用マウントを実施することができます。

### blockdev を使用した、リムーバブルメディアの読み取り専用マウントの強制

すべてのリムーバブルメディアを読み取り専用にマウントするには、以下の内容が **80-readonly-removables.rules** 含まれる **/etc/udev/rules.d/** ディレクトリー（例：）に新しい **udev** 設定ファイルを作成します。

```
SUBSYSTEM=="block",ATTRS{removable}=="1",RUN{program}="/sbin/blockdev --setro %N"
```

上記の **udev** ルールは、**blockdev** ユーティリティーを使用して、新たに接続されたリムーバブルブロック（ストレージ）デバイスを自動的に読み取り専用として設定します。

### udisk を使用したファイルシステムの読み取り専用マウントの強制

すべてのファイルシステムを読み取り専用にマウントするには、**udev** で特別な **udisks** パラメーターを設定する必要があります。以下の内容を含む **/etc/udev/rules.d/** ディレクトリーにという名前の新規 **80-udisks.rules** の **udev** 設定ファイルを作成します（または、すでに存在する場合は以下の行を追加します）。

```
ENV{UDISKS_MOUNT_OPTIONS}="ro,noexec"
ENV{UDISKS_MOUNT_OPTIONS_ALLOW}="noexec,nodev,nosuid,atime,noatime,nodiratime,ro,sync,
dirsync"
```

デフォルトの **80-udisks.rules** ファイルは、**/lib/udev/rules.d/** ディレクトリーの udisks パッケージとともにインストールされていることに注意してください。このファイルには上記のルールが含まれていますが、コメントアウトされています。

上記の **udev** ルールは、**udisks** ディスクマネージャーに対し、ファイルシステムの読み取り専用マウントのみを許可するよう指示します。また、**noexec** パラメーターは、マウントされたファイルシステム上のバイナリーを直接実行するのを禁止します。このポリシーは、実際の物理デバイスがマウントされる方法に関わらず適用されます。つまり、ファイルシステムは、読み取り/書き込みのマウントされたデバイスでも読み取り専用でマウントされます。

### 新しい udev および udisk 設定の適用

この設定を有効にするには、新しい **udev** ルールを適用する必要があります。**udev** サービスは設定ファイルの変更を自動的に検出しますが、新しい設定は既存のデバイスには適用されません。新たに接続されたデバイスのみが、新しい設定の影響を受けます。したがって、接続されているリムーバブルメディアをすべてアンマウントして、新たに設定をプラグインしたときにそれらに適用されるようにする必要があります。

**udev** が既存のデバイスにすべてのルールを再適用するよう強制するには、**root** で次のコマンドを実行します。

```
~# udevadm trigger
```

**udev** が上記のコマンドを使用してすべてのルールを再適用するように強制すると、すでにマウントされているストレージデバイスには影響しないことに注意してください。

**udev** がすべてのルールを再読み込みするように強制するには、（何らかの理由で新しいルールが自動的に検出されない場合）、次のコマンドを使用します。

```
~# udevadm control --reload
```

## 2.2. サーバーセキュリティー

システムがパブリックネットワーク上のサーバーとして使用されると、攻撃のターゲットになります。このため、システムをハードニングし、サービスをロックダウンすることは、システム管理者にとって中程度の重要性があります。

特定の問題に対処する前に、サーバーセキュリティーの強化に関する以下の一般的なヒントを確認してください。

- すべてのサービスを最新の脅威から保護します。
- 可能な限りセキュアなプロトコルを使用します。
- 可能な場合は、マシンごとに1つのネットワークサービス種別のみを提供します。
- すべてのサーバーを注意して監視して、疑わしいアクティビティーを確認します。

### 2.2.1. TCP Wrapper および xinetd でのサービスのセキュリティー保護

*TCP Wrapper* は、さまざまなサービスに対してアクセス制御を提供します。SSH、Telnet、FTP などの最新のネットワークサービスは、受信要求と要求されたサービスとの間の保護を提供する TCP Wrapper を使用します。

TCP Wrapper が提供する利点は、追加のアクセス **xinetd**、ロギング、バインディング、リダイレクト、およびリソース使用制御を提供するスーパーサーバーです。

**注記**

TCP Wrapper とともに iptables ファイアウォールルールを使用し、サービスアクセス制御内に冗長性を作成 **xinetd** することが推奨されます。iptables コマンドを使用したファイアウォール 「ファイアウォール」 の実装の詳細は、を参照してください。

以下のサブセクションでは、各トピックの基本知識を想定し、特定のセキュリティーオプションに重点を置いています。

#### 2.2.1.1. TCP Wrapper によるセキュリティーの強化

TCP Wrapper は、サービスへのアクセスを拒否するよりもはるかに多くのことができます。このセクションでは、それらを使用して接続バナーの送信方法、特定のホストからの攻撃の警告、ロギング機能の強化方法を説明します。TCP Wrapper 機能および制御言語の詳細は、**hosts_options** man ページを参照してください。利用可能なフラグについては、オンラインで利用可能な **xinetd.conf** man ページを http://linux.die.net/man/5/xinetd.conf 参照してください。このフラグは、サービスに適用可能なオプションとして機能します。

##### 2.2.1.1.1. TCP Wrapper および接続バナー

ユーザーがサービスに接続する際に適切なバナーを表示することは、システム管理者が注意していることを攻撃者が把握できるようにするのに適した手段です。また、ユーザーに提示するシステムに関する情報を制御することもできます。サービスの TCP Wrappers バナーを実装するには、**banner** オプショ

ンを使用します。

この例では、のバナーを実装してい **vsftpd**ます。まずはバナーファイルを作成します。システムの任意の場所に指定できますが、デーモンと同じ名前を付ける必要があります。この例では、ファイルはという名前で **/etc/banners/vsftpd**、以下の行が含まれます。

```
220-Hello, %c
      220-All activity on ftp.example.com is logged.
      220-Inappropriate use will result in your access privileges being removed.
```

**%c** トークンは、ユーザー名、ホスト名、ユーザー名および IP アドレスなどのさまざまなクライアント情報を提供し、接続をさらに調整します。

このバナーを受信接続に表示するには、以下の行を **/etc/hosts.allow** ファイルに追加します。

```
vsftpd : ALL : banners /etc/banners/
```

#### 2.2.1.1.2. TCP Wrapper および Attack の警告

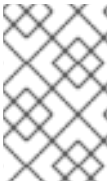
特定のホストまたはネットワークがサーバー攻撃を検出すると、TCP Wrapper を使用して、**spawn** ディレクティブを使用して、そのホストまたはネットワークからの後続の攻撃を管理者に警告できます。

この例では、206.182.68.0/24 ネットワークからの攻撃者がサーバー攻撃の試行を検出していることを前提としています。**/etc/hosts.deny** ファイルに以下の行を設定して、そのネットワークからの接続試行を拒否し、特別なファイルへの接続をログに記録します。

```
ALL : 206.182.68.0 : spawn /bin/echo `date` %c %d >> /var/log/intruder_alert
```

**%d** トークンは、攻撃者がアクセスしようとしているサービスの名前を提供します。

接続とログを許可するには、**spawn** ディレクティブを **/etc/hosts.allow** ファイルに置きます。



**注記**

**spawn** ディレクティブは shell コマンドを実行するため、管理者に通知したり、特定のクライアントがサーバーに接続しようとするときにコマンドを実行する特別なスクリプトを作成することが推奨されます。

#### 2.2.1.1.3. TCP Wrapper および強化されたロギング

特定の種類の接続が他の接続に問題がある場合は、**severity** オプションを使用して、そのサービスのログレベルを昇格できます。

この例では、FTP サーバーのポート 23（Telnet ポート）への接続を試みるユーザーが攻撃者であることを前提としています。これを指定するには、デフォルトの **emerg** フラグではなくログファイルにフラグを付け **info**、接続を拒否します。

これを行うには、に以下の行を追加し **/etc/hosts.deny**ます。

```
in.telnetd : ALL : severity emerg
```

これはデフォルトのロギングファシリティー **authpriv** を使用しますが、のデフォルト値から **info** に優先度が付けられ **emerg**、ログメッセージはコンソールに直接送信されます。

#### 2.2.1.2. xinetd によるセキュリティーの強化

本セクションでは、を使用 **xinetd** してトラップサービスを設定し、その **xinetd** サービスで使用できるリソースレベルを制御する方法を説明します。サービス *のリソース制限を設定すると、サービス拒否(DoS)を阻止することができます。*DoS）攻撃。**xinetd** およびの man ページは、利用可能なオプションの一覧 **xinetd.conf** を参照してください。

##### 2.2.1.2.1. トレースの設定

の重要な機能 **xinetd** は、ホストをグローバル **no_access** リストに追加する機能です。この一覧のホスト **xinetd** は、指定期間または再起動するまで、によって管理 **xinetd** されるサービスへの接続を拒否します。これは、**SENSOR** 属性を使用して実行できます。これは、サーバーでポートのスキャンを試行するホストを簡単にブロックする方法です。

を設定する最初のステップは、使用予定外のサービスを選択すること **SENSOR** です。この例では、Telnet を使用しています。

ファイルを編集し **/etc/xinetd.d/telnet**、**flags** 行を読み取りに変更します。

```
flags          = SENSOR
```

以下の行を追加します。

```
deny_time      = 30
```

これにより、ホストによってポートへの接続が 30 分間拒否されます。**deny_time** 属性の他の許容値は FOREVER です。これ **xinetd** は再起動するまで有効になる NEVER と、接続とログ記録を可能にする NEVER です。

最後に、最後の行は以下のようになります。

```
disable        = no
```

これにより、トラップ自体が有効になります。

を使用すると、望ましくないホストからの接続を検出して停止するのに **SENSOR** は適していますが、以下の 2 つの欠点があります。

- スキャンに対しては動作しません。
- が実行していることを認識している攻撃者 **SENSOR** は、IP アドレスを偽装し、禁止されているポートに接続することで、特定のホストに対してサービス攻撃をマウントできます。

##### 2.2.1.2.2. サーバーリソースの制御

のもう 1 つの重要な機能 **xinetd** は、制御下でサービスのリソース制限を設定することです。

これは、以下のディレクティブを使用してこれを行います。

- **cps = <number_of_connections> <wait_period>** : 受信接続の速度を制限します。このディレクティブは、以下の 2 つの引数を取ります。
  - **<number_of_connections>**: 処理する 1 秒あたりの接続数。受信接続の速度がこれよりも大きい場合、サービスは一時的に無効になっています。デフォルト値は gene(50)です。

- **<wait_period>** : 無効にした後にサービスを再度有効にするまで待機する秒数。デフォルトの間隔は 10（10 秒）です。

- **instances = <number_of_connections>** : サービスに許可される接続の合計数を指定します。このディレクティブは、整数またはのいずれかを受け入れ **UNLIMITED**ます。

- **per_source = <number_of_connections>** : 各ホストがサービスに許可される接続の数を指定します。このディレクティブは、整数またはのいずれかを受け入れ **UNLIMITED**ます。

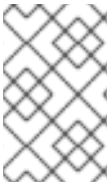
- **rlimit_as = <number[K|M]>** : サービスがキロバイトまたはメガバイトで占有できるメモリーアドレス空間の量を指定します。このディレクティブは、整数またはのいずれかを受け入れ **UNLIMITED**ます。

- **rlimit_cpu = <number_of_seconds>** : サービスが CPU を占有できる時間（秒単位）を指定します。このディレクティブは、整数またはのいずれかを受け入れ **UNLIMITED**ます。

このディレクティブを使用すると、単一の **xinetd** サービスがシステムに過度になり、サービス拒否が発生するのを防ぐことができます。

### 2.2.2. ポートマップのセキュリティー保護

**portmap** サービスは、NIS や NFS などの RPC サービスに対する動的ポート割り当てデーモンです。弱い認証メカニズムがあり、制御するサービスに幅広いポートを割り当てることができます。このため、セキュリティー保護は困難です。



**注記**

NFSv4 では必要がなくなったため、セキュリティー保護は NFSv2 および NFSv3 実装に **portmap** のみ影響します。NFSv2 サーバーまたは NFSv3 サーバーを実装する予定の場合 **portmap** は必須で、以下のセクションが適用されます。

RPC サービスを実行している場合は、以下の基本ルールに従います。

#### 2.2.2.1. TCP Wrapper によるポートマップの保護

TCP Wrapper を使用して、組み込み形式の認証がないため、**portmap** サービスにアクセスできるネットワークまたはホストを制限することが重要です。

さらに、サービスへのアクセスを制限する場合には、IP アドレス ***のみ*** を使用します。DNS ポイズニングおよびその他の方法で偽装できるので、ホスト名を使用しないでください。

#### 2.2.2.2. iptables を使用したポートマップの保護

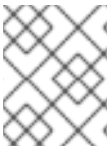
**portmap** サービスへのアクセスをさらに制限するには、サーバーに iptables ルールを追加し、特定のネットワークへのアクセスを制限することが推奨されます。

以下は、2 つの iptables コマンドの例です。192.168.0.0/24 ネットワークからポート 111（**portmap** サービスで使用される）への TCP 接続を最初に許可します。2 つ目は、ローカルホストから同じポートへの TCP 接続を許可します。これは、**Nautilus** が使用する **sgi_fam** サービスに必要です。その他のパケットはすべてドロップされます。

```
~]# iptables -A INPUT -p tcp -s ! 192.168.0.0/24 --dport 111 -j DROP
~]# iptables -A INPUT -p tcp -s 127.0.0.1 --dport 111 -j ACCEPT
```

同様に UDP トラフィックを制限するには、以下のコマンドを使用します。

```
~]# iptables -A INPUT -p udp -s ! 192.168.0.0/24 --dport 111 -j DROP
```



**注記**

iptables コマンドを使用したファイアウォール 「ファイアウォール」 の実装の詳細は、を参照してください。

### 2.2.3. NIS のセキュリティー保護

*ネットワーク情報サービス*（）NIS）は、と呼ばれる RPC サービスです。このサービスは **ypserv**、ユーザー名、パスワード、**portmap** およびその他の機密情報をドメイン内に存在するコンピューターにマップを配信するために使用します。

NIS サーバーは、複数のアプリケーションで構成されます。これには、以下が含まれます。

- **/usr/sbin/rpc.yppasswdd** : **yppasswdd** サービスとも呼ばれ、このデーモンは NIS パスワードを変更できます。
- **/usr/sbin/rpc.ypxfrd** : **ypxfrd** サービスとも呼ばれます。このデーモンは、ネットワーク経由で NIS マップ転送を行います。
- **/usr/sbin/yppush** : このアプリケーションは、変更した NIS データベースを複数の NIS サーバーに伝播します。
- **/usr/sbin/ypserv** : これは NIS サーバーデーモンです。

NIS は、現在の標準ではセキュリティー保護されていません。ホストの認証メカニズムがなく、パスワードハッシュなど、ネットワーク経由ですべての情報が暗号化されていない状態で送信されます。そのため、NIS を使用するネットワークを設定する際には、極端な注意が必要です。これは、NIS のデフォルト設定が本質的に安全ではないため、さらに複雑になります。

で説明されているように、NIS サーバーの実装を計画している場合には、まず **portmap** サービスのセキュリティーを確保し 「ポートマップのセキュリティー保護」、ネットワークプランニングなどの以下の問題に対処することが推奨されます。

#### 2.2.3.1. ネットワークの慎重に計画

NIS はネットワーク経由で機密情報を暗号化せずに送信するため、サービスはファイアウォールの内側と、セグメント化されたセキュアなネットワークで実行することが重要です。NIS 情報が安全ではないネットワーク上で送信されるたびに、傍受されるリスクがあります。ネットワーク設計は、深刻なセキュリティー侵害を防ぐのに役立ちます。

#### 2.2.3.2. パスワードのような NIS ドメイン名およびホスト名の使用

NIS ドメイン内のマシンは、ユーザーが NIS サーバーの DNS ホスト名と NIS ドメイン名を認識している限り、コマンドを使用して認証なしでサーバーから情報を抽出できます。

たとえば、ラップトップコンピューターをネットワークに接続するか、ネットワークを外部（また内部 IP アドレスに管理）から侵入した場合、以下のコマンドは **/etc/passwd** マップを明確にします。

```
ypcat -d <NIS_domain> -h <DNS_hostname> passwd
```

この攻撃者が root ユーザーであれば、以下のコマンドを入力して **/etc/shadow** ファイルを取得できます。

```
ypcat -d <NIS_domain> -h <DNS_hostname> shadow
```

**注記**

Kerberos を使用すると、**/etc/shadow** ファイルは NIS マップに保存されません。

攻撃者が NIS マップにアクセスするには、などの DNS ホスト名のランダムな文字列を作成し **o7hfawtgmhwg.domain.com**ます。同様に、**異なる** 無作為な NIS ドメイン名を作成します。これにより、攻撃者が NIS サーバーにアクセスするのが非常に困難になります。

#### 2.2.3.3. /var/yp/securenets ファイルの編集

**/var/yp/securenets** ファイルが空白であるか、または存在しない場合（デフォルトインストール後など）、NIS は全ネットワークをリッスンします。最初に行うべきことは、ネットマスク/ネットワークのペアをファイルに追加して、適切なネットワークからの要求に **ypserv** のみ応答するようにすることです。

以下は、**/var/yp/securenets** ファイルからのエントリーの例です。

```
255.255.255.0     192.168.0.0
```

**警告**

**/var/yp/securenets** ファイルを作成せずに、NIS サーバーを最初に起動しないでください。

この技術は、IP スプーフィング攻撃に対する保護は提供しませんが、NIS サーバーサービスがどのネットワークにでも制限されます。

#### 2.2.3.4. 静的ポートの割り当ておよび iptables ルールの使用

NIS に関連するサーバーはすべて、ユーザーがログインパスワードを変更できるようにするデーモン以外 **rpc.yppasswdd** の特定のポートを割り当てることができます。他の 2 つの NIS サーバーデーモンにポートを割り当てる **rpc.ypxfrd** と **ypserv**、ファイアウォールルールの作成により、NIS サーバーデーモンが侵入者からさらに保護されます。

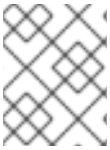
これを行うには、に以下の行を追加し **/etc/sysconfig/network**ます。

```
YPSERV_ARGS="-p 834"
YPXFRD_ARGS="-p 835"
```

以下の iptables ルールは、サーバーがこれらのポートをリッスンするネットワークを強制するために使用できます。

```
~]# iptables -A INPUT -p ALL -s ! 192.168.0.0/24 --dport 834 -j DROP
~]# iptables -A INPUT -p ALL -s ! 192.168.0.0/24 --dport 835 -j DROP
```

つまり、サーバーはプロトコルに関係なく、リクエストが 192.168.0.0/24 ネットワークからの場合に 834 および 835 への接続のみを許可することを意味します。



**注記**

iptables コマンドを使用したファイアウォール 「ファイアウォール」 の実装の詳細は、を参照してください。

#### 2.2.3.5. Kerberos 認証の使用

認証に NIS を使用する場合に考慮すべき問題の 1 つは、ユーザーがマシンにログインするたびに、**/etc/shadow** マップのパスワードハッシュがネットワーク経由で送信されることです。侵入者が NIS ドメインにアクセスし、ネットワークトラフィックを傍受する場合、ユーザー名とパスワードハッシュを収集できます。十分な時間があれば、パスワードクラッキングプログラムにより、パスワードの弱いパスワードが推測され、攻撃者はネットワークで有効なアカウントにアクセスできるようになります。

Kerberos は秘密鍵の暗号化を使用するため、ネットワーク上でパスワードハッシュが送信されないため、システムがより安全になりました。Kerberos の詳細は、「『シングルサインオンおよびスマートカード の管理』 」を参照してください。

### 2.2.4. NFS のセキュア化

**重要**

Red Hat Enterprise Linux 6、NFSv4 に含まれる NFS のバージョンは、で説明されているように **portmap** サービスを必要としなくなりました 「ポートマップのセキュリティー保護」。NFS トラフィックは、UDP ではなくすべてのバージョンで TCP を使用するため、NFSv4 を使用する場合はこれを必要とします。NFSv4 には、**RPCSEC_GSS** カーネルモジュールの一部として Kerberos ユーザーおよびグループ認証が含まれるようになりました。の情報 **portmap** は、Red Hat Enterprise Linux 6 では NFSv2 および NFSv3 をサポートしており、どちらも利用されています **portmap**。

#### 2.2.4.1. ネットワークの慎重に計画

NFSv2 および NFSv3 は従来、安全でないデータを渡していました。NFS のすべてのバージョンでは、Kerberos を使用して通常のファイルシステム操作を認証（およびオプションで暗号化）できるようになりました。NFSv4 では、すべての操作で Kerberos を使用できますが、v2 または v3 では、ファイルのロックとマウントは使用されません。NFSv4.0 を使用する場合は、クライアントが NAT またはファイアウォールの背後にある場合に委譲をオフにできます。委譲が NAT およびファイアウォールを通過できるようにするための NFSv4.1 の使用についての情報は、『『ストレージ管理ガイド』』 の pNFS に関するセクションを参照してください。

#### 2.2.4.2. NFS マウントオプションのセキュリティー保護

**/etc/fstab** ファイルの **mount** コマンドの使用方法については、『ストレージ管理ガイド』で説明しています。セキュリティー管理からは、NFS マウントオプションもで指定できるので **/etc/nfsmount.conf**、カスタムのデフォルトオプションを設定するのに使用できることに注意してください。

#### 2.2.4.2.1. NFS サーバーの確認

**警告**

ファイルシステム全体のみをエクスポートします。ファイルシステムのサブディレクトリーのエクスポートは、セキュリティー上の問題となる可能性があります。クライアントが、エクスポートしたファイルシステムのエクスポートした部分を「縮小」し、エクスポートされていない部分になる場合があります（ **exports(5)** man ページのサブツリーチェックのセクションを参照してください）。

マウントしたファイルシステムに書き込むことができるユーザー数を減らして、可能な限りファイルシステムを読み取り専用としてエクスポートする場合は、**ro** オプションを使用します。**rw** オプションは特に必要な場合にのみ使用します。詳細はの man **exports(5)** ページを参照してください。書き込みアクセスを許可すると、シンボリックリンク攻撃などのリスクが高まります。これには、やなどの一時ディレクトリーが含ま **/tmp** れ **/usr/tmp**ます。

ディレクトリーを **rw** オプションでマウントする必要があると、リスクを軽減できる限り全面的に書き込みができないようにする必要があります。ホームディレクトリーのエクスポートは、一部のアプリケーションはクリアテキストでパスワードを保存するか、または暗号化されていないものにするため、リスクとして見られています。アプリケーションコードの確認および改善により、このリスクが軽減されます。ユーザーが SSH 鍵にパスワードを設定しないため、ホームディレクトリーもリスクを生じさせることになります。パスワードの使用または Kerberos の使用により、このリスクが軽減されます。

アクセスが必要なクライアントにのみエクスポートを制限します。NFS サーバーで **showmount -e** コマンドを使用して、サーバーがエクスポートしている内容を確認します。特別に必要のないものはエクスポートしないでください。

**no_root_squash** オプションを使用して、既存のインストールを確認し、インストールが使用されていないことを確認してください。詳細は「オプションを使用し **no_root_squash** ない」を参照してください。

**secure** オプションは、エクスポートを制限するために使用されるサーバー側のエクスポートオプションです。 「予備」 ポート。デフォルトでは、サーバーは、クライアントの通信のみを許可します。「予備」 従来のクライアントが許可されるのは 1024 未満のポート（ポート番号が 1024 未満）「trusted」 これらのポートを使用するコード（カーネル内の NFS クライアントなど）。ただし、多くのネットワークでは、一部のクライアントでルートとなるのは困難ではないため、予約済みポートからの通信が特権であることをサーバーが想定しても安全ではありません。したがって、予約ポートの制限は限定的な値であるため、特定のクライアントへの Kerberos、ファイアウォール、およびエクスポートの制限に依存します。

ほとんどのクライアントは、可能な場合は予約ポートを使用します。ただし、予約ポートは限定的なリソースであるため、クライアント（特に NFS マウントが多数ある場合）は、番号が大きいポートも使用するように選択できます。Linux クライアントは、を使用してこれを行うことができます。「noresvport」 マウントオプション。エクスポートでこれを許可する場合は、でその操作を行うことができます。 「insecure」 エクスポートオプション。

ユーザーがサーバーへのログインを許可しないことが推奨されます。NFS サーバーの上記の設定を確認する際に、サーバーにアクセスできるユーザーと何を確認します。

#### 2.2.4.2.2. NFS クライアントの確認

setuid プログラムの使用を無効にするには、**nosuid** オプションを使用します。**nosuid** オプションは、**set-user-identifier** または **set-group-identifier** ビットを無効にします。これにより、リモートユーザーが setuid プログラムを実行してより高い権限を取得できなくなります。クライアントとサーバー側でこのオプションを使用します。

**noexec** オプションは、クライアント上の実行ファイルをすべて無効にします。このパラメーターを使用して、ユーザーがファイルシステムを共有するファイルを誤って実行するのを防ぎます。**nosuid** および **noexec** オプションは、ほとんどのファイルシステム（すべてではない場合は）の標準オプションです。

**nodev** オプションを指定して回避します。 「device-files」 クライアントがハードウェアデバイスとして処理できないようにします。

**resvport** オプションはクライアント側のマウントオプションで、対応するサーバー側のエクスポートオプション **secure** です（上記の説明を参照してください）。「予約されたポート」への通信を制限します。予約済みまたは「well known」ポートは、root ユーザーなどの特権ユーザーやプロセス用に予約されます。このオプションを設定すると、クライアントは予約済みソースポートを使用してサーバーと通信します。

NFS のすべてのバージョンが、Kerberos 認証でのマウントに対応するようになりました。これを有効にするマウントオプションはです **sec=krb5**。

NFSv4 は、整合性と **krb5p** プライバシー保護 **krb5i** のためのを使用した Kerberos によるマウントをサポートします。これらは **sec=krb5**、でマウントするときに使用されますが、NFS サーバーで設定する必要があります。詳細は、エクスポートの man ページ(**man 5 exports**)を参照してください。

NFS の man ページ(**man 5 nfs**)には、があります。 「セキュリティーに関する考慮事項」 セクションには、NFSv4 のセキュリティー強化と、すべての NFS 固有のマウントオプションが含まれています。

#### 2.2.4.3. 構文エラーに注意してください。

NFS サーバーは、エクスポートするファイルシステムと、**/etc/exports** ファイルを参照してこのディレクトリーをエクスポートするホストを決定します。このファイルを編集する際には、余分なスペースを追加しないでください。

たとえば、**/etc/exports** ファイルの以下の行は、ディレクトリーを読み取り/書き込み権限を **bob.example.com** 持つホスト **/tmp/nfs/** に共有します。

```
/tmp/nfs/     bob.example.com(rw)
```

一方、**/etc/exports** ファイルの以下の行は、同じディレクトリーを **bob.example.com** 読み取り専用パーミッション **で** ホストに共有し、ホスト名の後に 1 つのスペース文字により、読み取り/書き込み権限のあるユーザーに共有します。

```
/tmp/nfs/     bob.example.com (rw)
```

**showmount** コマンドを使用して、共有内容を確認することで、設定された NFS 共有を確認することが推奨されます。

```
showmount -e <hostname>
```

#### 2.2.4.4. オプションを使用し no_root_squash ない

デフォルトでは、NFS 共有は root ユーザーを、権限のない **nfsnobody** ユーザーアカウントであるユーザーに変更します。これにより、すべてのルート作成されたファイルの所有者がに変更されます。これにより **nfsnobody**、setuid ビットセットでのプログラムのアップロードができなくなります。

を使用すると、リモートの root ユーザー **no_root_squash** は共有ファイルシステム上の任意のファイルを変更し、他のユーザーが誤って実行されるよう、トリックの木に関するアプリケーションを残すことができます。

#### 2.2.4.5. NFS ファイアウォールの設定

NFS に使用されるポートは rpcbind により動的に割り当てられます。これは、ファイアウォールルールの作成時に問題が発生する可能性があります。このプロセスを単純化するには、**/etc/sysconfig/nfs** ファイルを使用して、使用するポートを指定します。

- **MOUNTD_PORT** : mountd(rpc.mountd)の TCP ポートおよび UDP ポート
- **STATD_PORT** : ステータスの TCP ポートおよび UDP ポート(rpc.statd)
- **LOCKD_TCPPORT** : nlockmgr(rpc.lockd)の TCP ポート
- **LOCKD_UDPPORT** - UDP ポート nlockmgr(rpc.lockd)

指定したポート番号は、他のサービスでは使用できません。ファイアウォールを、指定したポート番号と TCP および UDP ポート 2049(NFS)を許可するように設定します。

NFS サーバーで **rpcinfo -p** コマンドを実行し、使用しているポートおよび RPC プログラムを確認します。

### 2.2.5. Apache HTTP サーバーのセキュア化

Apache HTTP Server は、Red Hat Enterprise Linux に同梱される最も安定し安全なサービスの 1 つです。Apache HTTP Server のセキュリティーを保護するためには多くのオプションや手法が利用できます。ここでは、多くのオプションや手法が展開されます。以下のセクションでは、Apache HTTP Server の実行時にの適切なプラクティスを簡単に説明します。

システムで実行中のスクリプトが、実稼働環境に入る **前** に目的どおりに機能することを確認してください。また、スクリプトまたは CGI を含むディレクトリーに書き込み権限があるのは root ユーザーのみです。これを行うには、root で以下のコマンドを実行します。

```
chown root <directory_name>
```

```
chmod 755 <directory_name>
```

システム管理者は、以下の設定オプションを使用する場合は注意してください（で設定 **/etc/httpd/conf/httpd.conf**）。

**FollowSymLinks**

このディレクティブはデフォルトで有効になっているため、Web サーバーのドキュメントルートへのシンボリックリンクを作成する場合は注意してください。たとえば、にシンボリックリンクを提供することは適切ではありません /。

**Indexes**

このディレクティブはデフォルトで有効になっていますが、望ましいとは限りません。サーバーでファイルを参照できないようにするには、このディレクティブを削除します。

**UserDir**

システムにユーザーアカウントが存在することを確認することができるため、**UserDir** ディレクティブはデフォルトで無効になります。サーバーでユーザーディレクトリーの閲覧を有効にするには、以下のディレクティブを使用します。

```
UserDir enabled
UserDir disabled root
```

これらのディレクティブは、以外の全ユーザーディレクトリーを検索するユーザーディレクトリーをアクティブにし /**root**/ます。無効化されたアカウントの一覧にユーザーを追加するには、**UserDir disabled** 行にユーザーのスペースで区切られたリストを追加します。

**ServerTokens**

**ServerTokens** ディレクティブは、クライアントに返すサーバー応答ヘッダーフィールドを制御します。これには、以下のパラメーターを使用してカスタマイズできるさまざまな情報が含まれています。

- **ServerTokens Full**（デフォルトオプション）: 以下のような利用可能なすべての情報（OS タイプおよび使用されるモジュール）を提供します。

  ```
  Apache/2.0.41 (Unix) PHP/4.2.2 MyMod/1.2
  ```

- **ServerTokens Prod** または： 以下の情報を **ServerTokens ProductOnly** 提供します。

  ```
  Apache
  ```

- **ServerTokens Major**: 以下の情報を提供します。

  ```
  Apache/2
  ```

- **ServerTokens Minor**: 以下の情報を提供します。

  ```
  Apache/2.0
  ```

- **ServerTokens Min** または： 以下の情報を **ServerTokens Minimal** 提供します。

  ```
  Apache/2.0.41
  ```

- **ServerTokens OS**: 以下の情報を提供します。

  ```
  Apache/2.0.41 (Unix)
  ```

攻撃者がお使いのシステムに関する有用な情報を取得しないように、この **ServerTokens Prod** オプションを使用することが推奨されます。

**重要**

**IncludesNoExec** ディレクティブを削除しないでください。デフォルトでは、 *サーバー側の包含(* )SSI）モジュールはコマンドを実行できません。絶対的に必要でない限り、攻撃者がシステム上でコマンドを実行するのを有効にしない限り、この設定は変更しないことが推奨されます。

**httpd モジュールの削除**
特定のシナリオでは、HTTP サーバーの機能を制限するために特定の **httpd** モジュールを削除することが利点があります。これを行うには、**/etc/httpd/conf/httpd.conf** ファイルで削除するモジュールを読み込む行全体をコメントアウトします。たとえば、プロキシーモジュールを削除するには、ハッシュ記号でプリペンドして以下の行をコメントアウトします。

```
#LoadModule proxy_module modules/mod_proxy.so
```

**/etc/httpd/conf.d/** ディレクトリーには、モジュールの読み込みに使用する設定ファイルが含まれていることに注意してください。

**httpd および SELinux**
Apache HTTP Server および SELinux 『の詳細は、『Managing Confined Services Guide』を参照してください』。

### 2.2.6. FTP のセキュリティー保護

*ファイル転送プロトコル（)* FTP）は、ネットワーク上でファイルを転送するために設計された古い TCP プロトコルです。ユーザー認証を含むサーバーでのすべてのトランザクションは暗号化されないため、安全ではないプロトコルと見なされ、注意して設定する必要があります。

Red Hat Enterprise Linux は、3 つの FTP サーバーを提供します。

- **gssftpd** : ネットワーク経由で認証情報を送信しない Kerberos **xinetd**ベースの FTP デーモン。
- **Red Hat コンテンツフレーム** (**tux**)- FTP 機能のあるカーネル空間の Web サーバーです。
- **vsftpd** : FTP サービスのスタンドアロンのセキュリティー指向の実装。

**vsftpd** FTP サービスを設定する際には、以下のセキュリティーガイドラインが挙げられます。

#### 2.2.6.1. FTP Greetingbanner

ユーザー名とパスワードを送信する前に、すべてのユーザーに greeting バナーが表示されます。デフォルトでは、このバナーには、システム内でのネゴシエーターの特定を試みる攻撃者が役立つバージョン情報が含まれています。

の greeting バナーを変更するには **vsftpd**、以下のディレクティブを **/etc/vsftpd/vsftpd.conf** ファイルに追加します。

```
ftpd_banner=<insert_greeting_here>
```

上記のディレクティブの *<insert_greeting_here>* を greeting メッセージのテキストに置き換えます。

mutli-line バナーの場合は、バナーファイルを使用することが推奨されます。複数のバナーの管理を簡素化するには、すべてのバナーをという名前の新しいディレクトリーに配置し **/etc/banners/**ます。この例では、FTP 接続のバナーファイルはです **/etc/banners/ftp.msg**。以下は、このようなファイルの例です。

```
######### Hello, all activity on ftp.example.com is logged. #########
```

**注記**

で指定されている **220** ように、ファイルの各行を開始する必要はありません 「TCP Wrapper および接続バナー」。

この greeting バナーファイルを参照するには **vsftpd**、以下のディレクティブを **/etc/vsftpd/vsftpd.conf** ファイルに追加します。

```
banner_file=/etc/banners/ftp.msg
```

また、で説明されているように TCP Wrappers を使用して、追加のバナーを受信接続に送信することもでき 「TCP Wrapper および接続バナー」 ます。

#### 2.2.6.2. Anonymous Access

**/var/ftp/** ディレクトリーが存在すると、匿名アカウントがアクティブになります。

このディレクトリーを作成するための最も簡単な方法は、**vsftpd** パッケージをインストールすることです。このパッケージは、匿名ユーザーのディレクトリーツリーを確立し、匿名ユーザーの読み取り専用にディレクトリーの権限を設定します。

デフォルトでは、匿名ユーザーはどのディレクトリーにも書き込みできません。

**警告**

FTP サーバーへの匿名アクセスを有効にする場合は、機密データの保存場所に注意してください。

**手順2.1 Anonymous Upload**

1. 匿名ユーザーがファイルをアップロードできるようにするには、ディレクトリーに書き込み専用ディレクトリーを作成することが推奨され **/var/ftp/pub/** ます。root で以下のコマンドを実行して、という名前のそのディレクトリーを作成し **/upload**/ます。

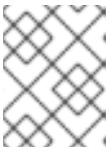
   ```
   ~]# mkdir /var/ftp/pub/upload
   ```

2. 次に、匿名ユーザーがディレクトリーのコンテンツを表示できないようにパーミッションを変更します。

   ```
   ~]# chmod 730 /var/ftp/pub/upload
   ```

   ディレクトリーの長い形式のリストは以下のようになります。

   ```
   ~]# ls -ld /var/ftp/pub/upload
   drwx-wx---. 2 root ftp 4096 Nov 14 22:57 /var/ftp/pub/upload
   ```



**注記**

管理者は、匿名ユーザーがディレクトリーへの読み取りおよび書き込みを許可する場合は多くの場合、サーバーがソフトウェアのリポジトリーになります。

3. セクションで **vsftpd**、以下の行を **/etc/vsftpd/vsftpd.conf** ファイルに追加します。

```
anon_upload_enable=YES
```

4. Red Hat Enterprise Linux では、SELinux はデフォルトで Enforcing モードで実行されています。したがって、**vsftpd** がファイルをアップロードできるようにするには、**allow_ftpd_anon_write** ブール値を有効にする必要があります。

```
~]# setsebool -P allow_ftpd_anon_write=1
```

5. **/upload/** ディレクトリーに **public_content_rw_t** SELinux コンテキストでラベルを付けます。

```
~]# semanage fcontext -a -t public_content_rw_t '/var/ftp/pub/upload(/.*)'
```

**注記**

**semanage** ユーティリティーは、policycoreutils-python パッケージにより提供され、デフォルトではインストールされません。インストールするには、root で次のコマンドを実行します。

```
~]# yum install policycoreutils-python
```

6. **restorecon** ユーティリティーを使用して **/upload/**、とそのファイルのタイプを変更します。

```
~]# restorecon -R -v /var/ftp/pub/upload
```

ディレクトリーには **public_content_rw_t** が適切にラベル付けされ、SELinux が Enforcing モードで適切にラベル付けされ、匿名ユーザーがファイルをアップロードできるようになりました。

```
~]$ ls -dZ /var/ftp/pub/upload
drwx-wx---. root root unconfined_u:object_r:public_content_t:s0 /var/ftp/pub/upload/
```

SELinux の使用に関する詳細は、『Security Enhanced Linux ユーザーガイド』および『Confined Services』ガイドを参照してください。

#### 2.2.6.3. ユーザーアカウント

FTP は、認証用にセキュリティー保護されていないネットワーク上で暗号化されていないユーザー名とパスワードを送信するため、ユーザーアカウントからサーバーへのアクセスを拒否することが推奨されます。

にあるすべてのユーザーアカウントを無効にするには **vsftpd**、に以下のディレクティブを追加し **/etc/vsftpd/vsftpd.conf**ます。

```
local_enable=NO
```

#### 2.2.6.3.1. ユーザーアカウントの制限

root ユーザーや **sudo** 特権のあるアカウントなど、特定アカウントまたはアカウントの特定のグループの FTP アクセスを無効にするには、の説明に従って PAM リストファイルを使用するのが最も簡単な方法です 「Root アクセスの拒否」。の PAM 設定ファイル **vsftpd** はです **/etc/pam.d/vsftpd**。

また、各サービス内のユーザーアカウントを直接無効にすることもできます。

で特定のユーザーアカウントを無効にするに **vsftpd**は、ユーザー名をに追加します。 **/etc/vsftpd/ftpusers**

### 2.2.6.4. TCP Wrapper を使用した制御アクセスの使用

に従って、TCP Wrappers を使用して FTP デーモンへのアクセスを制御し 「TCP Wrapper によるセキュリティーの強化」 ます。

## 2.2.7. Postfix のセキュリティー保護

Postfix は、SMTP(Simple Mail Transfer Protocol)を使用して他の MTA と電子メールクライアントまたは配信エージェントとの間で電子メッセージを送信するメール転送エージェント(MTA)です。多くの MTA は相互にトラフィックを暗号化できますが、ほとんどはないため、パブリックネットワークを介して電子メールを送信することは本質的に安全でない通信とみなされます。

Postfix サーバーの実装を計画しているお客様は、以下の問題に対応することが推奨されます。

### 2.2.7.1. サービス拒否攻撃の制限

電子メールの性質上、決定された攻撃者はメールでサーバーをあふれることができ、サービス拒否を引き起こす可能性があります。このような攻撃の効果は、**/etc/postfix/main.cf** ファイルでディレクティブの制限を設定することで制限できます。すでに存在するディレクティブの値を変更したり、必要な値を以下の形式で追加したりできます。

```
<directive> = <value>
```

サービス拒否攻撃を制限するために使用できるディレクティブの一覧を以下に示します。

- **smtpd_client_connection_rate_limit**: クライアントが時間単位ごとにこのサービスに送信できる最大接続試行回数（以下で説明します）。デフォルト値は 0 で、Postfix が許可されるため、クライアントは時間単位ごとに接続を行うことができます。デフォルトでは、信頼されるネットワークのクライアントは除外されます。

- **anvil_rate_time_unit**: この時間単位は、レート制限の計算に使用されます。デフォルト値は 60 秒です。

- **smtpd_client_event_limit_exceptions**: 接続および流量制御コマンドから除外されるクライアント。デフォルトでは、信頼されるネットワークのクライアントは除外されます。

- **smtpd_client_message_rate_limit**: クライアントが時間単位あたりにリクエストできるメッセージの最大数（Postfix が実際にこれらのメッセージを受け入れるかどうかは注意してください）。

- **default_process_limit**: 指定のサービスを提供する Postfix 子プロセスの最大数。この制限は、**master.cf** ファイル内の特定のサービスに対して上書きすることが可能です。デフォルト値は 100 です。

- **queue_minfree**: メールを受信するのに必要なキューファイルシステムの最小空き領域（バイ

ト単位）。これは現在 Postfix SMTP サーバーで、任意のメールを受け入れるかどうかを決めます。デフォルトでは、Postfix SMTP サーバーは、message_size_limit の空き領域が 1.5 未満になると **MAIL FROM** コマンドを拒否します。空き領域の上限をより高く指定するには、queue_minfree 値を指定します。最低でも 1.5 倍の message_size_limit を指定します。デフォルトでは queue_minfree の値は 0 です。

- **header_size_limit** : メッセージヘッダーを保存するメモリーの最大量（バイト単位）。ヘッダーが大きい場合、余分は破棄されます。デフォルト値は 102400 です。
- **message_size_limit** : 重要情報を含む、メッセージの最大サイズ（バイト単位）。デフォルト値は 10240000 です。

#### 2.2.7.2. NFS および Postfix

NFS 共有ボリュームにメールスプールディレクトリーを **/var/spool/postfix/**配置することはありません。

NFSv2 および NFSv3 はユーザーおよびグループ ID に対する制御を維持しないため、複数のユーザーが同じ UID を使用し、相互のメールを受信して読み込むことができます。

**注記**

**SECRPC_GSS** カーネルモジュールは UID ベースの認証を使用しないため、Kerberos を使用する NFSv4 では、これは当てはまりません。ただし、引き続き、NFS 共有ボリュームにメールスプールディレクトリーを配置し **ない** ことが推奨されます。

#### 2.2.7.3. メールのみのユーザー

Postfix サーバーでローカルユーザーが悪用しないようにするには、メールユーザーが電子メールプログラムを使用して Postfix サーバーのみにアクセスすることが推奨されます。メールサーバーのシェルアカウントは許可されず、**/etc/passwd** ファイル内のすべてのユーザーシェルは（root ユーザー **/sbin/nologin** を除く）に設定する必要があります。

#### 2.2.7.4. Postfix ネットワークリスティングの無効化

デフォルトでは、Postfix はローカルのループバックアドレスのみをリッスンするように設定されています。これは、ファイルを表示して確認でき **/etc/postfix/main.cf**ます。

ファイル **/etc/postfix/main.cf** を表示して、以下の ***inet_interfaces*** 行のみが表示されることを確認します。

```
inet_interfaces = localhost
```

これにより、Postfix はネットワークからではなく、ローカルシステムからのメールメッセージ（cron ジョブレポートなど）のみを受け入れるようになります。これはデフォルト設定で、ネットワーク攻撃から Postfix を保護します。

ローカルホストの制限を削除し、Postfix がすべてのインターフェースをリッスンできるようにするため、***inet_interfaces = all*** 設定は使用できます。

#### 2.2.7.5. Postfix が SASL を使用するよう設定

**Postfix** の Red Hat Enterprise Linux バージョンでは、 *SMTP 認証（または SMTP AUTH）*に **Dovecot** または **Cyrus SASL** 実装を使用できます。SMTP 認証は、**簡易メール転送プロトコル** の拡張です。これを有効にすると、サーバーと **クライアント** の両方がサポートおよび許可される認証方法を使用して

**SMTP** クライアントを認証する必要があります。本セクションでは、**Dovecot SASL** 実装を使用するように **Postfix** を設定する方法を説明します。

**Dovecot POP/IMAP** サーバーをインストールして、システムで **Dovecot SASL** 実装を使用できるようにするには、**root** ユーザーとして以下のコマンドを実行します。

```
~]# yum install dovecot
```

**Postfix SMTP** サーバーは、*UNIX-domain ソケットまたは TCP ソケット* のいずれかを使用して **Dovecot SASL** 実装と通信できます。**Postfix** アプリケーションと **Dovecot** アプリケーションが別のマシンで実行されている場合のみ、最後のメソッドが必要になります。本ガイドでは、より優れたプライバシーを提供する UNIX-domain ソケットメソッドを優先します。

**Postfix** に **Dovecot SASL** 実装を使用するように指示するには、両方のアプリケーションに対して多くの設定変更を実行する必要があります。以下の手順に従って、これらの変更を適用します。

**Dovecot の設定**

1. 主な **Dovecot** 設定ファイルを変更し **/etc/dovecot/conf.d/10-master.conf**、以下の行を追加します（デフォルトの設定ファイルには関連するセクションの大半が含まれ、行はコメント解除する必要があります）。

```
service auth {
  unix_listener /var/spool/postfix/private/auth {
    mode = 0660
    user = postfix
    group = postfix
  }
}
```

上記の例では、**Postfix** と **Dovecot** 間の通信に UNIX-domain ソケットを使用していることを前提としています。また、**/var/spool/postfix/** ディレクトリーにあるメールキューと **postfix** ユーザーおよびグループで実行されているアプリケーションなど、**Postfix SMTP** サーバーのデフォルト設定も前提としています。このようにして、読み取り/書き込みパーミッションは **postfix** ユーザーおよびグループに限定されます。

以下の設定を使用して **Dovecot** を設定して、**TCP** 経由で **Postfix** 認証リクエストをリッスンするようにできます。

```
service auth {
  inet_listener {
    port = 12345
  }
}
```

上記の例では、は使用するポートの数に **12345** 置き換えてください。

2. **/etc/dovecot/conf.d/10-auth.conf** 設定ファイルを編集して **Dovecot** に **plain** および **login** 認証メカニズムを使用して **Postfix SMTP** サーバーを提供するよう指示します。

```
auth_mechanisms = plain login
```

**Postfix の設定**

**Postfix** の場合には、主要な設定ファイルのみを変更する **/etc/postfix/main.cf**必要があります。以下の設定ディレクティブを追加または編集します。

1. **Postfix** SMTP**サーバー** で SMTP 認証を有効にします。

```
smtpd_sasl_auth_enable = yes
```

2. **Postfix** が SMTP 認証に **Dovecot SASL** 実装を使用するように指示します。

```
smtpd_sasl_type = dovecot
```

3. **Postfix** キューディレクトリーに対する認証パスを提供します（相対パスを使用すると、**Postfix** サーバーが **chroot** で実行しているかどうかに関わらず、設定が確実に機能します）。

```
smtpd_sasl_path = private/auth
```

この手順では、**Postfix** と **Dovecot** 間の通信に UNIX-domain ソケットを使用することを前提としています。通信に TCP ソケットを使用する場合に、通信に **TCP** ソケットを使用する場合に **Postfix** が別のマシンで **Dovecot** を検索するようにするには、以下のような設定値を使用します。

```
smtpd_sasl_path = inet:127.0.0.1:12345
```

上記の例では、を **Dovecot マシンの IP アドレスと Dovecot** の **/etc/dovecot/conf.d/10-master.conf** 設定ファイル **12345** で指定したポートで置き換える **127.0.0.1** 必要 **が**あります。

4. **Postfix SMTP** サーバーがクライアントで利用可能にする **SASL** メカニズムを指定します。暗号化セッションと暗号化されていないセッションには、異なるメカニズムを指定できることに注意してください。

```
smtpd_sasl_security_options = noanonymous, noplaintext
smtpd_sasl_tls_security_options = noanonymous
```

上記の例では、暗号化されていないセッションでは匿名認証は許可されず、暗号化されていないユーザー名またはパスワードを送信するメカニズムがないことを示しています。暗号化セッション（ **TLS**を使用）では、非匿名認証メカニズムのみが許可されます。

許可 される **SASL** メカニズムを制限するためのサポートされるすべてのポリシーの一覧は、http://www.postfix.org/SASL_README.html#smtpd_sasl_security_options を参照してください。

**その他のリソース**

以下のオンラインリソースは、**SASL** で **Postfix** SMTP 認証を設定するのに役立つ追加情報を提供します。

- http://wiki2.dovecot.org/HowTo/PostfixAndDovecotSASL: SMTP 認証に **Dovecot SASL** 実装を使用するように **Postfix** を設定する方法に関する情報が含まれています。
- http://www.postfix.org/SASL_README.html#server_sasl: SMTP 認証に **Dovecot** または **Cyrus SASL** 実装のいずれかを使用するように **Postfix** を設定する方法に関する情報が含まれています。

### 2.2.8. Sendmail のセキュリティー保護

Sendmail は、SMTP(Simple Mail Transfer Protocol)を使用して他の MTA と電子メールクライアントまたは配信エージェントとの間で電子メッセージを送信するメール転送エージェント(MTA)です。多くの

MTA は相互にトラフィックを暗号化できますが、ほとんどはないため、パブリックネットワークを介して電子メールを送信することは本質的に安全でない通信とみなされます。

Sendmail サーバーの実装を計画している場合には、以下の問題に対応することが推奨されます。

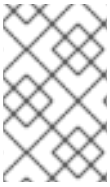
#### 2.2.8.1. サービス拒否攻撃の制限

電子メールの性質上、決定された攻撃者はメールでサーバーをあふれることができ、サービス拒否を引き起こす可能性があります。の以下のディレクティブに制限を設定すると **/etc/mail/sendmail.mc**、このような攻撃の影響度は制限されます。

- **confCONNECTION_RATE_THROTTLE**: サーバーが 1 秒あたりに受信できる接続の数。デフォルトでは、Sendmail は接続数を制限しません。制限が設定され、到達すると、接続が遅延します。
- **confMAX_DAEMON_CHILDREN**: サーバーが生成できる子プロセスの最大数。デフォルトでは、Sendmail は子プロセスの数に制限を割り当てません。制限が設定され、到達すると、接続が遅延します。
- **confMIN_FREE_BLOCKS**: サーバーがメールを受け入れるために利用できる空きブロックの最小数。デフォルトは 100 ブロックです。
- **confMAX_HEADERS_LENGTH**: メッセージヘッダーの許容可能な最大サイズ（バイト単位）。
- **confMAX_MESSAGE_SIZE**: 1 つのメッセージの許容可能な最大サイズ（バイト単位）。

#### 2.2.8.2. NFS および Sendmail

NFS 共有ボリュームにメールスプールディレクトリーを **/var/spool/mail/**配置することはありません。NFSv2 および NFSv3 はユーザーおよびグループ ID に対する制御を維持しないため、複数のユーザーが同じ UID を使用し、相互のメールを受信して読み込むことができます。



**注記**

**SECRPC_GSS** カーネルモジュールは UID ベースの認証を使用しないため、Kerberos を使用する NFSv4 では、これは当てはまりません。ただし、引き続き、NFS 共有ボリュームにメールスプールディレクトリーを配置し **ない** ことが推奨されます。

#### 2.2.8.3. メールのみのユーザー

Sendmail サーバーでローカルユーザーが悪用しないようにするには、メールユーザーが電子メールプログラムを使用して Sendmail サーバーのみにアクセスすることが推奨されます。メールサーバーのシェルアカウントは許可されず、**/etc/passwd** ファイル内のすべてのユーザーシェルは（root ユーザー **/sbin/nologin** を除く）に設定する必要があります。

#### 2.2.8.4. Sendmail ネットワークリスティングの無効化

Sendmail は、デフォルトでは、ローカルのループバックアドレスのみをリッスンするように設定されています。これは、ファイル **/etc/mail/sendmail.mc** を表示して、以下の行が表示されることを確認します。

```
DAEMON_OPTIONS(`Port=smtp,Addr=127.0.0.1, Name=MTA')dnl
```

これにより、Sendmail はネットワークからではなく、ローカルシステムからのメールメッセージ（cron ジョブレポートなど）のみを受け入れるようになります。これはデフォルト設定であり、ネットワーク攻撃から Sendmail を保護します。

ローカルホストの制限を削除するには、Addr=127.0.0.1 文字列を削除する必要があります。Sendmail の設定を変更するには、sendmail-cf パッケージをインストールして **.mc** ファイルを編集し、**sendmail** を実行し、再起動 **/etc/mail/make** する必要があります。**.cf** 設定ファイルが再生成されます。システムクロックが正しく機能している必要があり、設定ファイルを自動的に再生成するには、このアクション間でシステムクロック時間が経過しないことに注意してください。

### 2.2.9. ポートが一覧表示されるかどうかの確認

システムの攻撃対象領域が大きくなるため、不要なオープンポートを回避する必要があります。システムがサービスを提供していると、予期せぬオープンポートがリスニング状態にあることが検出された場合は、侵入の署名である可能性があるため、調査する必要があります。

コンソールから以下のコマンドを発行し、ネットワークからの接続をリッスンしているポートを決定します。

```
~]# netstat -tanp | grep LISTEN
tcp        0      0 0.0.0.0:45876               0.0.0.0:*                   LISTEN      1193/rpc.statd
tcp        0      0 192.168.122.1:53            0.0.0.0:*                   LISTEN      1241/dnsmasq
tcp        0      0 127.0.0.1:631               0.0.0.0:*                   LISTEN      1783/cupsd
tcp        0      0 127.0.0.1:25                0.0.0.0:*                   LISTEN      7696/sendmail
tcp        0      0 0.0.0.0:111                 0.0.0.0:*                   LISTEN      1167/rpcbind
tcp        0      0 127.0.0.1:30003             0.0.0.0:*                   LISTEN      1118/tcsd
tcp        0      0 :::631                      :::*                        LISTEN      1/init
tcp        0      0 :::35018                    :::*                        LISTEN      1193/rpc.statd
tcp        0      0 :::111                      :::*                        LISTEN      1167/rpcbind
```

システムに必要なサービスでコマンドの出力を確認し、特に不要または承認されていないサービスをオフにし、チェックを繰り返します。次に、ネットワークに接続された別のシステムから最初のシステムに接続された別のシステムから **nmap** を使用して外部チェックを行います。これは、**iptables** のルールの検証に使用できます。外部システムから、**netstat** 出力に示される IP アドレス（localhost 127.0.0.0 または ::1 範囲を除く）のすべての IP アドレスをスキャンします。IPv6 アドレスのスキャンには、**-6** オプションを使用します。詳細は **man nmap(1)** を参照してください。

以下は、別のシステムのコンソールから発行されるコマンドの例で、ネットワークからの TCP 接続をリッスンしているポートを特定します。

```
~]# nmap -sT -O 192.168.122.1
```

以下を参照してください。 netstat(8),nmap(1)、および services(5) 詳細は man ページです。

### 2.2.10. ソースルーティングの無効化

ソースルーティングは、IP パケットが情報を伝送できるようにするインターネットプロトコルメカニズムです。アドレスの一覧は、ルーターにパケットが取得する必要のあるパスを指示します。ルートがトラバースされると、ホップを記録するオプションもあります。取得したホップの一覧「ルートレコード」は、宛先にソースへの戻りパスを提供します。これにより、ソース（送信ホスト）はルートを緩やか、または厳密に指定でき、一部またはすべてのルーターのルーティングテーブルを無視できます。これにより、ユーザーは悪意のある目的でネットワークトラフィックをリダイレクトできます。そのため、ソースベースのルーティングを無効にする必要があります。

**accept_source_route** オプションを指定すると、ネットワークインターフェースが *Strict Source*

*Route*（）のあるパケットを受信します。SSR）、または *Roose ソースルーティング*（LSR）オプションを設定します。ソースルーティングパケットの受け入れは **sysctl** 設定によって制御されます。root で以下のコマンドを発行し、SSR オプションまたは LSR オプションが設定されたパケットを破棄します。

```
~]# /sbin/sysctl -w net.ipv4.conf.all.accept_source_route=0
```

パケットの転送の無効化は、可能な場合は上記と併せて行う必要があります（転送を無効にすると、仮想化に干渉する可能性があります）。root で以下に一覧表示されているコマンドを実行します。

このコマンドにより、全インターフェースでの IPv4 パケットおよび IPv6 パケットの転送が無効になります。

```
~]# /sbin/sysctl -w net.ipv4.conf.all.forwarding=0
```

```
~]# /sbin/sysctl -w net.ipv6.conf.all.forwarding=0
```

これらのコマンドは、すべてのインターフェースにおけるマルチキャストパケットの転送を無効にします。

```
~]# /sbin/sysctl -w net.ipv4.conf.all.mc_forwarding=0
```

```
~]# /sbin/sysctl -w net.ipv6.conf.all.mc_forwarding=0
```

ICMP リダイレクトを受け入れることは、正当な使用はほとんどありません。特に必要でない限り、ICMP リダイレクトされたパケットの受け入れと送信を無効にします。

以下のコマンドは、すべてのインターフェースで、すべての ICMP リダイレクトされたパケットの受け入れを無効にします。

```
~]# /sbin/sysctl -w net.ipv4.conf.all.accept_redirects=0
```

```
~]# /sbin/sysctl -w net.ipv6.conf.all.accept_redirects=0
```

このコマンドは、すべてのインターフェース上で、安全な ICMP リダイレクトされたパケットの受け入れを無効にします。

```
~]# /sbin/sysctl -w net.ipv4.conf.all.secure_redirects=0
```

このコマンドは、全インターフェースでの IPv4 ICMP リダイレクトされたパケットの送信を無効にします。

```
~]# /sbin/sysctl -w net.ipv4.conf.all.send_redirects=0
```

**重要**

net.ipv4.conf.all.send_redirects または net.ipv4.conf.*interface*.send_redirects オプションのいずれかが有効な場合でも、ICMP リダイレクトの送信は、アクティブのままになります。net.ipv4.conf.*interface.send_redirects オプションを、すべてのインターフェース* の **0** 値に設定するようにしてください 。新しいインターフェースを追加するたびに ICMP リクエストの送信を自動的に無効にするには、次のコマンドを実行します。

```
~]# /sbin/sysctl -w net.ipv4.conf.default.send_redirects=0
```

IPv4 リダイレクトされたパケットの送信を無効にするディレクティブのみがあります。の説明は、『RFC4294 を』 参照してください。 「IPv6 ノード要件」）。これにより、IPv4 と IPv6 間でこの差異が生じていました。

設定を永続化するには、に追加する必要があり **/etc/sysctl.conf**ます。

以下を参照してください。 sysctl(8) 詳細は man ページです。ソースベースのルーティングおよびそのバリアントに関連するインターネットオプションの説明は、『RFC791』 を参照してください。

**警告**

イーサネットネットワークは、ARP、MAC アドレススプーフィング、未承認の DHCP サーバー、IPv6 ルーター、周りの広告など、トラフィックをリダイレクトする方法を追加で提供します。さらに、ユニキャストトラフィックがブロードキャストされる可能性があるため、情報漏えいが生じます。これらの問題は、ネットワークオペレーターが実装した特定のカウンターのみで対応できます。ホストベースのカウンターは完全に有効ではありません。

### 2.2.11. 逆方向パス転送

逆方向パス転送は、1 つのインターフェースを介して別のインターフェースを介して送信しないようにするために使用します。発信ルートと受信ルートが異なる場合、*非対称ルーティング*と呼ばれることもあります。ルーターは、頻繁にパケットをルーティングしますが、ほとんどのホストはこれを行う必要はありません。例外として、あるリンクでトラフィックを送信し、別のサービスプロバイダーから別のリンクでトラフィックを受信する このようなアプリケーションのことです。たとえば、リースした行をと併用すると、 xDSL または satellite のリンク先 3G イタリア語。このようなシナリオが該当する場合は、受信インターフェースで逆方向パス転送を無効にする必要があります。短い場合には、必要でない限り、ローカルサブネットからの IP アドレスの偽装を防ぐため、有効にすることが推奨されます。これにより、ローカルサブネットからの **IP** アドレスが偽装され、可能性が低減するためです。 DDoS 攻撃。

**注記**

Red Hat Enterprise Linux 6（Red Hat Enterprise Linux 5 とは異なり）はデフォルトで *Strict Reverse Path Forwarding* を使用します。Red Hat Enterprise Linux 6 は、RFC 3704, Ingress Filtering for Multihomed Networks の厳密な逆方向パスの推奨事項に従います。現在、これは Red Hat Enterprise Linux 6 の **IPv4** にのみ適用されます。

**警告**

転送が有効になっている場合は、source-address 検証の他の手段がある場合に限り、逆方向パス転送を無効にする必要があります（たとえば **iptables** ルールなど）。

**rp_filter**

逆方向パス転送は、**rp_filter** ディレクティブで有効にします。**rp_filter** オプションは、3 つのモードのいずれかから選択するようにカーネルに指示するために使用されます。

デフォルトの動作を設定する際には、以下の形式を取ります。

```
~]# /sbin/sysctl -w net.ipv4.conf.default.rp_filter=INTEGER
```

ここで、*INTEGER* は以下のいずれかになります。

- **0**: ソースの検証はありません。
- **1**: RFC 3704 で定義される厳密モード
- **2**: RFC 3704 で定義される緩やかなモード。

この設定は、を使用してネットワークインターフェースごとに上書きでき **net.ipv4.*interface*.rp_filter**ます。これらの設定を再起動後も維持するには、**/etc/sysctl.conf** ファイルを変更します。

#### 2.2.11.1. その他のリソース

以下は、逆方向パス転送の詳細を確認するリソースです。

- **インストールされているドキュメント**

  **usr/share/doc/kernel-doc-*version*/Documentation/networking/ip-sysctl.txt**: このファイルには、**/proc/sys/net/ipv4/** ディレクトリーで利用可能なファイルおよびオプションの完全な一覧が記載されています。

- **便利な Web サイト**

  https://access.redhat.com/knowledge/solutions/53031 - Red Hat ナレッジベースアーティクル **rp_filter**

  マルチホームネットワーク向けの Ingress Filtering の説明は、RFC 3704 を参照してください。

## 2.3. シングルサインオン(SSO)

Red Hat Enterprise Linux の SSO 機能は、Red Hat Enterprise Linux デスクトップユーザーがパスワードを入力する必要がある回数を削減します。複数の主要アプリケーションは、同じ基盤となる認証および承認メカニズムを利用しているため、ユーザーがログイン画面から Red Hat Enterprise Linux にログインし、パスワードを再入力する必要はありません。これらのアプリケーションは以下で説明します。

プラグ可能な認証モジュールの詳細は、『Red Hat Enterprise Linux 6 Managing Single Sign-On and Smart Cards』を参照してください。

## 2.4. プラグ可能な認証モジュール(PAM)

プラグ可能な認証モジュールは、認証およびセキュリティーの共通のフレームワークです。Red Hat Enterprise Linux のシングルサインオンメソッド（Kerberos およびスマートカード）は、基礎となる PAM 設定に依存します。

プラグ可能な認証モジュールの詳細は、Red Hat Enterprise Linux 6 『Managing Single Sign-On and Smart Cards』の該当する章を参照してください。

## 2.5. KERBEROS

ネットワーク内でシステムのセキュリティーと整合性を維持することは重要です。また、ネットワークインフラストラクチャー内のすべてのユーザー、アプリケーション、サービス、およびサーバーが含まれます。これには、ネットワーク上で実行中のすべての内容と、これらのサービスが使用される仕組みを理解する必要があります。このセキュリティーの維持の中核となるのは、これらのアプリケーションおよびサービスへのアクセスを維持し、そのアクセスを強制することです。

Kerberos は、ユーザーとマシンの両方がネットワークに対して自らを識別し、管理者が設定した領域およびサービスへの定義済みかつ制限されたアクセスを受け取れるようにするメカニズムを提供します。Kerberos はアイデンティティーを確認してエンティティーを認証します。また、Kerberos はこの認証データも保護し、外部からアクセス、使用、改ざんされないようにします。

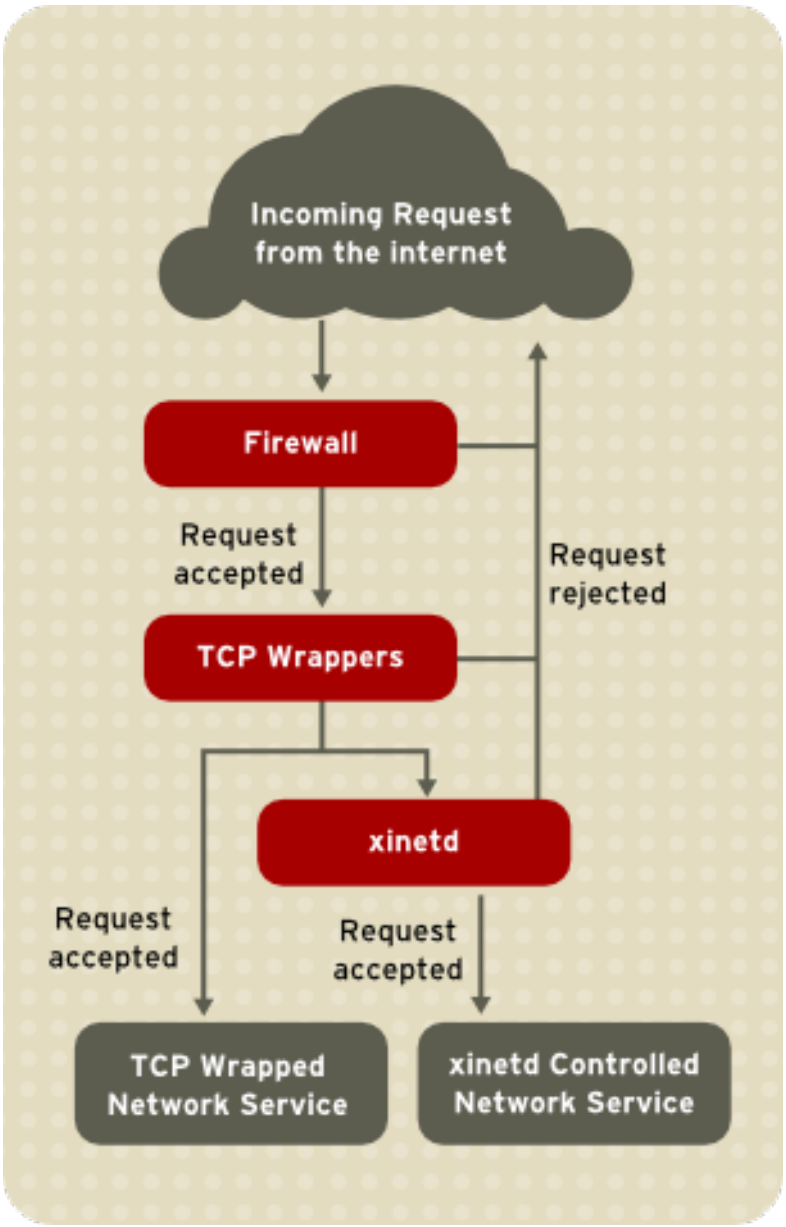
プラグ可能な認証モジュールの詳細は、Red Hat Enterprise Linux 6 『Managing Single Sign-On and Smart Cards』の該当する章を参照してください。

## 2.6. TCP WRAPPER および XINETD

ネットワークサービスへのアクセスを制御することは、サーバー管理者向けの最も重要なセキュリティータスクの 1 つです。Red Hat Enterprise Linux は、この目的でいくつかのツールを提供しています。たとえば、**iptables**ベースのファイアウォールは、カーネルのネットワークスタック内の未対応ネットワークパケットをフィルタリングします。ネットワークサービスを使用するネットワークサービスでは、*TCP Wrapper* は、どのホストが「**ラップ**された」ネットワークサービスに接続できないホストを定義して保護の層を追加します。このようなラップされたネットワークサービスの 1 つが **xinetd** の**スーパーサーバー** です。このサービスは、ネットワークサービスのサブセットへの接続を制御し、さらにアクセス制御を改良するため、スーパーサーバーと呼ばれます。

図2.4「ネットワークサービスへのアクセス制御」 は、これらのツールがどのように連携してネットワークサービスを保護する方法についての基本的な図です。

図2.4 ネットワークサービスへのアクセス制御




[D]

でファイアウォールを使用する方法は **iptables**、を参照してください 「iptables」 。

### 2.6.1. TCP Wrapper

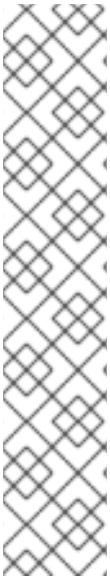
TCP Wrapper パッケージ（tcp_wrappers および tcp_wrappers-libs）はデフォルトでインストールされ、ネットワークサービスにホストベースのアクセス制御を提供します。パッケージ内で最も重要なコンポーネントは、**/lib/libwrap.so** または **/lib64/libwrap.so** ライブラリーです。通常、TCP-wrapped サービスは、**libwrap.so** ライブラリーに対してコンパイルされたサービスです。

TCP-wrapped サービスへの接続を試みる**/etc/hosts.allow** と、サービスは最初にホストのアクセスファイルを参照 **/etc/hosts.deny**し、クライアントが接続できるかどうかを判断します。ほとんどの場合、syslog デーモン(**syslogd**)を使用して要求するクライアントと要求されたサービスの名前を **/var/log/secure** またはに書き込み **/var/log/messages**ます。

クライアントが接続できる場合は、要求されたサービスへの接続の TCP Wrappers リリース制御がクライアントとサーバー間の通信には含まれません。

TCP Wrapper はアクセス制御とログに加えて、要求されたネットワークサービスへの接続制御を拒否または解放する前に、クライアントと対話するコマンドを実行します。

TCP Wrapper は、サーバー管理者のセキュリティーツールに有用な追加機能であるため、Red Hat Enterprise Linux 内のほとんどのネットワークサービスは **libwrap.so** ライブラリーにリンクされます。このようなアプリケーションには、**/usr/sbin/sshd** **/usr/sbin/sendmail**、および **/usr/sbin/xinetd** が含まれます。



**注記**

ネットワークサービスバイナリーがリンクされているかどうかを確認するには **libwrap.so**、root ユーザーとして以下のコマンドを実行します。

```
ldd <binary-name> | grep libwrap
```

*<binary-name>* をネットワークサービスバイナリーの名前に置き換えます。コマンドが出力なしでプロンプトに直接返された場合、ネットワークサービスはにリンクされ **ません libwrap.so**。

以下の例は、**/usr/sbin/sshd** がリンクされていることを **libwrap.so**示しています。

```
~]# ldd /usr/sbin/sshd | grep libwrap
        libwrap.so.0 => /lib/libwrap.so.0 (0x00655000)
```

#### 2.6.1.1. TCP Wrapper の利点

TCP Wrapper は、その他のネットワークサービス制御技術と比較して、以下の利点を提供します。

- **クライアントとラップされたネットワークサービスへの解析 - 接続クライアントとラップされたネットワークサービスの両方** が、TCP Wrapper が使用されていることを認識しません。禁止されているクライアントからの接続に失敗し、正当なユーザーはログを記録し、要求されたサービスに接続します。

- **複数のプロトコルの集中管理** (TCP Wrapper)は、保護するネットワークサービスとは別に動作するため、多くのサーバーアプリケーションは共通のアクセス制御設定ファイルセットを共有できるため、管理が簡単になります。

### 2.6.2. TCP Wrapper 設定ファイル

クライアントがサービスへの接続を許可されているかどうかを確認するには、TCP Wrapper は、ホストアクセスファイルと呼ばれる以下の 2 つの *ファイル* を参照します。

- **/etc/hosts.allow**

- **/etc/hosts.deny**

TCP でラップされたサービスがクライアント要求を受信すると、以下の手順が実行されます。

1. **参照 /etc/hosts.allow**- TCP-wrapped サービスは **/etc/hosts.allow** ファイルを順番に解析し、そのサービスに指定された最初のルールを適用します。一致するルールを見つけると、接続を許可します。そうでない場合は、次の手順に移動します。

2. **参照 /etc/hosts.deny**- TCP-wrapped サービスは **/etc/hosts.deny** ファイルを順次解析します。一致するルールが見つかると、接続を拒否します。そうでない場合には、サービスへのアクセスを付与します。

ネットワークサービスを保護するために TCP Wrappers を使用する際に考慮すべき重要な点を以下に示します。

- のアクセスルールは最初に適用 **hosts.allow** されるため、はで指定されているルールよりも優先され **hosts.deny**ます。そのため、サービスへのアクセスが許可されると **hosts.allow**、同じサービスへのアクセスを拒否するルール **hosts.deny** は無視されます。
- 各ファイル内のルールは上部から読み取られ、指定のサービスの最初のマッチングルールは適用されます。ルールの順序が非常に重要です。
- ファイルのルールが見つからない場合や、ファイルが存在しない場合は、サービスへのアクセスが許可されます。
- TCP でラップされたサービスは、ホストアクセスファイルからルールをキャッシュしないため、ネットワークサービスを再起動せずに、に対する変更は即座に **hosts.allow hosts.deny** 反映されます。

**警告**

ホストアクセスファイルの最後の行が（ **Enter** キーを押して作成した）改行文字でない場合、ファイルの最後のルールは失敗し、エラーが **/var/log/messages** またはに記録され **/var/log/secure**ます。これは、バックスラッシュ文字を使用せずに複数の行にまたがるルールでもあります。以下の例は、以下のいずれかの状況によりルールが失敗した場合のログメッセージの関連する部分を示しています。

```
warning: /etc/hosts.allow, line 20: missing newline or line too long
```

#### 2.6.2.1. アクセスルールのフォーマット

**/etc/hosts.allow** との両方の形式 **/etc/hosts.deny** は同一です。各ルールは、独自の行上にある必要があります。ハッシュ(#)で始まる空の行や行は無視されます。

各ルールでは、以下の基本形式を使用してネットワークサービスへのアクセスを制御します。

```
<daemon list> : <client list>[: <option> : <option> : ...]
```

- *<daemon list>*: プロセス名のコンマ区切りリスト（サービス名**以外** ）または **ALL** ワイルドカード。デーモンリストは、オペレーター（を参照 「Operator」）を受け入れて柔軟性を向上させます。
- *<client list>*: ルールの影響を受けるホストを識別するホスト名、ホスト IP アドレス、特殊パターン、またはワイルドカードのコンマ区切りリスト。クライアントリストは、にリストされているオペレーターも受け入れ、柔軟性が向上 「Operator」 します。
- *<option>*: ルールがトリガーされる際に実行されるアクションのオプションまたはコロン区切りのアクションリストです。オプションフィールドは、拡張、シェルコマンドの起動、アクセスの許可または拒否、ロギングの動作の変更をサポートします。

**注記**

上記の用語の一部の詳細については、本ガイドの他の箇所で参照してください。

- 「ワイルドカード」
- 「パターン」
- 「expansions」
- 「オプションフィールド」

以下は、ホストアクセスルールの基本例です。

```
vsftpd : .example.com
```

このルールは 、**example.com** ドメインのホストから FTP デーモン(**vsftpd**)への接続を監視するように TCP Wrappers に指示します。このルールがに表示されると **hosts.allow**、接続が許可されます。このルールがに表示されると **hosts.deny**、接続は拒否されます。

次のホストアクセスルールの例はより複雑で、以下の 2 つのオプションフィールドを使用します。

```
sshd : .example.com  \
 : spawn /bin/echo `/bin/date` access denied>>/var/log/sshd.log \
 : deny
```

各オプションフィールドの前にバックスラッシュ(\)が設定されていることに注意してください。バックスラッシュを使用すると、長さが原因でルールの失敗を防ぐことができます。

このサンプルルールは、SSH デーモン(**sshd**)への接続が **example.com** ドメインのホストから試行された場合、**echo** コマンドを実行して特別なログファイルに追加し、接続を拒否していることを示しています。オプションの **deny** ディレクティブが使用されるため、この行は **hosts.allow** ファイルに記載されている場合でもアクセスを拒否します。利用可能なオプション 「オプションフィールド」 の詳細は、を参照してください。

#### 2.6.2.1.1. ワイルドカード

ワイルドカードを使用すると、TCP Wrapper がデーモンまたはホストのグループと簡単に一致できるようになります。これらは、アクセスルールのクライアントリストフィールドで最も頻繁に使用されます。

以下のワイルドカードを使用できます。

- **ALL** : すべてと一致します。デーモンリストとクライアント一覧の両方に使用できます。
- **LOCAL** : localhost など、ピリオド(.)が含まれないホストと一致します。
- **KNOWN** : ホスト名およびホストアドレスが分かっているホスト、またはユーザーが分かっている場所を照合します。
- **UNKNOWN** : ホスト名またはホストアドレスが不明なホスト、またはユーザーが不明な場所を照合します。
- **PARANOID** : ホスト名を取得するために、ソース IP アドレスで逆引き DNS ルックアップが実行されます。次に、IP アドレスを解決するために DNS ルックアップが実行されます。2 つの IP アドレスが接続に一致しない場合、ログは更新されます。

**重要**

、**KNOWN UNKNOWN**、および **PARANOID** ワイルドカードは、正しく操作するために機能する DNS サーバーに依存するため、注意して使用する必要があります。名前解決の中断により、正当なユーザーがサービスにアクセスできなくなる可能性があります。

#### 2.6.2.1.2. パターン

パターンは、アクセスルールのクライアントフィールドで使用することで、クライアントホストのグループをより正確に指定できます。

以下は、クライアントフィールドのエントリーの一般的なパターンの一覧です。

- ホスト**名のピリオド(.)- ホスト名の開始時にピリオドを配置すると**、その名前のコンポーネントを共有するすべてのホストと一致します。以下の例は、**example.com** ドメイン内のホストに適用されます。

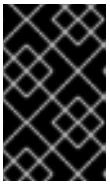
```
ALL : .example.com
```

- **IP アドレスの末尾にピリオド(.)- IP アドレスの末尾のピリオドを配置して、IP アドレスの最初の数値グループを共有するすべてのホストに一致させ** ます。以下の例は、**192.168.x.x** ネットワーク内のホストに適用されます。

```
ALL : 192.168.
```

- **IP アドレス/ネットマスクのペア** - 特定の IP アドレスグループへのアクセスを制御するために、ネットマスク式もパターンとして使用できます。以下の例は、アドレス範囲が **192.168.0.0** から **192.168.1.255** までのホストに適用されます。

```
ALL : 192.168.0.0/255.255.254.0
```



**重要**

IPv4 アドレス空間で作業を行う場合は、アドレス/接頭辞の長さ(prefixlen)のペア宣言（）CIDR 表記はサポートされていません。IPv6 ルールだけがこの形式を使用できます。

- **[ipv6 address]/prefixlen pair** - [net]/prefixlen ペアは、IPv6 アドレスの特定グループへのアクセスを制御するパターンとしても使用できます。以下の例では、3ffe:505: **2:1:: 3ffe:505:ffff:ffff:ffff:ffff:ffff:ffff:ffff:ffff:ffff:ffff:ffff:ffff:ffff:ffff:ffff:ffff:ffff:ffff:** ffff:ffff までのアドレス範囲が **3ffe:505:2:1:ffff:ffff:ffff:ffff:ffff:ffff:ffff:ffff:ffff:ffff:ffff:ffff:ffff:ffff:ffff:ffff**

```
ALL : [3ffe:505:2:1::]/64
```

- **アスタリスク(*)- アスタリスクは**、他のタイプのパターンが含まれるクライアント一覧で混在しない限り、ホスト名または IP アドレスのグループ全体を一致させるために使用できます。以下の例では、**example.com** ドメイン内のホストに適用されます。

```
ALL : *.example.com
```

- **スラッシュ(/)- クライアント一覧がスラッシュで始まる場合は**、ファイル名として処理されます。これは、多数のホストを指定するルールが必要な場合に役立ちます。以下の例では、すべての Telnet 接続について TCP Wrappers を **/etc/telnet.hosts** ファイルを参照します。

```
in.telnetd : /etc/telnet.hosts
```

また、使用されていないパターンも TCP Wrapper によって受け入れられます。詳細は **hosts_access** man 5 ページを参照してください。

**警告**

ホスト名およびドメイン名を使用する場合は、十分に注意してください。攻撃者は、さまざまなトリックを使用して正確な名前解決を回避できます。さらに、DNS サービスの中断により、許可されたユーザーがネットワークサービスを使用できなくなります。したがって、可能な限り IP アドレスを使用するのが最も適しています。

#### 2.6.2.1.3. Portmap および TCP Wrapper

**Portmap**の TCP Wrapper の実装はホストのルックアップをサポートしません。つまり、ホスト名を使用してホストを特定 **portmap** できません。したがって、**hosts.allow** またはのポートマップのアクセス制御ルールは、ホストを指定する **ALL**ために IP アドレスまたはキーワードを使用する **hosts.deny** 必要があります。

**portmap** アクセス制御ルールの変更はすぐには反映されない場合があります。**portmap** サービスを再起動する必要がある場合があります。

NIS や NFS などの広く使用されているサービスは、操作 **portmap** に依存するため、これらの制限に注意してください。

#### 2.6.2.1.4. Operator

現時点で、アクセス制御ルールは、1 つのオペレーター（）を受け入れ **EXCEPT**ます。デーモンの一覧とルールのクライアントリストの両方で使用できます。

**EXCEPT** Operator は、同一ルール内で特定の例外のマッチをさらに増やすことができます。

以下の例では **hosts.allow**、**example.com** ホストはすべて attacker. **example.com** 以外のすべてのサービスに接続できます。

```
ALL : .example.com EXCEPT attacker.example.com
```

**hosts.allow** ファイルからの別の例では、**192.168.0.*x*** ネットワークからのクライアントは FTP 以外のサービスを使用できます。

```
ALL EXCEPT vsftpd : 192.168.0.
```

**注記**

組織的には、Operator の使用を避けることが容易に **EXCEPT** なります。これにより、他の管理者は適切なファイルをすばやくスキャンして、**EXCEPT** Operator をソートしなくても、どのホストがサービスへのアクセスを許可または拒否されているかを確認することができます。

#### 2.6.2.2. オプションフィールド

アクセスを許可および拒否する基本的なルールに加え、TCP Wrapper の Red Hat Enterprise Linux 実装は、*オプションフィールド* を介したアクセス制御言語への拡張機能をサポートします。ホストアクセスルールで option フィールドを使用すると、管理者はログの動作の変更、アクセス制御の分離、シェルコマンドの起動などのさまざまなタスクを実行できます。

##### 2.6.2.2.1. ロギング

オプションフィールドを使用すると、管理者は **severity** ディレクティブを使用してルールのログファシリティーおよび優先度を簡単に変更できます。

以下の例では、**example.com** ドメインのホストから SSH デーモンへの接続が、優先度がのデフォルトの **authpriv syslog** ファシリティーに記録され **emerg**ます。

```
sshd : .example.com : severity emerg
```

**severity** オプションを使用してファシリティーを指定することもできます。以下の例では、**example.com** ドメインから優先順位の **local0** ファシリティーに、ホストによる SSH 接続の試行をすべてログに記録し **alert**ます。

```
sshd : .example.com : severity local0.alert
```

**注記**

実際には、この例では、syslog デーモン(**syslogd**)がファシリティーにログを記録するように設定されるまでは **local0** 機能しません。カスタムログ機能の設定に関する詳細は、の **syslog.conf** man ページを参照してください。

##### 2.6.2.2.2. アクセス制御

オプションフィールドを使用すると、管理者はまたは **deny** ディレクティブを最終オプションとして追加することで、1つのルールでホストを明示的に許可 **allow** または拒否できます。

たとえば、以下の 2 つのルールでは **client-1.example.com** からの SSH 接続を許可しますが、 **client-2.example.com** からの接続を拒否します。

```
sshd : client-1.example.com : allow
sshd : client-2.example.com : deny
```

ルールごとにアクセス制御を許可することで、オプションフィールドを使用すると、管理者はすべてのアクセス制御を 1つのファイル（ **hosts.allow** または）に統合できます **hosts.deny**。管理者は、アクセスルールを簡単に整理する方法を考慮しています。

##### 2.6.2.2.3. シェルコマンド

オプションフィールドは、以下の 2 つのディレクティブを使用してシェルコマンドを起動できるアクセスルールです。

- **spawn**: 子プロセスとしてシェルコマンドを起動します。このディレクティブは、を使用して、要求するクライアントの詳細情報を **/usr/sbin/safe_finger** 取得したり、**echo** コマンドを使用して特別なログファイルを作成したりできます。

  以下の例では、**example.com** ドメインから Telnet サービスにアクセスしようとすると、特別なファイルに記録されます。

  ```
  in.telnetd : .example.com \
   : spawn /bin/echo `/bin/date` from %h>>/var/log/telnet.log \
   : allow
  ```

- **twist**: 要求されたサービスを指定されたコマンドに置き換えます。このディレクティブは、多くの場合、侵入者向けのトラップを設定するのに使用されます（「ホイットポット」とも呼ばれます）。クライアントを接続するメッセージを送信するのに使用することもできます。**twist** ディレクティブは、ルール行の最後で行う必要があります。

  以下の例では、**example.com** ドメインから FTP サービスにアクセスしようとしているクライアントは、**echo** コマンドを使用してメッセージを送信します。

  ```
  vsftpd : .example.com \
   : twist /bin/echo "421 This domain has been black-listed. Access denied!"
  ```

shell コマンドのオプションの詳細は、の **hosts_options** man ページを参照してください。

#### 2.6.2.2.4. expansions

拡張（ **spawn** および **twist** ディレクティブとともに使用する場合）は、関連するクライアント、サーバー、およびプロセスに関する情報を提供します。

以下は、サポートされる拡張の一覧です。

- **%a**: クライアントの IP アドレスを返します。
- **%A**: サーバーの IP アドレスを返します。
- **%c**: ユーザー名やホスト名、ユーザー名および IP アドレスなどのさまざまなクライアント情報を返します。
- **%d**: デーモンプロセス名を返します。
- **%h**: クライアントのホスト名（またはホスト名が利用できない場合は IP アドレス）を返します。
- **%H**: サーバーのホスト名（またはホスト名が利用できない場合は IP アドレス）を返します。
- **%n**: クライアントのホスト名を返します。使用できない場合 **unknown** は、が表示されます。クライアントのホスト名とホストアドレスが一致しない場合 **paranoid** は、が表示されます。
- **%N**: サーバーのホスト名を返します。使用できない場合 **unknown** は、が表示されます。サーバーのホスト名とホストアドレスが一致しない場合 **paranoid** は、が表示されます。
- **%p**: デーモンのプロセス ID を返します。

- **%s** - デーモンプロセスやサーバーのホストまたは IP アドレスなど、さまざまな種類のサーバー情報を返します。
- **%u**: クライアントのユーザー名を返します。使用できない場合 **unknown** は、が表示されます。

以下のサンプルルールは、**spawn** コマンドとともに拡張を使用して、カスタマイズされたログファイルのクライアントホストを特定します。

SSH デーモン(**sshd**)への接続が **example.com** ドメインのホストから試行される場合は、（拡張を使用して）クライアントのホスト名を含む、（ **%h** 拡張による）などの試行を、特別なファイルに **echo** 記録します。

```
sshd : .example.com  \
 : spawn /bin/echo `/bin/date` access denied to %h>>/var/log/sshd.log \
 : deny
```

同様に、拡張を使用してメッセージをクライアントにカスタマイズすることもできます。以下の例では、**example.com** ドメインから FTP サービスにアクセスしようとすると、サーバーから禁止されたことが通知されます。

```
vsftpd : .example.com \
: twist /bin/echo "421 %h has been banned from this server!"
```

利用可能な拡張の詳細と、追加のアクセス制御オプションは、の man ページのセクション 5 **hosts_access man 5 hosts_access**と、の man ページを参照してください **hosts_options**。

TCP Wrappers 「その他のリソース」 の詳細は、を参照してください。

### 2.6.3. xinetd

**xinetd** デーモンは、FTP、IMAP、Telnet など、一般的なネットワークサービスのサブセットへのアクセスを *制御する TCP でラップされたスーパー* サービスです。また、アクセス制御、強化されたロギング、バインディング、リダイレクト、およびリソース使用制御のためのサービス固有の設定オプションも提供します。

クライアントが **xinetd** が制御するネットワークサービスへの接続を試みると、スーパーサービスはリクエストを受け取り、TCP Wrappers アクセス制御ルールを確認します。

アクセスが許可されると、**xinetd** は、そのサービスの独自のアクセスルールで接続が許可されていることを確認します。また、サービスにより多くのリソースが割り当てられていることを確認し、定義したルールに違反していないことを確認します。

これらの条件がすべて満たされている場合（つまり、サービスへのアクセスが許可され、サービスはリソース制限に達しておらず、サービスが定義されたルールに違反していない場合）、**xinetd** は要求されたサービスのインスタンスを開始し、接続の制御をこれに渡します。接続が確立されると、**xinetd** はクライアントとサーバー間の通信にはこれ以上実行されません。

### 2.6.4. xinetd 設定ファイル

**xinetd** の設定ファイルは以下のとおりです。

- **/etc/xinetd.conf**: グローバルな **xinetd** 設定ファイル
- **/etc/xinetd.d/**: サービス固有のファイルをすべて含むディレクトリー

#### 2.6.4.1. /etc/xinetd.conf ファイル

この **/etc/xinetd.conf** ファイルには、**xinetd**の制御下のすべてのサービスに影響する一般的な設定が含まれます。これは、**xinetd** サービスが最初に開始された際に読み取られるため、設定の変更を有効にするには、**xinetd** サービスを再起動する必要があります。**/etc/xinetd.conf** ファイルの例を以下に示します。

```
defaults
{
  instances             = 60
  log_type              = SYSLOG authpriv
  log_on_success        = HOST PID
  log_on_failure        = HOST
  cps                   = 25 30
}
includedir /etc/xinetd.d
```

これらの行は、**xinetd** の以下の側面を制御します。

- **instances** : **xinetd** が処理できる同時要求の最大数を指定します。
- **log_type** : **authpriv** ログエントリーを **/var/log/secure** ファイルに書き込むログファシリティーを使用するように **xinetd** を設定します。等のディレクティブを追加する **FILE /var/log/xinetdlog** と、**/var/log/** ディレクトリー **xinetdlog** にという名前のカスタムログファイルが作成されます。
- **log_on_success** : 正常な接続試行をログに記録するように **xinetd** を設定します。デフォルトでは、リモートホストの IP アドレスと、要求が記録されるサーバーのプロセス ID です。
- **log_on_failure** : 失敗した接続試行をログに記録するよう **xinetd** を設定するか、または接続が拒否された場合。
- **cps** : 特定のサービスへの 1 秒あたりの 25 を超える接続を許可するように **xinetd** を設定します。この制限を超えると、サービスは 30 秒間廃止されます。
- **includedir /etc/xinetd.d/** : **/etc/xinetd.d/** ディレクトリーにあるサービス固有の設定ファイルに宣言されたオプションが含まれます。詳細は「/etc/xinetd.d/ ディレクトリー」を参照してください。

**注記**

多くの場合、**log_on_success** との **log_on_failure** 設定 **/etc/xinetd.conf** は、サービス固有の設定ファイルでさらに変更されます。したがって、詳細は、ファイルが示すファイルよりも、指定のサービスのログファイルに表示される **/etc/xinetd.conf** 可能性があります。詳細はを 「ロギングのオプション」参照してください。

#### 2.6.4.2. /etc/xinetd.d/ ディレクトリー

**/etc/xinetd.d/** ディレクトリーには **xinetd** が管理する各サービスの設定ファイルが含まれ、ファイル名はサービスと関連します。と同様に **xinetd.conf**、このディレクトリーは **xinetd** サービスが開始する場合にのみ読み取られます。変更を有効にするには、管理者が **xinetd** サービスを再起動する必要があります。

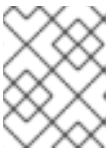
**/etc/xinetd.d/** ディレクトリー内のファイルの形式では、と同じ規則を使用し **/etc/xinetd.conf**ます。各サービスの設定が個別のファイルに保存される主な理由は、カスタマイズを容易にし、他のサービスに影響を及ぼす可能性が低くなります。

これらのファイルがどのように構成されるかを理解するには、ファイルを考慮してください **/etc/xinetd.d/krb5-telnet**。

```
service telnet
{
  flags           = REUSE
  socket_type     = stream
  wait            = no
  user            = root
  server          = /usr/kerberos/sbin/telnetd
  log_on_failure  += USERID
  disable         = yes
}
```

これらの行は、**telnet** サービスのさまざまな側面を制御します。

- **service** : サービス名を指定します（通常は **/etc/services** ファイルに記載されているもののいずれか）。
- **flags** : 接続に任意の属性を設定します。**xinetd** は Telnet 接続のソケットを再利用するように **REUSE** 指示します。

  **注記**

  **REUSE** フラグは非推奨になりました。すべてのサービスが **REUSE** フラグを暗黙的に使用するようになりました。

- **socket_type** : ネットワークソケットタイプをに設定し **stream**ます。
- **wait** : サービスがシングルスレッド(**yes**)またはマルチスレッド(**no**)であるかを指定します。
- **user** : プロセスがで実行されるユーザー ID を指定します。
- **server** : を起動するバイナリーの実行ファイルを指定します。
- **log_on_failure** : ですでに定義されているログパラメーター **log_on_failure** に加えて、ロギングパラメーターを指定し **xinetd.conf**ます。
- **disable** : サービスを無効にする（）か、有効(**yesno**)であるかを指定します。

これらのオプションとその使用方法の詳細は、**xinetd.conf** man ページを参照してください。

### 2.6.4.3. xinetd 設定ファイルの変更

**xinetd** が保護するサービスには、さまざまなディレクティブを使用できます。本セクションでは、一般的に使用されるオプションの一部について説明します。

#### 2.6.4.3.1. ロギングのオプション

**/etc/xinetd.d/** ディレクトリー内のサービス固有の設定ファイルには **/etc/xinetd.conf**、以下のロギングオプションを使用できます。

以下は、一般的に使用されるロギングオプションの一部です。

- **ATTEMPT** : 失敗したことをログに記録します(**log_on_failure**)。

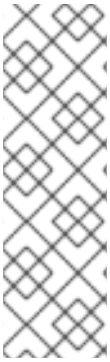
- **DURATION** : サービスがリモートシステム(**log_on_success**)で使用される期間をログに記録します。
- **EXIT** : サービスの終了ステータスまたは終了シグナルをログに記録します(**log_on_success**)。
- **HOST** : リモートホストの IP アドレス（**log_on_failure** および **log_on_success**）をログに記録します。
- **PID** : 要求を受信するサーバーのプロセス ID をログに記録します(**log_on_success**)。
- **USERID** : RFC 1413 で定義されたすべてのマルチスレッドストリームサービス（**log_on_failure** および **log_on_success**）で定義されたメソッドを使用してリモートユーザーをログに記録します。

ロギングオプションの完全なリストは、の **xinetd.conf** man ページを参照してください。

#### 2.6.4.3.2. アクセス制御オプション

**xinetd** サービスのユーザーは、TCP Wrappers ホストアクセスルールの使用、**xinetd** 設定ファイルによるアクセス制御の提供、または両方の組み合わせを選択できます。TCP Wrappers ホストアクセス制御ファイル 「TCP Wrapper 設定ファイル」 の詳細は、を参照してください。

本セクションでは、**xinetd** を使用してサービスへのアクセスを制御する方法を説明します。



**注記**

TCP Wrapper とは異なり、アクセス制御の変更は、**xinetd 管理者が xinetd サービスを再起動する場合にのみ有効になります。**

**また、TCP Wrapper とは異なり、xinetd を使用するアクセス制御は、xinetd が制御するサービスにのみ影響し ます。**

**xinetd ホストのアクセス制御は、TCP Wrappers で使用される方法とは異なります。TCP Wrapper は、すべてのアクセス設定を 2 つのファイル内に配置 /etc/hosts.allow しますが /etc/hosts.deny、xinetdのアクセス制御は /etc/xinetd.d/ ディレクトリーの各サービスの設定ファイルにあります。**

**xinetd では、以下のホストアクセスオプションがサポートされます。**

- **only_from : 指定されたホストのみがサービスを使用できるようにします。**
- **no_access : 一覧表示されたホストがサービスを使用しないようにします。**
- **access_times : 特定のサービスを使用する期間を指定します。時間の範囲は、24 時間表**

記の HH:MM-HH:MM で指定する必要があります。

only_from および no_access オプションは、IP アドレスまたはホスト名の一覧を使用するか、ネットワーク全体を指定できます。TCP Wrapper と同様に、xinetd アクセス制御が強化されたロギング設定と組み合わされると、各接続の試行を詳細に記録する間に、禁止されたホストからのリクエストをブロックし、セキュリティーを強化できます。

たとえば、次の /etc/xinetd.d/telnet ファイルを使用して、特定のネットワークグループからの Telnet アクセスをブロックし、ユーザーがログインできる時間範囲全体を制限することができます。

```
service telnet
{
  disable         = no
  flags           = REUSE
  socket_type     = stream
  wait            = no
  user            = root
  server          = /usr/kerberos/sbin/telnetd
  log_on_failure  += USERID
  no_access       = 172.16.45.0/24
  log_on_success  += PID HOST EXIT
  access_times    = 09:45-16:15
}
```

この例では、172.16.45.0/24 ネットワーク （例： 172.16.45.2）からのクライアントシステムが Telnet サービスにアクセスしようとすると 、 次のメッセージが表示されます。

```
Connection closed by foreign host.
```

さらに、ログイン試行は以下の /var/log/messages ようにログインします。

```
Sep  7 14:58:33 localhost xinetd[5285]: FAIL: telnet address from=172.16.45.107
Sep  7 14:58:33 localhost xinetd[5283]: START: telnet pid=5285 from=172.16.45.107
Sep  7 14:58:33 localhost xinetd[5283]: EXIT: telnet status=0 pid=5285 duration=0(sec)
```

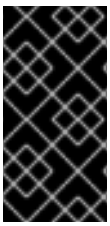
xinetd アクセス制御とともに TCP Wrapper を使用する場合は、2 つのアクセス制御メカニズム間の関係を理解することが重要です。

以下は、クライアントが接続を要求する際に xinetd が続くイベントシーケンスです。

1. xinetd デーモンは、libwrap.so ライブラリー呼び出しを使用して TCP Wrappers ホスト

アクセスルールにアクセスします。deny ルールがクライアントと一致する場合、接続は破棄されます。allow ルールがクライアントと一致する場合、接続は xinetd に渡されます。

2. xinetd デーモンは、xinetd サービスと要求されたサービスの両方について、独自のアクセス制御ルールを確認します。deny ルールがクライアントと一致する場合、接続は破棄されます。それ以外の場合は、xinetd は要求されたサービスのインスタンスを開始し、接続の制御をそのサービスに渡します。



重要

TCP Wrappers アクセス制御を xinetd アクセス制御とともに使用する場合は注意が必要です。設定が間違っていると、望ましくない結果が発生する可能性があります。

#### 2.6.4.3.3. バインディングおよびリダイレクトオプション

xinetd のサービス設定ファイルは、サービスを IP アドレスにバインドし、そのサービスの受信要求を別の IP アドレス、ホスト名、またはポートにリダイレクトします。

バインディングはサービス固有の設定ファイルの bind オプションで制御され、サービスをシステム上の 1 つの IP アドレスにリンクします。これが設定されている場合、bind オプションは正しい IP アドレスへのリクエストのみがサービスにアクセスできるようにします。この方法を使用すると、要件に応じて異なるサービスを異なるネットワークインターフェースにバインドできます。

これは、複数のネットワークアダプターがあるシステムや、複数の IP アドレスを持つシステムで特に有用です。このようなシステムでは、安全ではないサービス（Telnet など）は、プライベートネットワークに接続され、インターネットに接続されたインターフェースにはアクセスしないように、プライベートネットワークに接続されたインターフェースでのみリッスンするように設定することができます。

redirect オプションは、IP アドレスまたはホスト名の後にポート番号を受け入れます。このサービスで、このサービスの要求を指定されたホストおよびポート番号にリダイレクトするように設定します。この機能を使用して、同じシステムの別のポート番号を参照し、要求を同じマシンの別の IP アドレスにリダイレクトし、要求を全く異なるシステムおよびポート番号、またはこれらのオプションの任意の組み合わせにリダイレクトできます。このため、システムで特定のサービスに接続するユーザーは、中断なしで別のシステムへルーティングし直す可能性があります。

xinetd デーモンは、要求しているクライアントマシンとホストが実際にサービスを提供し、この 2 つのシステム間でデータを転送するプロセスを起動して、このリダイレクトを実行できます。

bind および redirect オプションの利点は、一緒に使用される場合の最も明確に明確になります。

サービスをシステムの特定の IP アドレスにバインドし、このサービスの要求を、最初のマシンにのみ表示可能な 2 番目のマシンにリダイレクトすることで、内部システムは全く異なるネットワーク用にサービスを提供するために使用できます。このオプションを使用すると、マルチホームマシンの特定のサービスの公開を既知の IP アドレスに制限したり、そのサービスに対する要求を特にその目的で設定された別のマシンにリダイレクトしたりできます。

たとえば、Telnet サービスにこの設定が含まれるファイアウォールとして使用するシステムがあるとします。

```
service telnet
{
  socket_type  = stream
  wait   = no
  server   = /usr/kerberos/sbin/telnetd
  log_on_success  += DURATION USERID
  log_on_failure  += USERID
  bind                 = 123.123.123.123
  redirect              = 10.0.1.13 23
}
```

このファイルの bind および redirect オプションにより、マシン上の Telnet サービスが外部 IP アドレス(123.123.123.123)にバインドされ、インターネットに到達するようになります。さらに、123.123.123 に送信された Telnet サービスの要求 は、2 番目のネットワークアダプターを介して、ファイアウォールと内部システムのみがアクセスできる内部 IP アドレス(10.0.1.13)にリダイレクトされます。その後、ファイアウォールはシステム間の通信を送信します。接続システムは、実際に別のマシンに接続している 場合 は 123.123.123.123 に接続されていると見なします。

この機能は、ブロードバンド接続があり、1 つの固定 IP アドレスのみを持つユーザーに特に役立ちます。ネットワークアドレス変換(NAT)を使用する場合、内部のみの IP アドレスを使用しているゲートウェイマシンの背後にあるシステムは、ゲートウェイシステムからは利用できません。ただし、xinetd が制御する特定のサービスが bind および redirect オプションで設定されている場合、ゲートウェイマシンは、サービスを提供するように設定された外部システムと、特定の内部マシンとの間のプロキシーとして機能します。さらに、さまざまな xinetd アクセス制御およびロギングオプションも、追加の保護のために利用できます。

#### 2.6.4.3.4. リソース管理オプション

xinetd デーモンは、サービス拒否(DoS)攻撃から基本的な保護レベルを追加できます。以下は、このような攻撃の影響を制限するのに役立つディレクティブの一覧です。

- per_source : 送信元 IP アドレスごとのサービスの最大インスタンス数を定義します。これは整数のみを引数として受け入れ、xinetd.d/ ディレクトリー内のサービス固有 xinetd.conf の設定ファイルの両方で使用できます。

cps : 1 秒あたりの最大接続数を定義します。このディレクティブは、空白で区切られた 2 つの整数引数を取ります。最初の引数は、1 秒あたりのサービスに許可される最大接続数です。2 つ目の引数は、サービスを再度有効にする前に xinetd が待機する必要がある秒数です。これは整数のみを引数として受け入れ、xinetd.d/ ディレクトリーの xinetd.conf ファイルまたはサービス固有の設定ファイルのいずれかで使用できます。

- max_load : サービスの CPU 使用率または負荷平均のしきい値を定義します。浮動小数点番号引数を受け入れます。

  負荷平均は、ある時点でアクティブなプロセス数に関連してあります。負荷平均の詳細は uptime、、who、および procinfo コマンドを参照してください。

xinetd により多くのリソース管理オプションを利用できます。詳細については xinetd.conf の man ページを参照してください。

### 2.6.5. その他のリソース

TCP Wrapper および xinetd の詳細は、システムのドキュメントおよびインターネットから入手できます。

#### 2.6.5.1. インストールされた TCP Wrapper ドキュメンテーション

システムのドキュメントは、TCP Wrapper、xinetd、およびアクセス制御の追加設定オプションを検索するのに適しています。

- /usr/share/doc/tcp_wrappers-*<version>*/ : このディレクトリーには、TCP Wrapper の仕組みと、存在するさまざまなホスト名およびホストのアドレスの偽装リスクを記述する README ファイルが含まれます。

- /usr/share/doc/xinetd-*<version>*/ : このディレクトリーには、アクセス制御の要素や、/etc/xinetd.d/ ディレクトリー内のサービス固有の設定 sample.conf ファイルを変更するための様々な README 注意のあるファイルに関するファイルが含まれます。

- TCP Wrapper および xinetd関連の man ページ - TCP Wrapper および xinetd に関連するさまざまなアプリケーションおよび設定ファイルの man ページが多数存在します。以下は、より重要な man ページです。

  サーバーアプリケーション

- man xinetd : xinetd の man ページです。

設定ファイル

- man 5 hosts_access : TCP Wrappers の man ページは、アクセス制御ファイルをホストします。

- man hosts_options : TCP Wrappers オプションフィールドの man ページです。

- man xinetd.conf : の man ページでは、xinetd 設定オプションが一覧表示されます。

#### 2.6.5.2. 関連書籍

- 『hacking Linux Exposed』 by Frank Hatch, andvillevilletz; Osbourne/McGraw-Hill - TCP Wrapper および xinetd に関する情報のある優れたセキュリティーリソースです。

## 2.7. 仮想プライベートネットワーク(VPN)のセキュリティー保護

Red Hat Enterprise Linux 6 での *仮想プライベートネットワーク* （）VPN）は、Libreswan アプリケーションで対応している IPsec トンネリングプロトコルを使用して設定できます。Libreswan は Openswan アプリケーションのフォークです。ドキュメントの例は交換可能となります。NetworkManager IPsec プラグインは、と呼ばれてい NetworkManager-openswanます。

注記

Libreswan は、Red Hat Enterprise Linux 6.8 で IPsec の推奨実装として Openswan に置き換えられました。6.8 よりも前のバージョンからのアップグレードを実行すると、openswan パッケージはに置き換え libreswanます。

Libreswan は、Red Hat Enterprise Linux 6 で利用可能なオープンソースのユーザー空間の IPsec 実装です。*インターネット鍵交換* （）を使用します。IKE）プロトコル。IKE バージョン 1 および 2 は、ユーザーレベルのデーモンとして実装されます。ip xfrm コマンドでは、手動による鍵確立が可能ですが、これは推奨されません。Libreswan は、netlink を使用して暗号化鍵を転送する Linux カーネルでインターフェースします。Linux カーネルでパケットの暗号化と復号が行われます。

Libreswan は、*ネットワークセキュリティーサービス* を使用します（NSS）暗号化ライブラリー。*連邦情報処理標準* （）に必要です。FIPS）セキュリティーコンプライアンス。

### 2.7.1. Libreswan を使用した IPsec VPN

Libreswan をインストールするには、root で以下のコマンドを実行します。libreswan パッケージは Extras リポジトリーから入手できます。このリポジトリーは、インストールを成功させるには有効にする必要があります。「How to enable/disable a repository using Red Hat Subscription Manager?」を参照してください。（ Extras リポジトリーの ID はです rhel-6-server-extras-rpms。）

```
~]# yum install libreswan
```

Libreswan がインストールされていることを確認するには、以下のコマンドを発行します。

```
~]$ yum info libreswan
```

Libreswan を新規インストールした後に、NSS データベースはインストールプロセスの一部として初期化される必要があります。ただし、新しいデータベースを起動する必要があります。最初に、以下のように古いデータベースを削除します。

```
~]# rm /etc/ipsec.d/*db
```

次に、新しい NSS データベースを初期化するには、root で以下のコマンドを発行します。

```
~]# ipsec initnss
Initializing NSS database
See 'man pluto' if you want to protect the NSS database with a password
```

Libreswan が提供する ipsec デーモンを起動するには、root で以下のコマンドを発行します。

```
~]# service ipsec start
```

デーモンが現在実行していることを確認します。

```
~]$ service ipsec status
pluto (pid  3496) is running...
```

Libreswan がシステムの起動時に起動するようにするには、root で以下のコマンドを実行します。

```
~]# chkconfig ipsec on
```

中間およびホストベースのファイアウォールを、ipsec サービスを許可するように設定します。ファイアウォール 「ファイアウォール」 の詳細と、特定のサービスが通過できるようには、を参照してください。Libreswan では、次のパケットを許可するファイアウォールが必要です。

- インターネット鍵交換 用UDP ポート 500（）IKE）プロトコル
- IKE NAT-TraversalのUDP ポート 4500
- カプセル化された セキュリティーペイロード （）ESP） IPsec パケット
- 認証されたヘッダー （プロトコル 51）AH） IPsec パケット（一般的でない）

Libreswan を使用した IPsec VPN の設定例を 3 つ示します。1 つ目は、ホストをセキュアに通信するために、2 つのホストを 1 つ接続する方法です。2 つのサイトを 1 つのネットワークに接続し、1 つのネットワークを構成する例を以下に示します。3 つ目は、このコンテキスト *でロード* リーダーとして知られるローミングユーザーをサポートします。

### 2.7.2. Libreswan を使用した VPN 設定

Libreswan は用語を使用しません。 「source」 または 「destination」.代わりに、用語を使用します。 「left」 ならびに 「right」 終了点（ホスト）を参照します。これにより、ほとんどの管理者はを使用しますが、ほとんどの場合で両方の終了点で同じ設定を使用できます。 「left」 ローカルホストの場合は、 「right」 リモートホスト。

エンドポイントの認証には、一般的に使用される 3 つの方法があります。

- 生の RSA 鍵は、静的なホスト間またはサブネット間の IPsec 設定に使用されます。ホストは、相互の公開 RSA 鍵を使用して手動で設定します。この方法は、1 ダース以上のホストで、互いに IPsec トンネルを設定する必要がある場合には、適切に調整されません。
- X.509 証明書は、共通の IPsec ゲートウェイに接続する必要のあるホストが多数存在する

大規模なデプロイメントに一般的に使用されます。中央の *認証局*（）CA）は、ホストまたはユーザーに RSA 証明書の署名に使用されます。この中央 CA は、個別のホストまたはユーザーの取り消しを含む、信頼のリレーを行います。

- *事前共有鍵*（）PSK）は、最も簡単な認証方法です。PSK はランダムな文字で構成されており、長さが 20 文字以上になります。非ランダムな PSK と短い PSK の所属により、これは認証の最も安全ではないため、生の RSA 鍵または証明書ベースの認証のいずれかを使用することが推奨されます。

### 2.7.3. Libreswan を使用したホスト間の VPN

Libreswan がホスト間の IPsec VPN を作成するように設定するには、と呼ばれる 2 つのホスト間で、 「left」 ならびに 「right」両方のホストで root で以下のコマンドを入力します（）「left」 ならびに 「right」）で、生の RSA 鍵のペアを新たに作成します。

```
~]# ipsec newhostkey --configdir /etc/ipsec.d \
--output /etc/ipsec.d/www.example.com.secrets
Generated RSA key pair using the NSS database
```

これにより、ホストの RSA 鍵ペアが生成されます。RSA 鍵の生成プロセスには数分かかる場合があります（特にエントロピーが少ない仮想マシン）。

公開鍵を表示するには、いずれかのホストで root で以下のコマンドを発行します。たとえば、で公開鍵を表示するには、 「left」 ホスト、以下を実行します。

```
~]# ipsec showhostkey --left
ipsec showhostkey loading secrets from "/etc/ipsec.secrets"
ipsec showhostkey loading secrets from "/etc/ipsec.d/www.example.com.secrets"
ipsec showhostkey loaded private key for keyid: PPK_RSA:AQOjAKLlL
 # rsakey AQOjAKLlL
 leftrsasigkey=0sAQOjAKLlL4a7YBv [...]
```

以下で説明されているように、設定ファイルに追加するには、このキーが必要です。

シークレット部分は /etc/ipsec.d/*.db ファイル（とも呼ばれます）に保存されます。 「NSS データベース」.

このホスト間トンネルの設定ファイルを作成するには、行 leftrsasigkey= と上 rightrsasigkey= からの行が、/etc/ipsec.d/ ディレクトリーに配置されるカスタム設定ファイルに追加されます。

root として実行しているエディターを使用して、以下の形式で適切な名前のファイルを作成します。

```
/etc/ipsec.d/my_host-to-host.conf
```

以下のようにファイルを編集します。

```
conn mytunnel
    leftid=@west.example.com
    left=192.1.2.23
    leftrsasigkey=0sAQOrlo+hOafUZDlCQmXFrje/oZm [...] W2n417C/4urYHQkCvuIQ==
    rightid=@east.example.com
    right=192.1.2.45
    rightrsasigkey=0sAQO3fwC6nSSGgt64DWiYZzuHbc4 [...] D/v8t5YTQ==
    authby=rsasig
    # load and initiate automatically
    auto=start
```

左側のホストと右のホストの両方で同じ設定ファイルを使用できます。この場合は、自動検出されます。 「left」 または 「right」.いずれかのホストがモバイルホスト（ IP アドレスを事前に認識しない）である場合は、モバイルホストでを IP アドレス %defaultroute として使用します。これにより、動的 IP アドレスが自動的に取得されます。着信モバイルホストからの接続を受け入れる静的ホストで、IP アドレスにを使用してモバイルホスト %any を指定します。

の leftrsasigkey 値がから取得されていることを確認します。 「left」 host と rightrsasigkey value はから取得されます。 「right」 host。

ipsec を再起動して、新しい設定を読み取ります。

```
~]# service ipsec --full-restart
```

トンネルがすぐに確立されていることを確認するには、以下のコマンドを入力します。

```
~]# ipsec whack --trafficstatus
```

/etc/ipsec.d/*.conf ファイルの auto=start オプションを使用しない場合やトンネルが正常に確立されていない場合は、root で次のコマンドを使用して IPsec トンネルを読み込みます。

```
~]# ipsec auto --add mytunnel
```

トンネルを起動するには、root で、左側のまたは右側で以下のコマンドを実行します。

```
~]# ipsec auto --up mytunnel
```

#### 2.7.3.1. Libreswan を使用したホスト間の VPN の検証

IKE ネゴシエーションは UDP ポート 500 で行われます。IPsec パケットは、カプセル化された セキュリティーペイロード (ESP)パケットとして表示されます。VPN 接続が NAT ルーターを通過する必要がある場合、ESP パケットはポート 4500 の UDP パケットでカプセル化されます。

パケットが VPN トンネル経由で送信されていることを確認するには、root で以下の形式でコマンドを実行します。

```
~]# tcpdump -n -i interface esp or udp port 500 or udp port 4500
00:32:32.632165 IP 192.1.2.45 > 192.1.2.23: ESP(spi=0x63ad7e17,seq=0x1a), length 132
00:32:32.632592 IP 192.1.2.23 > 192.1.2.45: ESP(spi=0x4841b647,seq=0x1a), length 132
00:32:32.632592 IP 192.0.2.254 > 192.0.1.254: ICMP echo reply, id 2489, seq 7, length 64
00:32:33.632221 IP 192.1.2.45 > 192.1.2.23: ESP(spi=0x63ad7e17,seq=0x1b), length 132
00:32:33.632731 IP 192.1.2.23 > 192.1.2.45: ESP(spi=0x4841b647,seq=0x1b), length 132
00:32:33.632731 IP 192.0.2.254 > 192.0.1.254: ICMP echo reply, id 2489, seq 8, length 64
00:32:34.632183 IP 192.1.2.45 > 192.1.2.23: ESP(spi=0x63ad7e17,seq=0x1c), length 132
00:32:34.632607 IP 192.1.2.23 > 192.1.2.45: ESP(spi=0x4841b647,seq=0x1c), length 132
00:32:34.632607 IP 192.0.2.254 > 192.0.1.254: ICMP echo reply, id 2489, seq 9, length 64
00:32:35.632233 IP 192.1.2.45 > 192.1.2.23: ESP(spi=0x63ad7e17,seq=0x1d), length 132
00:32:35.632685 IP 192.1.2.23 > 192.1.2.45: ESP(spi=0x4841b647,seq=0x1d), length 132
00:32:35.632685 IP 192.0.2.254 > 192.0.1.254: ICMP echo reply, id 2489, seq 10, length 64
```

*interface* は、トラフィックを伝送できるインターフェースです。tcpdump でキャプチャーを終了するには、を押し Ctrl+Cます。

注記

tcpdump コマンドは、IPsec と予期せずに対話します。送信プレーンテキストパケットではなく、送信暗号化パケットのみが表示されます。暗号化された受信パケットと、復号化された受信パケットが表示されます。可能であれば、エンドポイント自体ではなく、2 つのマシン間のルーターで tcpdump を実行します。

### 2.7.4. Libreswan を使用したサイト間の VPN

2 つのネットワークを参加させるサイト間の IPsec VPN を作成するには、1 つ以上のサブネットからのトラフィックが通過できるように設定されるエンドポイントという 2 つのホストの間で IPsec トンネルが作成されます。したがって、ネットワークのリモート部分へのゲートウェイとして見なすことが

できます。サイト間の VPN の設定は、設定ファイル内で複数のネットワークまたはサブネットを指定する必要がある点のみが、ホスト間の VPN とは異なります。

Libreswan がサイト間の IPsec VPN を作成するようにするには、最初に、の説明に従ってホスト間の IPsec VPN を設定 「Libreswan を使用したホスト間の VPN」 し、ファイルをなどの適切な名前を持つファイルにコピーまたは移動し /etc/ipsec.d/my_site-to-site.confます。root で実行中のエディターを使用して、以下の /etc/ipsec.d/my_site-to-site.conf ようにカスタム設定ファイルを編集します。

```
conn mysubnet
    also=mytunnel
    leftsubnet=192.0.1.0/24
    rightsubnet=192.0.2.0/24

conn mysubnet6
    also=mytunnel
    connaddrfamily=ipv6
    leftsubnet=2001:db8:0:1::/64
    rightsubnet=2001:db8:0:2::/64

conn mytunnel
   leftid=@west.example.com
   left=192.1.2.23
   leftrsasigkey=0sAQOrlo+hOafUZDlCQmXFrje/oZm [...] W2n417C/4urYHQkCvuIQ==
   rightid=@east.example.com
   right=192.1.2.45
   rightrsasigkey=0sAQO3fwC6nSSGgt64DWiYZzuHbc4 [...] D/v8t5YTQ==
   authby=rsasig
```

トンネルを稼動させるには、Libreswan を再起動するか、root で以下のコマンドを使用してすべての接続を手動で読み込み、開始します。

```
~]# ipsec auto --add mysubnet
```

```
~]# ipsec auto --add mysubnet6
```

```
~]# ipsec auto --add mytunnel
```

```
~]# ipsec auto --up mysubnet
104 "mysubnet" #1: STATE_MAIN_I1: initiate
003 "mysubnet" #1: received Vendor ID payload [Dead Peer Detection]
003 "mytunnel" #1: received Vendor ID payload [FRAGMENTATION]
106 "mysubnet" #1: STATE_MAIN_I2: sent MI2, expecting MR2
108 "mysubnet" #1: STATE_MAIN_I3: sent MI3, expecting MR3
003 "mysubnet" #1: received Vendor ID payload [CAN-IKEv2]
004 "mysubnet" #1: STATE_MAIN_I4: ISAKMP SA established {auth=OAKLEY_RSA_SIG
cipher=aes_128 prf=oakley_sha group=modp2048}
117 "mysubnet" #2: STATE_QUICK_I1: initiate
```

```
004 "mysubnet" #2: STATE_QUICK_I2: sent QI2, IPsec SA established tunnel mode
{ESP=>0x9414a615 <0x1a8eb4ef xfrm=AES_128-HMAC_SHA1 NATOA=none NATD=none
DPD=none}
```

```
~]# ipsec auto --up mysubnet6
003 "mytunnel" #1: received Vendor ID payload [FRAGMENTATION]
117 "mysubnet" #2: STATE_QUICK_I1: initiate
004 "mysubnet" #2: STATE_QUICK_I2: sent QI2, IPsec SA established tunnel mode
{ESP=>0x06fe2099 <0x75eaa862 xfrm=AES_128-HMAC_SHA1 NATOA=none NATD=none
DPD=none}
```

```
~]# ipsec auto --up mytunnel
104 "mytunnel" #1: STATE_MAIN_I1: initiate
003 "mytunnel" #1: received Vendor ID payload [Dead Peer Detection]
003 "mytunnel" #1: received Vendor ID payload [FRAGMENTATION]
106 "mytunnel" #1: STATE_MAIN_I2: sent MI2, expecting MR2
108 "mytunnel" #1: STATE_MAIN_I3: sent MI3, expecting MR3
003 "mytunnel" #1: received Vendor ID payload [CAN-IKEv2]
004 "mytunnel" #1: STATE_MAIN_I4: ISAKMP SA established {auth=OAKLEY_RSA_SIG
cipher=aes_128 prf=oakley_sha group=modp2048}
117 "mytunnel" #2: STATE_QUICK_I1: initiate
004 "mytunnel" #2: STATE_QUICK_I2: sent QI2, IPsec SA established tunnel mode
{ESP=>0x16bca4f7 >0x9c2ae273 xfrm=AES_128-HMAC_SHA1 NATOA=none NATD=none
DPD=none}
```

#### 2.7.4.1. Libreswan を使用したサイト間の VPN の確認

パケットが VPN トンネル経由で送信されていることを確認することは、の説明と同じ手順です「Libreswan を使用したホスト間の VPN の検証」。

### 2.7.5. Libreswan を使用したサイト間のシングルトンネリング VPN

多くの場合、サイト間トンネルを構築する場合には、ゲートウェイはパブリック IP アドレスではなく内部 IP アドレスを使用して相互に通信する必要があり ます。これは、1 つのトンネルを使用して実行できます。ホスト名が traffic の左ホストに内部 IP アドレス 192.0.1.254 と右のホスト が あり、ホスト名 east に内部 IP アドレス 192.0.2.254 がある場合は、単一のトンネルを使用した以下の設定を使用できます。

```
conn mysubnet
    leftid=@west.example.com
    leftrsasigkey=0sAQOrlo+hOafUZDlCQmXFrje/oZm [...] W2n417C/4urYHQkCvuIQ==
    left=192.1.2.23
    leftsourceip=192.0.1.254
    leftsubnet=192.0.1.0/24
    rightid=@east.example.com
    rightrsasigkey=0sAQO3fwC6nSSGgt64DWiYZzuHbc4 [...] D/v8t5YTQ==
    right=192.1.2.45
    rightsourceip=192.0.2.254
```

```
    rightsubnet=192.0.2.0/24
    auto=start
    authby=rsasig
```

### 2.7.6. Libreswan を使用したサブネットの追加

**IPsec は、ハブアンドスポークアーキテクチャーにデプロイされることが多くあります。各リーフノードには、大規模な範囲に含まれる IP 範囲があります。そのままはハブ経由で相互に通信します。これは *サブネットの拡張と呼ばれ* ます。以下の例では、10.0.0.0/8 のヘッドオフィスと、小規模な /24 サブネットを使用する 2 つのブランチを設定します。**

**オフィスでは以下のようになります。**

```
conn branch1
    left=1.2.3.4
    leftid=@headoffice
    leftsubnet=0.0.0.0/0
    leftrsasigkey=0sA[...]
    #
    right=5.6.7.8
    rightid=@branch1
    rightsubnet=10.0.1.0/24
    rightrsasigkey=0sAXXXX[...]
    #
    auto=start
    authby=rsasig

conn branch2
    left=1.2.3.4
    leftid=@headoffice
    leftsubnet=0.0.0.0/0
    leftrsasigkey=0sA[...]
    #
    right=10.11.12.13
    rightid=@branch2
    rightsubnet=10.0.2.0/24
    rightrsasigkey=0sAYYYY[...]
    #
    auto=start
    authby=rsasig
```

**「」で行います。「branch1」 オフィスでは、同じ接続を使用します。さらに、パススルー接続を使用して、ローカル LAN トラフィックがトンネルを介して送信されないようにします。**

```
conn branch1
    left=1.2.3.4
    leftid=@headoffice
    leftsubnet=0.0.0.0/0
    leftrsasigkey=0sA[...]
```

```
    #
    right=10.11.12.13
    rightid=@branch2
    rightsubnet=10.0.1.0/24
    rightrsasigkey=0sAYYYY[...]
    #
    auto=start
    authby=rsasig

conn passthrough
    left=1.2.3.4
    right=0.0.0.0
    leftsubnet=10.0.1.0/24
    rightsubnet=10.0.1.0/24
    authby=never
    type=passthrough
    auto=route
```

#### 2.7.7. Libreswan を使用したロードエリアアクセス VPN

ロードラーは、ノート PC など、動的に IP アドレスが割り当てられたモバイルクライアントを使用するユーザーです。これらは証明書を使用して認証されます。

サーバー上では以下の設定になります。

```
conn roadwarriors
    left=1.2.3.4
    # if access to the LAN is given, enable this
    #leftsubnet=10.10.0.0/16
    leftcert=gw.example.com
    leftid=%fromcert
    right=%any
    # trust our own Certificate Agency
    rightca=%same
    # allow clients to be behind a NAT router
    rightsubnet=vhost:%priv,%no
    authby=rsasig
    # load connection, don't initiate
    auto=add
    # kill vanished roadwarriors
    dpddelay=30
    dpdtimeout=120
    dpdaction=%clear
```

モバイルクライアントでは、上記の設定に多少変更を加える必要があります。

```
conn roadwarriors
    # pick up our dynamic IP
    left=%defaultroute
```

```
leftcert=myname.example.com
leftid=%fromcert
# right can also be a DNS hostname
right=1.2.3.4
# if access to the remote LAN is required, enable this
#rightsubnet=10.10.0.0/16
# trust our own Certificate Agency
rightca=%same
authby=rsasig
# Initiate connection
auto=start
```

### 2.7.8. Libreswan を使用したロードロードアクセス VPN および X.509 による XAUTH

**Libreswan では、XAUTH IPsec 拡張を使用して接続を確立する際に、VPN クライアントに IP アドレスと DNS 情報をネイティブに割り当てる方法を利用できます。XAUTH は、PSK または X.509 証明書を使用してデプロイできます。X.509 を使用する方が安全です。クライアント証明書は、証明書失効リストまたは *オンライン証明書ステータスプロトコル（）* で取り消すことができます。OCSP).X.509 証明書では、個々のクライアントはサーバーの権限を借用することができません。PSK(Group Password)を使用すると、これは理論的に可能になります。**

**XAUTH は、追加でユーザー名とパスワードで自身を識別するために VPN クライアントを必要とします。Google Authenticator または RSA SecureID トークンなどのワンタイムパスワード(OTP)では、ワンタイムトークンがユーザーパスワードに追加されます。**

**XAUTH には、以下の 3 つのバックエンドがあります。**

**xauthby=pam**

**これにより、の設定を使用 /etc/pam.d/pluto してユーザーを認証します。pam は、単独でさまざまなバックエンドを使用するように設定できます。システムアカウントのユーザーパスワードスキーム、LDAP ディレクトリー、RADIUS サーバー、またはカスタムパスワード認証モジュールを使用できます。**

**xauthby=file**

**これは、設定ファイルを使用します /etc/ipsec.d/passwd （混同しない /etc/ipsec.d/nsspassword）。このファイルの形式は Apache .htpasswd ファイルと似ていますが、Apache htpasswd コマンドを使用してこのファイルにエントリーを作成できます。ただし、ユーザー名とパスワードの後には、リモートユーザーに VPN を提供するために使用する IPsec 接続の接続名が含まれる 3 番目のコラムが必要になります。たとえば、を使用してリモートユーザーに VPN を conn remoteusers 提供する場合、パスワードファイルのエントリーは以下のようになります。**

```
user1:$apr1$MIwQ3DHb$1I69LzTnZhnCT2DPQmAOK.:remoteusers
```

注記： **htpasswd** コマンドを使用する場合は、各行に ***user:password*** 部分の後に接続名を手動で追加する必要があります。

**xauthby=alwaysok**

サーバーは、常に **XAUTH** ユーザーとパスワードの組み合わせが正しいことになります。クライアントはユーザー名とパスワードを指定する必要がありますが、サーバーはこれを無視します。これは、ユーザーが **X.509** 証明書ですでに特定されている場合、または **XAUTH** バックエンドを使用せずに **VPN** をテストする場合にのみ使用してください。

**X.509** 証明書が含まれる設定の例：

```
conn xauth-rsa
    auto=add
    authby=rsasig
    pfs=no
    rekey=no
    left=ServerIP
    leftcert=vpn.example.com
    #leftid=%fromcert
    leftid=vpn.example.com
    leftsendcert=always
    leftsubnet=0.0.0.0/0
    rightaddresspool=10.234.123.2-10.234.123.254
    right=%any
    rightrsasigkey=%cert
    modecfgdns1=1.2.3.4
    modecfgdns2=8.8.8.8
    modecfgdomain=example.com
    modecfgbanner="Authorized Access is allowed"
    leftxauthserver=yes
    rightxauthclient=yes
    leftmodecfgserver=yes
    rightmodecfgclient=yes
    modecfgpull=yes
    xauthby=pam
    dpddelay=30
    dpdtimeout=120
    dpdaction=clear
    ike_frag=yes
    # for walled-garden on xauth failure
    # xauthfail=soft
    #leftupdown=/custom/_updown
```

をソフトに設定すると、認証の失敗 **xauthfail** は無視され、**VPN** はユーザーが適切に認証されたかのように設定されます。カスタムのアップダウンスクリプトを使用して、環境変数を確認することができ **XAUTH_FAILED**ます。このようなユーザーは、たとえば **iptables DNAT** を使用してにリダイレク

トできます。 「グラデーション」 ここで管理者に連絡したり、サービスに有料サブスクリプションを更新したりできます。

VPN クライアントは modecfgdomain 値と DNS エントリーを使用して、指定したドメインのクエリーを指定されたネームサーバーにリダイレクトします。これにより、ローミングユーザーは内部 DNS 名を使用して内部のみのリソースにアクセスできます。

0.0.0.0/0 では、トンネリング設定要求 leftsubnet はクライアントに自動的に送信されます。たとえば、を使用すると leftsubnet=10.0.0.0/8、VPN クライアントは VPN を介して 10.0.0.0/8 のトラフィックのみを送信します。

### 2.7.9. その他のリソース

以下の資料は、Libreswan および ipsec デーモンに関するその他の情報を提供します。

#### 2.7.9.1. インストールされているドキュメント

- ipsec(8) の man ページ： ipsec のコマンドオプションを説明しています。
- ipsec.conf(5) の man ページ： ipsec の設定に関する情報が含まれています。
- ipsec.secrets(5) の man ページ： ipsec.secrets ファイルの形式を説明しています。
- ipsec_auto(8) man ページ： 鍵の自動交換を使用して確立された Libreswan IPsec 接続を操作する auto コマンドラインクライアントの使用を説明しています。
- ipsec_rsasigkey(8) の man ページ： RSA 署名鍵の生成に使用するツールが説明されています。
- /usr/share/doc/libreswan-*version*/README.nss：Libreswan pluto デーモンで使用する NSS 暗号ライブラリーで、生の RSA 鍵と証明書を使用するコマンドを説明します。

#### 2.7.9.2. オンラインドキュメント

https://libreswan.org

アップストリームプロジェクトの Web サイトです。

http://www.mozilla.org/projects/security/pki/nss/

Network Security Services(NSS)プロジェクト。

## 2.8. ファイアウォール

情報セキュリティーは通常、製品ではなくプロセスとして見なされます。ただし、標準のセキュリティー実装は通常、アクセス権限を制御し、承認、識別可能、およびトレース可能なユーザーに対してネットワークリソースを制限するために、いくつかの専用メカニズムを採用しています。Red Hat Enterprise Linux には、管理者およびセキュリティーエンジニアがネットワークレベルのアクセス制御の問題で役立つツールがいくつか含まれています。

ファイアウォールは、ネットワークセキュリティー実装の中核となるコンポーネントの 1 つです。いくつかのベンダーのマーケットファイアウォールソリューションは、市場のあらゆるレベルに対応します。1 つの PC からデータセンターソリューションを保護することで、重要なエンタープライズ情報を保護できます。ファイアウォールは、Cisco、Nokia、Sonicwall のファイアウォールアプライアンスなどのスタンドアロンのハードウェアソリューションになります。Checkpoint、mcAfee などのベンダーは、ホームおよびビジネスマーケット用のプロプライエタリーソフトウェアファイアウォールソリューションも開発しています。

ハードウェアとソフトウェアのファイアウォールの違いとは別に、ファイアウォール機能とは別のソリューションを分離する方法にも違いがあります。一般的なファイアウォールの種類と機能について 表 2.6「ファイアウォールの種類」 詳しく説明します。

**表2.6 ファイアウォールの種類**

| 方法 | 説明 | 利点 | デメリット |
|---|---|---|---|
| NAT | *ネットワークアドレス変換* (NAT)は、プライベート IP サブネットワークをパブリック IP アドレスの 1 つまたは小さなプールの背後で配置し、すべての要求を複数のソースではなく 1 つのソースにマスカレードします。Linux カーネルは、Netfilter カーネルサブシステムを介して、NAT 機能が組み込まれています。 | LAN 上のマシンに対して透過的に設定できます。<br><br>1 つ以上の外部 IP アドレスの背後にある多くのマシンおよびサービスを保護すると、管理作業が容易になります。<br><br>LAN からのユーザーアクセスの制限は、NAT ファイアウォール/ゲートウェイでポートを開いて閉じることで設定できます。 | ユーザーがファイアウォール外のサービスに接続したら、悪意のあるアクティビティーを防ぐことはできません。 |

| 方法 | 説明 | 利点 | デメリット |
| --- | --- | --- | --- |
| パケットフィルター | パケットフィルタリングファイアウォールは、LAN を通過する各データパケットを読み込みます。ファイアウォール管理者が実装するプログラム可能なルールのセットに基づいて、ヘッダー情報でパケットを読み取りおよび処理できます。Linux カーネルには、Netfilter カーネルサブシステムを介して、パケットフィルタリング機能が組み込まれています。 | フロントエンドユーティリティーでカスタマイズ **iptables** が可能です。<br><br>すべてのネットワークアクティビティーがアプリケーションレベルではなくルーターレベルでフィルターされるため、クライアント側でカスタマイズする必要はありません。<br><br>パケットはプロキシー経由で送信されないため、クライアントからリモートホストへの直接接続により、ネットワークパフォーマンスが向上します。 | プロキシーファイアウォールのようなコンテンツのパケットをフィルタリングすることはできません。<br><br>プロトコル層でパケットを処理しますが、アプリケーション層でパケットをフィルターすることはできません。<br><br>複雑なネットワークアーキテクチャーにより、パケットフィルタリングのルールを確立することが困難になります。特に、*IP マスカレード*またはローカルサブネット、DMZ ネットワークと組み合わせると、パケットフィルタリングのルールを確立するのが困難になります。 |
| Proxy | プロキシーファイアウォールは、LAN クライアントからの特定のプロトコルまたはタイプのすべての要求をプロキシーマシンに絞り込み、ローカルクライアントの代わりにインターネットへの要求を行います。プロキシーマシンは、悪意のあるリモートユーザーと内部ネットワーククライアントマシンとの間のバッファーとして機能します。 | 管理者は、LAN 外のアプリケーションやプロトコルを制御できます。<br><br>プロキシーサーバーの中には、インターネット接続を使用して要求しなくても、頻繁にアクセスされるデータをローカルでキャッシュできるものもあります。これにより、帯域幅の消費を減らすことができます。<br><br>プロキシーサービスはログに記録および監視できるので、ネットワーク上のリソース使用率をより詳細に制御できます。 | プロキシーは通常、アプリケーション固有（HTTP、Telnet など）、またはプロトコル制限（ほとんどのプロキシーは TCP 接続サービスでのみ機能します）です。<br><br>アプリケーションサービスはプロキシーの背後で実行できないため、アプリケーションサーバーは別のネットワークセキュリティー形式を使用する必要があります。<br><br>すべての要求および送信がクライアントから直接リモートサービスへ直接渡すのではなく、プロキシーがネットワークのボトルネックとなる可能性があります。 |

### 2.8.1. netfilter および IPTables

**Linux カーネルは、*Netfilter* と呼ばれる強力なネットワークサブシステムを特長としています。Netfilter サブシステムは、ステートフルまたはステートレスパケットフィルタリング、NAT および IP マスカレードサービスを提供します。また、netfilter には、高度なルーティングおよび接続状態管理用に *IP* ヘッダー情報をマスジングする機能もあります。netfilter は、iptables ツールを使用して制御します。**

#### 2.8.1.1. iptables の概要

Netfilter の機能および柔軟性は、iptables 管理ツール（以前のコマンドラインツール）を使用して実装されます。これは ipchains、Linux カーネル 2.4 以上の Netfilter/iptables に置き換えられます。

iptables Netfilter サブシステムを使用して、ネットワーク接続、検査、処理を強化します。iptables 高度なロギング、事前およびルーティング後のアクション、ネットワークアドレス変換、およびポート転送すべてはすべて 1 つのコマンドラインインターフェースに保管されます。

本セクションでは、の概要を説明し iptablesます。詳細はを参照してください 「iptables」。

### 2.8.2. ファイアウォールの基本設定

ビルド内のファイアウォールは、発生が広がらないように試行するのと同様に、悪意のあるソフトウェアがコンピューターに分散されるのを防ぐコンピューターファイアウォールが試行されます。また、承認されていないユーザーがコンピューターにアクセスできないようにするのに役立ちます。

デフォルトの Red Hat Enterprise Linux インストールでは、コンピューターまたはネットワークと信頼できないネットワーク（インターネットなど）との間にファイアウォールが存在します。コンピューターのリモートユーザーがアクセスできるサービスを決定します。ファイアウォールが適切に設定されていると、システムのセキュリティーが大幅に向上する可能性があります。インターネット接続のある Red Hat Enterprise Linux システムのファイアウォールを設定することが推奨されます。

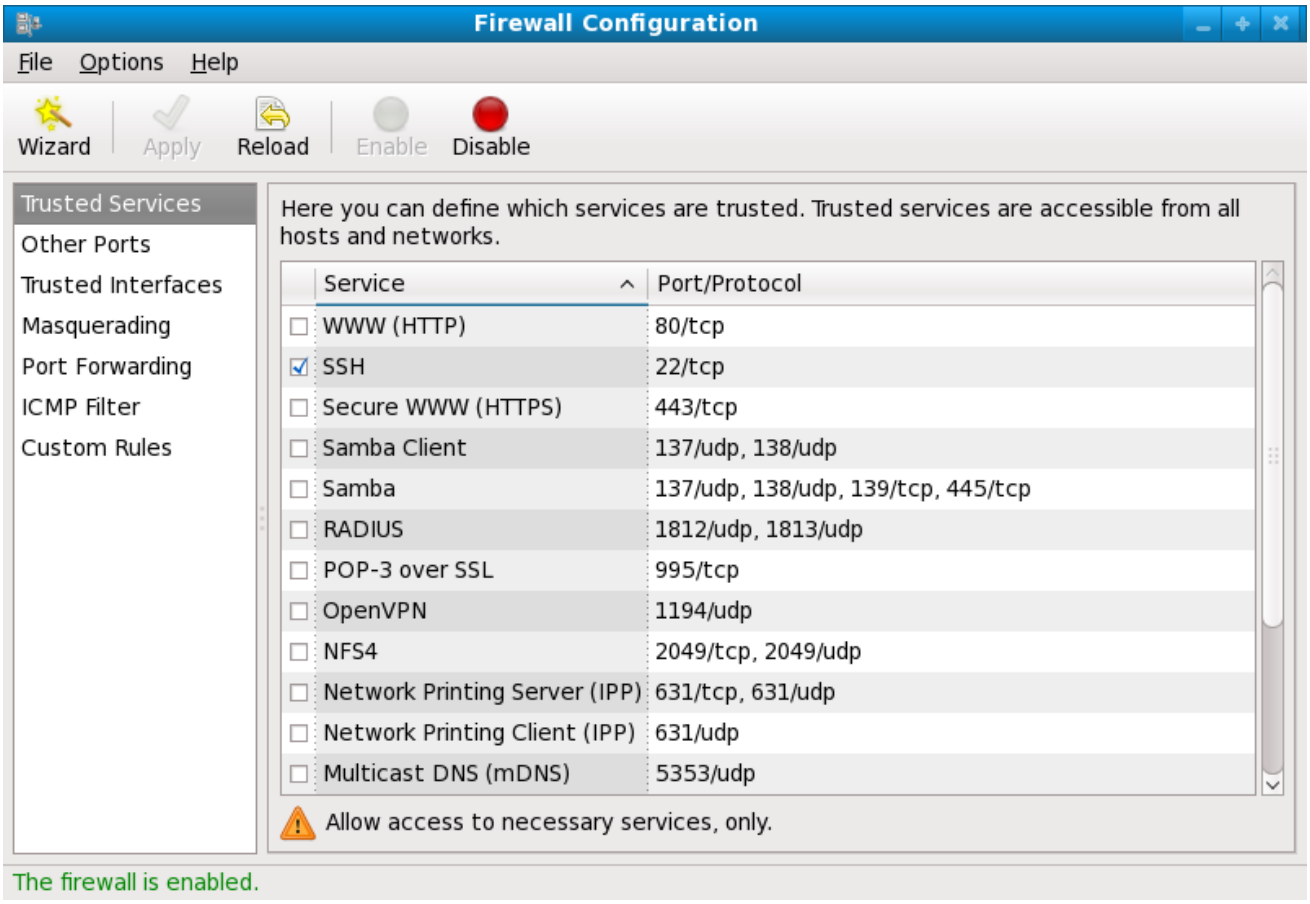
#### 2.8.2.1. ファイアウォール設定ツール

Red Hat Enterprise Linux インストールの ファイアウォール設定 画面で、基本的なファイアウォールを有効にし、特定のデバイス、着信サービス、ポートを許可するオプションを指定できました。

インストール後に Firewall Configuration Tool を使用すると、この設定を変更できます。

このアプリケーションを起動するには、パネル System → Administration → Firewall からを選択するか、シェルプロンプト system-config-firewall でを入力します。

**図2.5 ファイアウォール設定ツール**




[D]

注記

ファイアウォール設定ツール は基本的なファイアウォールのみを設定します。システムに複雑なルールが必要な場合は、で特定のルールの設定 「iptables」 に関する詳細を参照してください iptables。

Red Hat Enterprise Linux 6.5 以降、iptables サービスおよび ip6tables サービスは、デフォルト設定を適用できない場合にフォールバックのファイアウォール設定を割り当てる機能を提供するようになりました。からのファイアウォールルールの適用に /etc/sysconfig/iptables 失敗した場合、フォールバックファイルは存在する場合に適用されます。フォールバックファイルはという名前で /etc/sysconfig/iptables.fallback、と同じファイル形式を使用し /etc/sysconfig/iptablesます。フォールバックファイルの適用に失敗した場合、フォールバックはありません。フォールバックファイルを作成するには、標準のファイアウォール設定ツールを使用して、名前を変更したか、フォールバックファイルにコピーします。

ip6tables サービスでは、上記の例 iptables ip6tables で発生したをすべてに置き換えます。

警告

iptables ユーティリティーを直接使用してカスタムパケットフィルタリングルールを設定している場合（を参照 「iptables」）、system-config-firewall ユーティリティーを実行すると、これらのカスタムルールがすぐに消去されます。

#### 2.8.2.2. ファイアウォールの有効化および無効化

ファイアウォールには、以下のいずれかのオプションを選択します。

- disabled- ファイアウォールを有効にすると、システムへの完全なアクセスが可能になり、セキュリティーの確認は行われません。これは、信頼できるネットワーク（インターネットではなく）で実行されているか、iptables コマンドラインツールを使用してカスタムファイアウォールを設定する必要がある場合にのみ選択する必要があります。

警告

ファイアウォール設定とカスタマイズされたファイアウォールルールは、/etc/sysconfig/iptables ファイルに保存されます。Disabled を選択し OK、クリックすると、これらの設定およびファイアウォールルールが失われます。

- enabled - このオプションは、DNS 返信や DHCP 要求など、アウトバウンド要求に応答しない着信接続を拒否するように設定します。このマシンで実行中のサービスへのアクセスが必要な場合は、特定サービスに対してファイアウォールの通過許可を選択できます。

システムをインターネットに接続しているが、サーバーを実行する予定がない場合は、これが最も安全な選択肢となります。

#### 2.8.2.3. 信頼できるサービス

信頼されたサービス 一覧でオプションを有効にすると、指定したサービスがファイアウォールを通過できるようになります。

**WWW (HTTP)**

HTTP プロトコルは、Web ページを提供するために Apache（およびその他の Web サーバー）によって使用されます。Web サーバーを一般に利用できるようにする場合は、このチェックボックスを選択します。このオプションは、ページをローカルで表示したり、Web ページを開発する際には必要ありません。このサービスでは、httpd パッケージをインストールする必要があります。

WWW(HTTP) を有効にすると、SSL バージョンの HTTP である HTTPS のポートは開かなくなります。このサービスが必要な場合は、Secure WWW(HTTPS) のチェックボックスを選択します。

**FTP**

FTP プロトコルは、ネットワーク上のマシン間でファイル転送に使用されます。FTP サーバーを一般に利用できるようにする場合は、このチェックボックスを選択します。このサービスでは、vsftpd パッケージをインストールする必要があります。

**SSH**

SSH(Secure Shell)は、リモートマシンにログインして実行するツールスイートです。SSH 経由でマシンにリモートアクセスできるようにするには、このチェックボックスを選択します。このサービスでは、openssh-server パッケージをインストールする必要があります。

**Telnet**

Telnet は、リモートマシンにログインするためのプロトコルです。Telnet 通信は暗号化されず、ネットワークスヌーピングのセキュリティーを提供しません。着信 Telnet アクセスを許可することは推奨されません。telnet 経由でマシンにリモートアクセスできるようにするには、このチェックボックスを選択します。このサービスでは、telnet-server パッケージをインストールする必要があります。

**Mail(SMTP)**

SMTP は、リモートホストを直接マシンに接続してメールを配信できるようにするプロトコルです。POP3 または IMAP を使用して、またはなどのツールを使用している場合は、このサービスを有効にする必要はありません fetchmail。お使いのマシンへのメールの配信を許可するには、このチェックボックスを選択します。適切に設定された SMTP サーバーは、リモートマシンがサーバーを使用してスパムを送信することができることに注意してください。

**NFS4**

ネットワークファイルシステム(NFS)は、*NIX システムで一般的に使用されるファイル共有プロトコルです。このプロトコルのバージョン 4 は以前のプロトコルよりも安全です。システムの

ファイルまたはディレクトリーを他のネットワークユーザーと共有する場合は、このチェックボックスを選択します。

**Samba**

Samba は、Microsoft のプロプライエタリー SMB ネットワークプロトコルの実装です。ファイル、ディレクトリー、またはローカルに接続したプリンターを Microsoft Windows マシンと共有する必要がある場合は、このチェックボックスを選択します。

#### 2.8.2.4. その他のポート

ファイアウォール設定ツール には、カスタム IP ポートをが信頼済みとして指定するその他の ポートセクションが含まれ iptablesます。たとえば、IRC およびインターネットプリントプロトコル(IPP) がファイアウォールを通過できるようにするには、その他のポート セクションに以下を追加します。

```
194:tcp,631:tcp
```

#### 2.8.2.5. 設定の保存

OK をクリックして変更を保存し、ファイアウォールを有効または無効にします。Enable firewall を 選択すると、選択したオプションが iptables コマンドに変換され、/etc/sysconfig/iptables ファイルに書き込まれます。また、選択したオプションを保存した直後にファイアウォールがアクティブになるように、iptables サービスが起動します。ファイアウォールを無効 にすると、/etc/sysconfig/iptables ファイルが削除され、iptables サービスがすぐに停止されます。

選択したオプションは /etc/sysconfig/system-config-firewall ファイルに書き込まれるため、アプリケーションの次回起動時に設定を復元できます。このファイルは手動で編集しないでください。

ファイアウォールがすぐにアクティブになっている場合でも、iptables サービスはシステムの起動時に自動的に起動するように設定されません。詳細は「IPTables サービスのアクティブ化」を参照してください。

#### 2.8.2.6. IPTables サービスのアクティブ化

ファイアウォールルールは、iptables サービスが実行している場合にのみアクティブになります。サービスを手動で起動するには、root で以下のコマンドを使用します。

```
~]# service iptables restart
iptables: Applying firewall rules:                         [  OK  ]
```

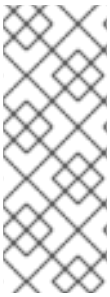
システムの起動 iptables 時にが起動するようにするには、以下のコマンドを使用します。

```
~]# chkconfig --level 345 iptables on
```

### 2.8.3. IPTables の使用

に使用する最初の手順は、iptables サービスを起動すること iptables です。root ユーザーで次のコマンドを実行して、iptables サービスを起動します。

```
~]# service iptables restart
iptables: Applying firewall rules:                         [  OK  ]
```



**注記**

ip6tables サービスのみを使用する場合は、iptables サービスをオフにできます。ip6tables サービスを非アクティブにする場合は、必ず IPv6 ネットワークを非アクティブにします。一致するファイアウォールなしでネットワークデバイスをアクティブのままにしないでください。

システムの起動時 iptables にデフォルトで起動するように強制するには、root で次のコマンドを実行します。

```
~]# chkconfig --level 345 iptables on
```

これにより iptables、システムがランレベル 3、4、または 5 で起動するたびに起動します。

#### 2.8.3.1. iptables コマンドの構文

以下に示す iptables コマンド例は、基本的なコマンド構文を示しています。

```
iptables -A <chain> -j <target>
```

-A オプションは、ルールを <chain> に追加することを指定し *ます*。各チェーンは、1 つ以上の *ルールで構成され、ルール* セットとも呼ばれ *ます*。

この 3 つの組み込みチェーンは INPUT、OUTPUT、および FORWARD です。これらのチェーンは永続的であるため、削除できません。チェーンは、パケットを操作するポイントを指定します。

**-j *<target>*** **オプションは、ルールのターゲットを指定します。つまり、パケットがルールにマッチした場合に実行する動作です。ビルトインターゲットの例は ACCEPT、DROP、および REJECT です。**

**利用可能なチェーン、オプション、およびターゲットの詳細は、iptables man ページを参照してください。**

#### 2.8.3.2. 基本的なファイアウォールポリシー

**基本的なファイアウォールポリシーを確立すると、より詳細なユーザー定義ルールを構築するための基盤が作成されます。**

**各 iptables チェーンはデフォルトのポリシーで構成されており、ファイアウォールの全体的なルールセットを定義するデフォルトポリシーと動作するゼロ以上のルールで構成されます。**

**チェーンのデフォルトポリシーは DROP または ACCEPT のどちらかです。セキュリティー関連管理者は通常、DROP のデフォルトポリシーを実装し、ケースごとに特定のパケットのみを許可します。たとえば、以下のポリシーにより、ネットワークゲートウェイ上の送受信パケットがすべてブロックされます。**

```
~]# iptables -P INPUT DROP
~]# iptables -P OUTPUT DROP
```

**また、ファイアウォールから移行先ノードへルーティング *されるネットワークトラフィック（拒否されるパケット*）も推奨され、内部クライアントがインターネットへの不正な公開を防ぎます。これを行うには、以下のルールを使用します。**

```
~]# iptables -P FORWARD DROP
```

**各チェーンのデフォルトポリシーを確立したら、特定のネットワークおよびセキュリティー要件に対して追加のルールを作成して保存することができます。**

**以下のセクションでは、iptables ルールを保存する方法と、iptables ファイアウォールをビルドする際に実装するルールの一部を概説します。**

#### 2.8.3.3. IPTables ルールの保存および復元

への変更 iptables が推移的です。システムが再起動したり、iptables サービスが再起動すると、ルールは自動的にフラッシュおよびリセットされます。ルールを保存し、iptables サービスの起動時に読み込まれるようにルールを保存するには、root ユーザーとして以下のコマンドを実行します。

```
~]# service iptables save
iptables: Saving firewall rules to /etc/sysconfig/iptables:[  OK  ]
```

ルールはファイルに保存され /etc/sysconfig/iptables、サービスが起動またはマシンが再起動されるたびに適用されます。

### 2.8.4. 一般的な IPTables フィルター

リモートの攻撃者が LAN にアクセスできないことは、ネットワークセキュリティーの最も重要な側面の 1 つです。LAN の整合性は、厳格なファイアウォールルールを使用して、悪意のあるリモートユーザーから保護する必要があります。

ただし、デフォルトのポリシーでは、すべての着信パケット、送信パケット、および転送されたパケットをブロックするように設定すると、ファイアウォール/ゲートウェイおよび内部 LAN ユーザーが相互に通信したり、外部のリソースと通信したりすることはできません。

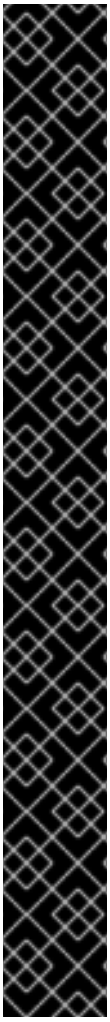
ユーザーがネットワーク関連の機能を実行したり、ネットワークアプリケーションを使用できるようにするために、管理者は通信のために特定のポートを開く必要があります。

たとえば、ファイアウォールのポート 80 へのアクセスを許可するに は、以下のルールを追加します。

```
~]# iptables -A INPUT -p tcp -m tcp --dport 80 -j ACCEPT
```

これにより、ユーザーは標準のポート 80 を使用して通信する Web サイトを閲覧できます。Web サイト（例： https://www.example.com/）へのアクセスを許可するには、以下のように 443 ポートへのアクセスも提供する必要があります。

```
~]# iptables -A INPUT -p tcp -m tcp --dport 443 -j ACCEPT
```



重要

iptables ruleset を作成する際には、順番が重要です。

ルールで 192.168.100.0/24 サブネットからのパケットがドロップされることを指定し、その後に 192.168.100.13 からのパケットを許可するルール（ドロップされたサブネット内にある）を許可するルールが無視されます。

192.168.100.13 からのパケットを許可するルールは、残りのサブネットをドロップするルールの前に付ける必要があります。

既存のチェーン内の特定の場所にルールを追加するには、-I オプションを使用します。以下に例を示します。

```
~]# iptables -I INPUT 1 -i lo -p all -j ACCEPT
```

このルールは、INPUT チェーンの最初のルールとして挿入され、ローカルループバックデバイスのトラフィックを許可します。

LAN へのリモートアクセスが必要な場合があります。SSH などのセキュアなサービスは、LAN サービスへの暗号化されたリモート接続に使用できます。

PPP ベースのリソース（連邦アカウントや一括アカウントなど）を持つ管理者は、ネットワークアクセスを使用してファイアウォールバリアを安全に回避できます。直接接続であるため、通常はファイアウォール/ゲートウェイの背後にあります。

ただし、ブロードバンド接続のあるリモートユーザーの場合は、特別なケースを作成することもできます。リモート SSH クライアントからの接続 iptables を受け入れるようにを設定できます。たとえば、以下のルールを使用すると、リモート SSH アクセスが可能になります。

```
~]# iptables -A INPUT -p tcp --dport 22 -j ACCEPT
~]# iptables -A OUTPUT -p tcp --sport 22 -j ACCEPT
```

このルールにより、インターネットに直接接続された 1 つの PC やファイアウォール/ゲートウェイなど、個々のシステムに着信およびアウトバウンドアクセスが可能になります。ただし、ファイアウォール/ゲートウェイの背後にあるノードがこのようなサービスにアクセスするのを許可しません。こ

れらのサービスへの LAN アクセスを許可するには、*ネットワークアドレス変換*（）を使用します。NAT）は、iptables フィルタールールに置き換えます。

### 2.8.5. FORWARD および NAT ルール

大半のリリースでは、それらが提供する組織に対して、公開されたルーティング可能な IP アドレスの数だけが提供されます。

このため、管理者は LAN 上のすべてのノードにパブリック IP アドレスを付与せずにインターネットサービスへのアクセスを共有する代替方法を見つける必要があります。プライベート IP アドレスの使用は、LAN 上のすべてのノードが内部ネットワークサービスおよび外部ネットワークサービスを適切にアクセスできるようにする最も一般的な方法です。

エッジルーター（ファイアウォールなど）はインターネットから受信送信を受信し、パケットを目的の LAN ノードにルーティングすることができます。同時に、ファイアウォール/ゲートウェイは、LAN ノードからリモートインターネットサービスに発信要求をルーティングすることもできます。

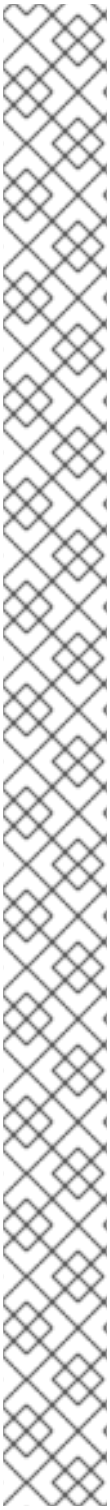
ネットワークトラフィックのこの転送は、特に 内部 IP アドレスを偽装し、LAN 上のノードとして動作させることができる最新のクラッキングツールの可用性で危険にさらされる可能性があります。

これを防ぐために、ネットワークリソースの異常な使用を防ぐために実装できるルーティングおよび転送ポリシーを iptables 提供します。

FORWARD チェーンを使用すると、管理者は LAN 内でパケットをルーティングできる場所を制御できます。たとえば、LAN 全体の転送を許可するには、（ファイアウォール/ゲートウェイに eth1 の内部 IP アドレスが割り当てられていることを想定）、以下のルールを使用します。

```
~]# iptables -A FORWARD -i eth1 -j ACCEPT
~]# iptables -A FORWARD -o eth1 -j ACCEPT
```

このルールにより、ファイアウォール/ゲートウェイの背後で内部ネットワークへのアクセスが可能になります。ゲートウェイは、1 つの LAN ノードから目的のノードにパケットをルーティングし、その eth1 デバイス経由ですべてのパケットを渡します。



**注記**

**デフォルトでは、Red Hat Enterprise Linux カーネルの IPv4 ポリシーは、IP 転送のサポートを無効にします。これにより、Red Hat Enterprise Linux を実行するマシンが専用のエッジルーターとして機能しなくなります。IP 転送を有効にするには、root で以下のコマンドを使用します。**

```
~]# sysctl -w net.ipv4.ip_forward=1
net.ipv4.ip_forward = 1
```

**この設定は現行セッションでのみ有効で、再起動またはネットワークサービスの再起動後も維持されません。IP 転送を永続的に設定するには、以下のように /etc/sysctl.conf ファイルを編集します。**

**以下の行を見つけます。**

```
net.ipv4.ip_forward = 0
```

**以下のように読み込むように編集します。**

```
net.ipv4.ip_forward = 1
```

**root ユーザーとして以下のコマンドを実行し、sysctl.conf ファイルへの変更を有効にします。**

```
~]# sysctl -p /etc/sysctl.conf
net.ipv4.ip_forward = 1
net.ipv4.conf.default.rp_filter = 1
net.ipv4.conf.default.accept_source_route = 0
[output truncated]
```

#### 2.8.5.1. POSTROUTING および IP マスカレード

**ファイアウォールの内部 IP デバイスを介して転送されたパケットを受け入れると、LAN ノードが相互に通信できるようになりますが、インターネット上では外部と通信することができません。**

**プライベート IP アドレスを持つ LAN ノードが外部のパブリックネットワークと通信できるようにするには、*IP マスカレード*のファイアウォールを設定します。これは、LAN ノードからの要求を、ファイアウォールの外部デバイスの IP アドレス（この場合は eth0）でマスクします。**

```
~]# iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE
```

**このルールは NAT パケットマッチングテーブル(-t nat)を使用して、ファイアウォールの外部ネットワークデバイス(-A POSTROUTING)上の NAT（）用のビルトイン POSTROUTING チェーンを指定し-o eth0ます。**

**POSTROUTING により、ファイアウォールの外部デバイスを残す際にパケットを変更できます。**

**-j MASQUERADE ターゲットは、ノードのプライベート IP アドレスをファイアウォール/ゲートウェイの外部 IP アドレスでマスクするように指定します。**

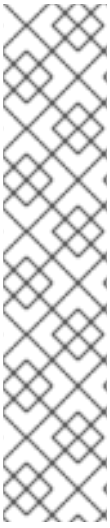
#### 2.8.5.2. PREROUTING

**内部ネットワークにサーバーを外部で使用できるようにするには、NAT の PREROUTING チェーンの -j DNAT ターゲットを使用して、内部サービスへの接続を要求する着信パケットが転送先 IP アドレスとポートを指定できます。**

**たとえば、受信 HTTP 要求を 172.31.0.23 で専用の Apache HTTP Server に転送する場合は、root ユーザーとして以下のコマンドを使用します。**

```
~]# iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 80 -j DNAT --to 172.31.0.23:80
```

**このルールは、を指定します。 NAT テーブルは、組み込みの PREROUTING チェーンを使用して、受信 HTTP 要求を一覧表示された宛先 IP アドレス 172.31.0.23 のみに転送します。**



**注記**

**FORWARD チェーンにデフォルトの DROP ポリシーがある場合は、受信するすべての HTTP 要求を転送するルールを追加して、宛先 NAT ルーティングができるようにする必要があります。これを行うには、root ユーザーで次のコマンドを実行します。**

```
~]# iptables -A FORWARD -i eth0 -p tcp --dport 80 -d 172.31.0.23 -j ACCEPT
```

**このルールは、ファイアウォールから目的の宛先（ファイアウォールの背後の Apache HTTP サーバー）に受信 HTTP リクエストをすべて転送します。**

#### 2.8.5.3. DMZ および IPTables

専用の HTTP サーバーや FTP サーバーなどの特定のマシンにトラフィックをルーティングする iptables ルールを Demilitarized *zone*（）に作成できます。DMZ).A DMZ は、インターネットなどのパブリックペイロードのサービスを提供する専用の特別なローカルサブネットワークです。

たとえば、受信 HTTP 要求を 10.0.4.2（LAN の 192.168.1.0/24 の範囲外）にルーティングするルールを設定するには、NAT が PREROUTING テーブルを使用して適切な宛先にパケットを転送します。

```
~]# iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 80 -j DNAT \
  --to-destination 10.0.4.2:80
```

このコマンドでは、LAN の外部からポート 80 への HTTP 接続はすべて、その他の内部ネットワークとは別のネットワークにある HTTP サーバーにルーティングされます。このようなネットワークのセグメンテーションは、ネットワーク上のマシンへの HTTP 接続を許可するよりも安全であることを証明できます。

HTTP サーバーがセキュアな接続を受け入れるように設定されている場合、ポート 443 も転送する必要があります。

### 2.8.6. 悪意のあるソフトウェアおよびなりすましの IP アドレス

LAN 内の特定のサブネットまたは特定のノードへのアクセスを制御する、より詳細なルールを作成できます。トリックの木車、写真、その他のクライアント/サーバーのウイルスなど、特定の永続アプリケーションやプログラムをサーバーへの接続に制限することもできます。

たとえば、31337 から 31340 までのポート上のサービスに対するネットワークをスキャンするもの（クラッキング用語で elite ポートと呼ばれます）。

これらの標準ポートを介して通信する正当なサービスはありません。ブロックすると、ネットワーク上のノードがリモートマスターサーバーと個別に通信する可能性が低減される可能性があります。

以下のルールは、ポート 31337 の使用を試みる TCP トラフィックをすべて破棄します。

```
~]# iptables -A OUTPUT -o eth0 -p tcp --dport 31337 --sport 31337 -j DROP
~]# iptables -A FORWARD -o eth0 -p tcp --dport 31337 --sport 31337 -j DROP
```

また、プライベート IP アドレス範囲の偽装を試みる外部接続をブロックして、LAN に侵入することもできます。

たとえば、LAN が 192.168.1.0/24 の範囲を使用している場合は、インターネット向けネットワークデバイス（例： eth0）に指示するルールを設計し、LAN IP 範囲内のアドレスを持つそのデバイスへのパケットを破棄することができます。

転送されたパケットをデフォルトポリシーとして拒否することが推奨されます。そのため、外部向けデバイス(eth0)へのその他の偽装 IP アドレスは自動的に拒否されます。

```
~]# iptables -A FORWARD -s 192.168.1.0/24 -i eth0 -j DROP
```

注記

追加 したルールを処理する場合は DROP、と REJECT ターゲット間に区別があります。

REJECT ターゲットはアクセスを拒否し、サービスへの接続を試みるユーザーに connection refused エラーを返します。DROP ターゲットは、名前が示すように、警告なしでパケットをドロップします。

管理者は、これらのターゲットを使用する際に独自の判断を使用できます。

### 2.8.7. iptables と接続追跡

接続 *状態に基づいて、サービスへの接続を検査および制限できます。*モジュールは、*接続追跡* と呼ばれるメソッドを iptables 使用して受信接続に関する情報を保存します。以下の接続状態に基づいてアクセスを許可または拒否できます。

- NEW : HTTP リクエストなどの新しい接続を要求するパケット。
- ESTABLISHED : 既存の接続の一部であるパケット。
- RELATED : 新しい接続を要求しているが、既存の接続の一部であるパケット。たとえば、FTP はポート 21 を使用して接続を確立しますが、データは別のポート（通常はポート 20）で転送されます。
- INVALID : 接続追跡テーブル内の接続の一部ではないパケット。

プロトコル自体がステートレスである場合でも（UDP など）、すべてのネットワークプロトコルで iptables 接続追跡のステートフル機能を使用できます。以下の例は、接続追跡を使用して、確立された接続に関連するパケットのみを転送するルールを示しています。

```
~]# iptables -A FORWARD -m state --state ESTABLISHED,RELATED -j ACCEPT
```

#### 2.8.8. IPv6

IPv6 と呼ばれる次世代のインターネットプロトコルの導入は、IPv4（または IP）の 32 ビットアドレス制限を超えて拡張されます。IPv6 は 128 ビットのアドレスをサポートし、IPv6 対応している伝送ネットワークは、IPv4 よりも多くのルーティング可能なアドレスに対応することができます。

Red Hat Enterprise Linux は、Netfilter 6 サブシステムと ip6tables コマンドを使用して IPv6 ファイアウォールルールをサポートします。Red Hat Enterprise Linux 6 では、IPv4 サービスおよび IPv6 サービスの両方がデフォルトで有効になっています。

ip6tables コマンド構文は、128 ビットアドレス iptables をサポートする点を除き、すべての側面と同じです。たとえば、以下のコマンドを使用して IPv6 対応ネットワークサーバーで SSH 接続を有効にします。

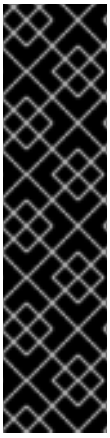
```
~]# ip6tables -A INPUT -i eth0 -p tcp -s 3ffe:ffff:100::1/128 --dport 22 -j ACCEPT
```

IPv6 ネットワークの詳細は、http://www.ipv6.org/ の IPv6 情報ページを参照して ください。

#### 2.8.9. iptables

Red Hat Enterprise Linux に含まれるのは、*ネットワークパケットフィルタリングの高度なツールです。これは、カーネル内でネットワークスタックを入力、移動、および終了する際にネットワークパケット* を制御するプロセスです。2.4 よりも前のバージョンのカーネルは、パケットフィルタリングに依存し、フィルタリングプロセスの各ステップで ipchains パケットに適用されるルールの一覧を使用していました。2.4 カーネルが導入された（ *netfilter*と iptables も呼ばれます）。これはと似ています ipchains が、ネットワークパケットのフィルタリングに使用できる範囲と制御が大きくなります。

本章では、パケットフィルタリングの基本を説明します。また、iptables コマンドで利用可能なさまざまなオプションを説明します。また、システムの再起動間でルールをどのように保持できるかについて説明します。



重要

2.4 以降のカーネルにおけるデフォルトのファイアウォールメカニズムはですが iptables、がすでに実行している場合 ipchains は使用 iptables できません。システムの起動時に ipchains が存在する場合、カーネルはエラーを発行し、起動に失敗し iptablesます。

の機能はこれらのエラーの影響を受け ipchains ません。

#### 2.8.9.1. パケットのフィルタリング

Linux カーネルは Netfilter 機能を使用してパケットをフィルターし、それらの一部はシステム上で受信または通過でき、その他の停止中にシステムを通過できます。この機能は Linux カーネルに構築され、以下のように 5 つの組み込み *テーブル* または *ルール一覧* を持ちます。

- filter : ネットワークパケットを処理するデフォルトの表。
- nat : 新しい接続を作成し、 *ネットワークアドレス変換* (*NAT*)に使用するパケットを変更するために使用されます。
- mangle : 特定のタイプのパケット変更に使用されます。
- raw : 主に NOTRACK ターゲットと組み合わせて接続追跡からの除外を設定するために使用されます。
- security : SECMARK および CONNSECMARK ターゲットにより有効になっているなど、MAC(Mandatory Access Control)ネットワークルールに使用されます。

各テーブルには組み込みチェーンのグループがあり 、 これはでパケットで実行されるアクションに対応し netfilterます。

filter テーブルの組み込みチェーンは以下のとおりです。

- *INPUT* - ホスト用のターゲットとなるネットワークパケットに適用されます。

- ***OUTPUT*: ローカルに生成されるネットワークパケットに適用されます。**
- ***FORWARD*: ホスト経由でルーティングされるネットワークパケットに適用されます。**

**nat テーブルの組み込みチェーンは以下のとおりです。**

- ***PREROUTING*: 受信時にネットワークパケットに適用されます。**
- ***OUTPUT*: ローカルに生成されるネットワークパケットの送信前に適用されます。**
- ***POSTROUTING* - 送信前にネットワークパケットに適用されます。**

**mangle テーブルの組み込みチェーンは以下のとおりです。**

- ***INPUT* - ホストターゲットのネットワークパケットに適用されます。**
- ***OUTPUT*: ローカルに生成されるネットワークパケットの送信前に適用されます。**
- ***FORWARD*: ホスト経由でルーティングされるネットワークパケットに適用されます。**
- ***PREROUTING*: ルーティングの前に着信ネットワークパケットを適用します。**
- ***POSTROUTING* - 送信前にネットワークパケットに適用されます。**

**raw テーブルの組み込みチェーンは以下のとおりです。**

- ***OUTPUT*: ローカルに生成されるネットワークパケットの送信前に適用されます。**

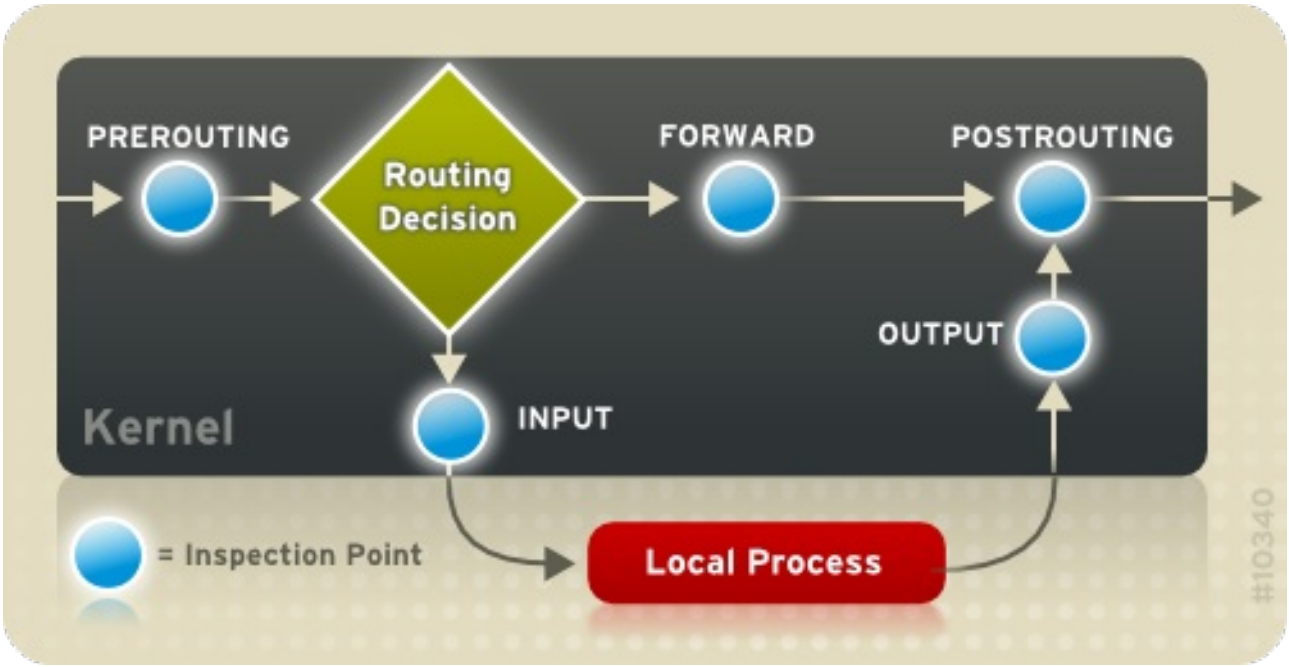
- *PREROUTING*: ルーティングの前に着信ネットワークパケットを適用します。

security テーブルの組み込みチェーンは以下のとおりです。

- *INPUT* - ホストターゲットのネットワークパケットに適用されます。
- *OUTPUT*: ローカルに生成されるネットワークパケットの送信前に適用されます。
- *FORWARD*: ホスト経由でルーティングされるネットワークパケットに適用されます。

Linux システムから送受信したネットワークパケットはすべて、少なくとも 1 つのテーブルを持ちます。ただし、パケットは、チェーンの最後で変わる前に、各テーブル内の複数のルールが適用される場合があります。これらのルールの構造と目的は異なる場合がありますが、特定のプロトコルとネットワークサービスを使用する場合に、通常は特定の IP アドレスまたはアドレスのセットから送信されるパケットの特定を試みます。以下の図は、iptables サブシステムによりパケットのフローがどのように調査されるかを示しています。

図2.6 IPTable でのパケットフィルタリング




[D]

重要

デフォルトでは、ファイアウォールルールは /etc/sysconfig/iptables または /etc/sysconfig/ip6tables ファイルに保存されます。

この iptables サービスは、Linux システムが起動したときに DNS 関連のサービスの前に起動します。これは、ファイアウォールルールが数値の IP アドレスのみを参照できることを意味します（例： 192.168.0.1）。このようなルールのドメイン名（例： host.example.com）によりエラーが生じます。

あるテーブルでパケットが特定のルールと一致する場合、宛先に関係なく *ターゲット* またはアクションが適用されます。ルールが一致するパケットの ACCEPT ターゲットを指定する場合、パケットは残りのルールチェックを省略し、宛先の継続が許可されます。ルールが DROP ターゲットを指定した場合、そのパケットはシステムへのアクセスを拒否し、パケットを送信したホストに何も送信されません。ルールが QUEUE ターゲットを指定する場合、パケットはユーザー空間に渡されます。ルールがオプションの REJECT ターゲットを指定する場合、パケットは破棄されますが、エラーパケットはパケットのオリジンに送信されます。

すべてのチェーンには ACCEPT、、、DROP REJECT、またはのデフォルトポリシーがありQUEUEます。チェーン内のルールがパケットに適用されない場合、パケットはデフォルトポリシーに従って処理されます。

iptables コマンドはこれらのテーブルを設定し、必要に応じて新しいテーブルを設定します。

注記

netfilter モジュールは、デフォルトではロードされません。したがって、すでに使用されているか、または読み込み済みのものだけを示すため、ユーザーは /proc/ ディレクトリーを検索してそれらのすべてを表示しません。つまり、netfilter の機能の使用前に利用可能な機能を確認することはできません。

#### 2.8.9.2. IPTables のコマンドオプション

パケットをフィルタリングするルールは、iptables コマンドを使用して作成されます。パケットの以下の側面は、基準として最もよく使用されます。

- packet Type: コマンドフィルターのパケットのタイプを指定します。
-

パケットソース/宛先: パケットの送信元または宛先に基づいて、コマンドがフィルターするパケットを指定します。

- target - 上記の基準に一致するパケットに対して実行するアクションを指定します。

パケットのこのような側面に対応する特定のオプション「ターゲットオプション」 については「iptables マッチングオプション」、およびを参照してください。

特定のルールで使用するオプションは、iptables ルールが有効になるように、全体的なルールの目的と条件に基づいて論理的にグループ化する必要があります。本セクションの残りの部分では、iptables コマンドに一般的に使用されるオプションを説明します。

#### 2.8.9.2.1. IPTables コマンドオプションの構造

多くの iptables コマンドの構造は次のとおりです。

```
iptables [-t <table-name>] <command> <chain-name> \
  <parameter-1> <option-1> \
  <parameter-n> <option-n>
```

***<table-name>***: ルールが適用されるテーブルを指定します。省略すると、filter テーブルが使用されます。

***<command>***: ルールの追加や削除など、実行するアクションを指定します。

***<chain-name>***: 編集、作成、または削除を行うチェーンを指定します。

***<parameter>-<option>*** ペア： ルールに一致するパケットを処理する方法を指定するパラメーターおよび関連オプション。

iptables コマンドの長さと複雑性はその目的に基づいて大幅に変更される可能性があります。

たとえば、チェーンからルールを削除するコマンドは、非常に短い場合があります。

```
iptables -D <chain-name> <line-number>
```

-

一方、さまざまな特定のパラメーターやオプションを使用して特定のサブネットからパケットをフィルターするルールを追加するコマンドは、実際には長い場合があります。iptables コマンドを構築する際には、一部のパラメーターおよびオプションには、有効なルールを作成するために追加のパラメーターとオプションが必要である点を念頭に置いてください。これにより、cascading 効果が生成され、追加のパラメーターが必要となりますが、他のパラメーターは必須となります。別のオプションセットを必要とするパラメーターおよびオプションがすべて満たされるまで、ルールは有効ではありません。

iptables コマンド構造の包括的な一覧 iptables -h を表示するには、と入力してください。

#### 2.8.9.2.2. コマンドオプション

特定のアクション iptables を実行するように指示するコマンドオプション。1 つのコマンドにつき 1 つのコマンドオプションのみが許可され iptables ます。help コマンドを除くと、すべてのコマンドは大文字で記述されます。

iptables コマンドのオプションは以下のとおりです。

- -A : 指定したチェーンの最後にルールを追加します。以下の -I オプションとは異なり、整数引数を取りません。指定したチェーンの最後にルールを常に追加します。

- -D <integer> | <rule> : 特定のチェーン内のルールを数字で削除します（チェーン内 5 の 5 番目のルールなど）、またはルールの指定により削除します。ルール仕様は、既存のルールと完全に一致している必要があります。

- -E : ユーザー定義チェーンの名前を変更します。ユーザー定義チェーンは、デフォルトの既存チェーン以外のチェーンです。（ユーザー定義チェーンの作成の詳細については、以下の -N オプションを参照してください。） これはメカニズム的な変更で、テーブルの構造には影響を与えません。

  注記

  デフォルトチェーンのいずれかの名前を変更しようとすると、システムは Match not found エラーを報告します。デフォルトのチェーンの名前を変更することはできません。

- -F : 選択したチェーンをフラッシュし、チェーンのすべてのルールを効果的に削除します。chain が指定されていない場合、このコマンドはすべてのチェーンからすべてのルールを

フラッシュします。

- -h : コマンド構造の一覧と、コマンドパラメーターおよびオプションの簡単なサマリーを提供します。

- -I [<integer>] : ユーザー定義の整数引数で指定されたポイントに、指定されたチェーンにルールを挿入します。引数を指定しないと、ルールはチェーンの上部に挿入されます。

重要

上記のように、チェーン内のルールの順序は、どのルールを適用するかを決定します。これは、-A または -I オプションのいずれかを使用してルールを追加する際に注意することが重要です。

これは -I、整数引数を使用してルールを追加する場合に特に重要になります。チェーンにルールを追加するときに既存の数字を指定すると、既存のルールの 前 に新しいルールを iptables 追加します（またはそれ以上）。

- -L : コマンドの後に指定したチェーンのすべてのルールを一覧表示します。デフォルト filter テーブルのすべてのチェーンに含まれるすべてのルールを一覧表示するには、チェーンまたはテーブルを指定しないでください。それ以外の場合は、以下の構文を使用して、特定の表内の特定のチェーンのルールを一覧表示する必要があります。

```
iptables -L <chain-name> -t <table-name>
```

ルール番号を提供し、さらに詳細なルールの説明を許可する -L コマンドオプションの追加オプションについては、で説明し 「オプションの一覧表示」 ます。

- -N : ユーザーが指定した名前で新しいチェーンを作成します。チェーン名は一意である必要があります。一意でなければエラーメッセージが表示されます。

- -P : 指定したチェーンのデフォルトポリシーを設定し、パケットがルールにマッチせずにチェーン全体を通過する場合は、ACCEPT や DROP などの指定されたターゲットに送信されます。

- -R : 指定したチェーンのルールを置き換えます。ルールの数は、チェーン名の後に指定する必要があります。チェーンの最初のルールは、ルール番号 1 に対応します。

- **-X : ユーザー指定のチェーンを削除します。組み込みチェーンは削除できません。**

- **-Z : テーブルのすべてのチェーンのバイトカウンターとパケットカウンターをゼロに設定します。**

#### 2.8.9.2.3. iptables パラメーターオプション

**特定のチェーン内でルールを追加、追加、削除、挿入、または置き換えるのに使用する iptables コマンドなど、パケットフィルタリングルールを構成するためにさまざまなパラメーターが必要になります。**

- **-c : 特定のルールのカウンターをリセットします。このパラメーターは PKTS、と BYTES オプションを指定して、リセットするカウンターを指定します。**

- **-d : ルールに一致するパケットの宛先ホスト名、IP アドレス、またはネットワークを設定します。ネットワークを照合する場合、以下の IP アドレス/ネットマスク形式がサポートされます。**

  - ***N.N.N.N*/*M.M.M.M* : *N.N.N.N* は IP アドレス範囲で、 *M.M.M.M* はネットマスクです。**

  - ***N.N.N.N*/*M* : *N.N.N.N* は IP アドレス範囲で、 *M* はビットマスクです。**

- **-f : このルールを適用するのは、断片化されたパケットにのみ適用されます。**

  **このパラメーターの前に感嘆符文字(!)オプションを使用して、アンフラグされたパケットのみが一致するように指定できます。**

注記

フラグメント化されたパケットとフラグメントされていないパケットを区別することは可能ですが、断片化されたパケットは IP プロトコルの標準部分となります。

当初、IP パケットが異なるフレームサイズを持つネットワークを通過できるように設計されており、この日付の断片化は、不適切なパケットを使用して DoS 攻撃を生成するのに一般的に使用されます。また、IPv6 は断片化を完全に拒否することを認識してください。

- -i : は、eth0 やなどの受信ネットワークインターフェースを設定し ppp0ます。では iptables、このオプションのパラメーターを、テーブルとおよび filter テーブルを持つ PREROUTING チェーンと使用する場合のみ INPUT チェーンおよび FORWARD チェーン nat と併用でき mangle ます。

  このパラメーターは、以下の特別なオプションもサポートします。

  - exclamation point(!): ディレクティブを逆にします。つまり、指定されたインターフェースがこのルールから除外されます。

  - プラス文字(+)- 指定した文字列に一致するすべてのインターフェースを照合するために使用されるワイルドカード文字。たとえば、パラメーター -i eth+ は、このルールをイーサネットインターフェースに適用しますが、等の他のインターフェースは除外し ppp0ます。

  -i パラメーターが使用されていてもインターフェースが指定されていない場合、すべてのインターフェースがルールの影響を受けます。

- -j : パケットが特定のルールにマッチすると、指定したターゲットに移動します。

  標準ターゲットは ACCEPT、、DROP QUEUE、およびです RETURN。

  拡張オプションは、Red Hat Enterprise Linux iptables RPM パッケージでデフォルトで読み込まれるモジュールでも利用できます。これらのモジュールの有効なターゲットには LOG

MARK、、REJECT、が含まれます。これらおよびその他のターゲットの詳細は、iptables man ページを参照してください。

このオプションを使用して、パケットに一致するパケットを現在のチェーン外のユーザー定義チェーンに転送し、他のルールをパケットに適用することもできます。

ターゲットを指定しないと、パケットはアクションを実行せずにルールを渡します。ただし、このルールのカウンターは 1 つ増えます。

- -o : ルールの発信ネットワークインターフェースを設定します。このオプションは、filter テーブルの OUTPUT チェーンおよび FORWARD チェーンと、nat および mangle テーブルの POSTROUTING チェーンにのみ有効です。このパラメーターは、受信ネットワークインターフェースパラメーター(-i)と同じオプションを受け入れます。

- -p <protocol> : ルールの影響を受ける IP プロトコルを設定します。これは icmp、、tcp udp、またはのいずれかを使用するか all、またはこれらのいずれかまたは別のプロトコルを表す数値の値になります。/etc/protocols ファイルに記載されているプロトコルを使用することもできます。

「all」プロトコルは、サポートされるすべてのプロトコルにルールが適用されることを意味します。このルールでプロトコルが一覧表示されていない場合、デフォルトはall"" に設定されます。

- -s : destination(-d)パラメーターと同じ構文を使用して、特定のパケットのソースを設定します。

#### 2.8.9.2.4. iptables マッチングオプション

異なるネットワークプロトコルは、そのプロトコルを使用して特定のパケットと一致するように設定できる特殊なマッチングオプションを提供します。ただし、プロトコルは最初に iptables コマンドで指定する必要があります。たとえば、は、指定されたプロトコルのオプションを **-p *<protocol-name>*** 有効にします。プロトコル名の代わりにプロトコル ID を使用することもできます。以下の例を参照してください。各例は、同じ効果を持ちます。

```
~]# iptables -A INPUT -p icmp --icmp-type any -j ACCEPT
~]# iptables -A INPUT -p 5813 --icmp-type any -j ACCEPT
```

サービス定義は /etc/services ファイルで提供されます。読みやすくするため、ポート番号ではなくサービス名を使用することが推奨されます。

**警告**

**/etc/services ファイルを保護し、承認されていない編集を防ぎます。このファイルを編集すると、攻撃者はそれを使用して、他の方法で閉じているマシンのポートを有効にすることができます。このファイルを保護するには、root で以下のコマンドを実行します。**

```
~]# chown root.root /etc/services
~]# chmod 0644 /etc/services
~]# chattr +i /etc/services
```

**これにより、ファイルの名前変更や削除、またはリンクの作成ができなくなります。**

#### 2.8.9.2.4.1. TCP プロトコル

**これらの一致オプションは TCP プロトコル(-p tcp)で利用できます。**

- **--dport : パケットの宛先ポートを設定します。**

  **このオプションを設定するには、ネットワークサービス名（www、smtp など）、ポート番号、またはポート番号を使用します。**

  **ポート番号の範囲を指定するには、2 つの数字をコロン(:)で区切ります。例: -p tcp --dport 3000:3200.許容できる最大有効範囲はです 0:65535。**

  **--dport オプションの後に感嘆符(!)を使用して、ネットワークサービスまたはポートを使用し ない すべてのパケットに一致させます。**

  **ネットワークサービスの名前とエイリアス、およびそれらが使用するポート番号を参照するには、/etc/services ファイルを表示します。**

  **--destination-port match オプションは、と同一のものです --dport。**

- **--sport :** と同じオプションを使用して、パケットのソースポートを設定し **--dport**ます。**--source-port match** オプションは、と同一のものです **--sport**。

- **--syn : 通信を開始するために設計されたすべての TCP パケット（通常は *SYN パケット*）に適用されます。データペイロードを伝送するパケットはいずれも伝送されません。**

  **--syn オプションの前に感嘆符(!)を使用して、すべての SYN パケットに一致させます。**

- **--tcp-flags <tested flag list> <set flag list> : 特定のビット（フラグ）が設定されている TCP パケットをルールに一致させることができます。**

  **--tcp-flags match オプションは、2 つのパラメーターを受け入れます。最初のパラメーターはマスクで、パケットで調べるフラグのコンマ区切りリストです。2 番目のパラメーターは、ルールが一致するように設定する必要のあるフラグのコンマ区切りリストです。**

  **可能なフラグは次のとおりです。**

  - **ACK**
  - **FIN**
  - **PSH**
  - **RST**
  - **SYN**
  - **URG**
  - **ALL**

- NONE

たとえば、以下の仕様が含まれる iptables ルールは、SYN フラグセットを持つ TCP パケットのみと一致し、ACK フラグおよび FIN フラグが設定されていません。

--tcp-flags ACK,FIN,SYN SYN

の後 --tcp-flags に感嘆符(!)を使用して、match オプションの効果を元に戻します。

- --tcp-option : 特定のパケット内で設定できる TCP 固有のオプションとの照合を試みます。この match オプションは、オプションの後に感嘆符(!)を使用して元に戻すこともできます。

#### 2.8.9.2.4.2. UDP プロトコル

UDP プロトコル(-p udp)には、これらの一致オプションを使用することができます。

- --dport : サービス名、ポート番号、またはポート番号の範囲を使用して、UDP パケットの宛先ポートを指定します。--destination-port match オプションは、と同一のものです --dport。
- --sport : サービス名、ポート番号、またはポート番号の範囲を使用して、UDP パケットのソースポートを指定します。--source-port match オプションは、と同一のものです --sport。

--dport および --sport オプションには、ポート番号の範囲を指定するには、2 つの数字をコロン(:)で区切ります。例: -p tcp --dport 3000:3200.許容できる最大有効範囲はです 0:65535。

#### 2.8.9.2.4.3. ICMP プロトコル

以下の match オプションは、Internet Control Message Protocol(ICMP)で利用でき-p icmpます。

- --icmp-type : ルールと一致するように ICMP タイプの名前または数を設定します。iptables -p icmp -h コマンドを入力して、有効な ICMP 名の一覧を取得できます。

#### 2.8.9.2.4.4. 追加の一致オプションモジュール

その他の一致オプションは、iptables コマンドで読み込まれるモジュールで利用できます。

一致するオプションモジュールを使用するには、を使用して名前でモジュールを読み込みます。ここで -m *<module-name>*、*<module-name>* はモジュールの名前になります。

多くのモジュールがデフォルトで利用できます。モジュールを作成して、追加機能を提供することもできます。

以下は、最も一般的に使用されるモジュールの一部です。

- limit module: 特定のルールに一致するパケット数の制限を配置します。

  LOG ターゲットと併用すると、limit モジュールは一致するパケットが繰り返し発生するメッセージでシステムログを埋めたり、システムリソースを使用したりすることを防ぐことができます。

  LOG ターゲット 「ターゲットオプション」 の詳細はを参照してください。

  limit モジュールは、以下のオプションを有効にします。

  - --limit : *<value>*/*<period>* ペアとして指定された特定の期間の最大一致数を設定します。たとえば、を使用 --limit 5/hour すると、1 時間あたり 5 つのルールに一致することができます。

    期間は秒単位、分、時間、または日で指定できます。

    数値および時間修飾子を使用しない場合は、のデフォルト値が想定 3/hour されます。

- --limit-burst : 一度にルールに一致できるパケット数の制限を設定します。

  このオプションは整数として指定されており、--limit オプションとともに使用する必要があります。

  値の指定がない場合は、デフォルト値の 5(5)が想定されます。

- state module: 状態一致を有効にします。

  state モジュールは、以下のオプションを有効にします。

  - --state : 以下の接続状態を持つパケットを照合します。

    - ESTABLISHED : 一致するパケットは、確立された接続内の他のパケットと関連付けられます。クライアントとサーバー間の接続を維持する場合は、この状態を受け入れる必要があります。
    - INVALID : 一致するパケットは既知の接続に関連付けられません。
    - NEW : 一致するパケットは新しい接続を作成するか、以前確認されていない双方向接続の一部です。サービスへの新規接続を許可する場合は、この状態を受け入れる必要があります。
    - RELATED : 一致するパケットは、既存の接続に関連する新しい接続を開始します。たとえば、制御トラフィック（ポート 21）に 1 つの接続を使用する FTP と、データ転送に個別の接続（ポート 20）が使用されています。

    これらの接続状態は、などのコンマで区切ることにより、相互に使用することができ -m state --state INVALID,NEWます。

- mac module - ハードウェアの MAC アドレス一致を有効にします。

mac モジュールは、以下のオプションを有効にします。

- --mac-source : パケットを送信するネットワークインターフェースカードの MAC アドレスを照合します。ルールから MAC アドレスを除外するには、--mac-source match オプションの後に感嘆符(!)を付けます。

モジュールで利用可能な他の一致オプションについては、iptables man ページを参照してください。

#### 2.8.9.2.5. ターゲットオプション

パケットが特定のルールにマッチする場合、ルールはパケットを複数の異なるターゲットに転送し、適切なアクションを決定することができます。各チェーンにはデフォルトのターゲットがあります。これは、そのチェーン上のルールがパケットにマッチする場合や、パケットに一致するルールがターゲットを指定するルールがない場合に使用されます。

以下は標準ターゲットになります。

- *<user-defined-chain>* : テーブル内のユーザー定義チェーン。ユーザー定義チェーン名は一意でなければなりません。このターゲットは、パケットを指定チェーンに渡します。
- ACCEPT : 宛先または別のチェーンへのパケットを許可します。
- DROP : リクエスターに応答せずにパケットをドロップします。パケットを送信したシステムは、失敗について通知されません。
- QUEUE : パケットは、ユーザー空間アプリケーションによって処理するためにキューに置かれます。
- RETURN : 現在のチェーンのルールに対するパケットの確認を停止します。RETURN ターゲットのあるパケットが別のチェーンから呼び出されたチェーン内のルールと一致する場合、そのパケットは最初のチェーンに返され、停止先のルールを確認するようになります。RETURN ルールが組み込みチェーンで使用され、パケットが以前のチェーンに移動できない場合は、現在のチェーンのデフォルトターゲットが使用されます。

さらに、他のターゲットを指定できるようにする拡張機能を使用できます。これらの拡張機能は

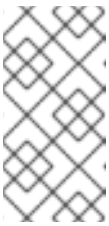
ターゲットモジュールまたは match オプションモジュールと呼ばれ、ほとんどの場合は特定のテーブルおよび状況にのみ適用されます。match オプションモジュール 「追加の一致オプションモジュール」 の詳細は、を参照してください。

多くの拡張ターゲットモジュールが存在し、そのほとんどは特定のテーブルまたは状況にのみ適用されます。Red Hat Enterprise Linux にデフォルトで含まれている最も一般的なターゲットモジュールには、以下のものがあります。

- LOG : このルールに一致するすべてのパケットをログに記録します。パケットはカーネルによりログ記録されるため、/etc/syslog.conf ファイルはこれらのログエントリーが書き込まれる場所を決定します。デフォルトでは、これらは /var/log/messages ファイルに配置されます。

  LOG ターゲットの後に追加オプションを使用して、ロギングが実行される方法を指定できます。

  - --log-level : ロギングイベントの優先度を設定します。優先順位の一覧は、syslog.conf man ページを参照してください。
  - --log-ip-options : IP パケットのヘッダーに設定されたオプションをログに記録します。
  - --log-prefix : 書き込み時にログ行の前に最大 29 文字の文字列を配置します。これは、syslog フィルターを作成してパケットロギングとともに使用する場合に便利です。

    注記

    このオプションの問題により、*log-prefix* の値に末尾のスペースを追加する必要があります。

  - --log-tcp-options : TCP パケットのヘッダーに設定されたオプションをログに記録します。
  - --log-tcp-sequence : ログ内のパケットの TCP シーケンス番号を書き込みます。

- **REJECT : エラーパケットをリモートシステムに戻し、パケットを破棄します。**

  **REJECT ターゲットの受け入れ（ *<type>* --reject-with *<type>* は rejection タイプ）により、エラーパケットで詳細情報が返されます。その他のオプション port-unreachable が使用されていない場合、メッセージはデフォルトのエラータイプです。*<type>* オプションの全一覧は、iptables man ページを参照してください。**

**テーブルを使用した IP マスカレードや、nat テーブルを使用したパケットの変更に役立つ他のターゲット拡張機能は mangle、iptables man ページを参照してください。**

#### 2.8.9.2.6. オプションの一覧表示

**デフォルトの list コマンドは iptables -L [<chain-name>]、デフォルトのフィルターテーブルの現在のチェーンに関する非常に基本的な概要を提供します。詳細は以下を提供します。**

- **-v : 各チェーンが処理したパケット数やバイト数、各ルールにマッチしたパケット数およびバイト数、特定のルールに適用するインターフェースを表示します。**

- **-x : 数字を正確な値に展開します。ビジー状態のシステムでは、特定のチェーンまたはルールによって処理されたパケット数およびバイト数は Kilobytes、、Megabytes、またはに省略され Gigabytesます。このオプションは、フル番号を強制的に表示するようにします。**

- **-n : デフォルトのホスト名およびネットワークサービスの形式ではなく、数字で IP アドレスとポート番号を表示します。**

- **--line-numbers : チェーン内の数値の順序の横にある各チェーンのルールを一覧表示します。このオプションは、チェーンで特定のルールを削除しようとする場合や、チェーン内でルールを挿入する場所を特定する場合に便利です。**

- **-t <table-name> : テーブル名を指定します。省略されている場合、デフォルトは filter テーブルに設定されます。**

#### 2.8.9.3. IPTables ルールの保存

**iptables コマンドで作成したルールはメモリーに保存されます。ルール iptables セットを保存する前にシステムが再起動すると、すべてのルールが失われます。netfilter ルールをシステム再起動後も維**

持するには、保存する必要があります。netfilter ルールを保存するには、root で次のコマンドを実行します。

```
~]# /sbin/service iptables save
iptables: Saving firewall rules to /etc/sysconfig/iptables:[  OK  ]
```

これにより iptables init スクリプトが実行され、/sbin/iptables-save プログラムを実行し、現在の iptables 設定をに書き込み /etc/sysconfig/iptablesます。既存の /etc/sysconfig/iptables ファイルはとして保存され /etc/sysconfig/iptables.saveます。

次にシステムが起動すると、iptables init スクリプトは /sbin/iptables-restore コマンドを使用して保存さ /etc/sysconfig/iptables れたルールを再適用します。

このファイルにコミットする前に新しい iptables ルールをテストすることが推奨されますが、この /etc/sysconfig/iptables ファイルの別のシステムのバージョンから iptables ルールをコピーすることは可能です。これにより、複数のマシンに iptables ルールのセットを簡単に分散できます。

ディストリビューション、バックアップなどの目的で、iptables ルールを別のファイルに保存することもできます。これを行うには、root で以下のコマンドを実行します。

```
iptables-save > <filename>
```

ここでの *<filename>* は、ルールセットのユーザー定義の名前です。

重要

/etc/sysconfig/iptables ファイルを他のマシンに分散する場合は、/sbin/service iptables reload または新しいルールを有効にする /sbin/service iptables restart ためにまたはを入力します。ファイアウォールが配置されていない期間は存在しないため、この reload コマンドを使用することが推奨されます。の reload コマンドの説明を参照してください 「iptables の制御スクリプト」。IPv6 ip6tables の場合は、本セクション iptables に一覧表示され /sbin/service ているコマンドのに置き換えます。IPv6 および netfilter の詳細は、を参照してください 「iptables および IPv6」。

注記

iptables 機能を構成するテーブルおよびチェーンの操作に使用される iptables コマンド (/sbin/iptables)と、サービス iptables 自体を有効または無効にするために使用される iptables サービス (/sbin/service iptables)の違いに留意してください。

#### 2.8.9.4. iptables の制御スクリプト

Red Hat Enterprise Linux では、以下の 2 つの基本的な方法 iptables で制御できます。

- Firewall Configuration Tool (system-config-firewall)- 基本的なファイアウォールルールを作成、アクティブ化、および保存するためのグラフィカルインターフェースです。詳細は「ファイアウォールの基本設定」を参照してください。

- /sbin/service iptables *<option>* : init スクリプトを使用するさまざまな機能を操作するために iptables 使用されます。以下のタイプが使用できます。

  - start : ファイアウォールが設定されている（ /etc/sysconfig/iptables 存在する場合）、実行中のはすべて完全に停止 iptables され、/sbin/iptables-restore コマンドを使用して開始します。このオプションは、ipchains カーネルモジュールが読み込まれて いない場合にのみ機能します。このモジュールが読み込まれているかどうかを確認するには、root で次のコマンドを実行します。

    ```
    ~]# lsmod | grep ipchains
    ```

    このコマンドによって出力が返されない場合、モジュールが読み込まれていないことを意味します。必要に応じて、/sbin/rmmod コマンドを使用してモジュールを削除します。

  - stop - ファイアウォールを実行している場合は、メモリー内のファイアウォールルールがフラッシュされ、すべての iptables モジュールとヘルパーがアンロードされます。

    /etc/sysconfig/iptables-config 設定ファイルの IPTABLES_SAVE_ON_STOP ディレクティブがデフォルト値からに変更される /etc/sysconfig/iptables と yes、現在のルールはに保存され、既存のルールはファイルに移動し /etc/sysconfig/iptables.saveます。

    iptables-config ファイル 「iptables の制御スクリプト設定ファイル」 の詳細は、を参照してください。

  - reload : ファイアウォールを実行している場合は、設定ファイルからファイアウォールルールがリロードされます。この reload コマンドは、以前使用されていたヘルパーをアンロードしませんが、( IPv4)および IP6TABLES_MODULES(IPv6)の場合は、IPTABLES_MODULES(IPv4)に追加されている新しいヘルパーを追加し ます。現在のファ

イアウォールルールをフラッシュしない利点は、ルールにエラーがあるため、新しいルールを適用できない場合は、古いルールがまだあることです。

- restart - ファイアウォールを実行している場合は、メモリー内のファイアウォールルールがフラッシュされ、でファイアウォールが設定されている場合はファイアウォールが再び起動し /etc/sysconfig/iptablesます。このオプションは、ipchains カーネルモジュールが読み込まれていない場合にのみ機能します。

  /etc/sysconfig/iptables-config 設定ファイルの IPTABLES_SAVE_ON_RESTART ディレクティブがデフォルト値からに変更される /etc/sysconfig/iptables と yes、現在のルールはに保存され、既存のルールはファイルに移動し /etc/sysconfig/iptables.saveます。

  iptables-config ファイル 「iptables の制御スクリプト設定ファイル」 の詳細は、を参照してください。

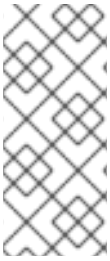
- status : ファイアウォールのステータスを表示し、アクティブなルールを一覧表示します。

  このオプションのデフォルト設定では、各ルールに IP アドレスが表示されます。ドメインおよびホスト名の情報を表示するには、/etc/sysconfig/iptables-config ファイルを編集し、の値をに変更 IPTABLES_STATUS_NUMERIC し noます。iptables-config ファイル 「iptables の制御スクリプト設定ファイル」 の詳細は、を参照してください。

- panic : ファイアウォールルールをすべてフラッシュします。設定されたすべてのテーブルのポリシーはに設定され DROPます。

  このオプションは、サーバーが危険にさらされることを認識している場合に役立ちます。ネットワークと物理的に切断したり、システムをシャットダウンしたりするのではなく、このオプションを使用して、追加のネットワークトラフィックを停止できますが、マシンを分析やその他のフォレンジックに準備が整った状態にすることができます。

- save : /etc/sysconfig/iptables を使用してファイアウォールルールを保存しiptables-saveます。詳細は「IPTables ルールの保存」を参照してください。



注記

同じ initscript コマンドを使用して IPv6 の netfilter を制御するには、本セクションに ip6tables iptables 記載 /sbin/service のコマンドのに置き換えます。IPv6 および netfilter の詳細は、を参照してください 「iptables および IPv6」。

#### 2.8.9.4.1. iptables の制御スクリプト設定ファイル

init スクリプトの動作は /etc/sysconfig/iptables-config 設定ファイル iptables によって制御されます。以下は、このファイルに含まれるディレクティブの一覧です。

- IPTABLES_MODULES : ファイアウォールがアクティブになると読み込む追加 iptables モジュールのスペース区切りの一覧を指定します。これには、接続追跡や NAT ヘルパーが含まれます。

- IPTABLES_MODULES_UNLOAD : 再起動して停止時にモジュールをアンロードします。このディレクティブは、以下の値を受け入れます。

  - yes : デフォルト値はです。ファイアウォールの再起動または停止の正しい状態を実現するには、このオプションを設定する必要があります。

  - no - このオプションは、netfilter モジュールのアンロードの問題がある場合にのみ設定する必要があります。

- IPTABLES_SAVE_ON_STOP : ファイアウォールが停止した /etc/sysconfig/iptables 場合に、現在のファイアウォールルールをに保存します。このディレクティブは、以下の値を受け入れます。

  - yes : 既存のルールをに保存し、ファイアウォールが停止した /etc/sysconfig/iptables 時に以前のバージョンを /etc/sysconfig/iptables.save ファイルに移動します。

  - no : デフォルト値はです。ファイアウォールが停止した時に既存のルールを保存しません。

- IPTABLES_SAVE_ON_RESTART : ファイアウォールが再起動すると、現在のファイアウォールルールを保存します。このディレクティブは、以下の値を受け入れます。

- yes : 既存のルールをに保存し、ファイアウォールが再起動し /etc/sysconfig/iptables たら、以前のバージョンを /etc/sysconfig/iptables.save ファイルに移動します。
- no : デフォルト値はです。ファイアウォールを再起動する際には、既存のルールを保存しません。

- IPTABLES_SAVE_COUNTER : すべてのチェーンおよびルールで、パケットとバイトカウンターをすべて保存して復元します。このディレクティブは、以下の値を受け入れます。
  - yes : カウンター値を保存します。
  - no : デフォルト値はです。カウンター値を保存しません。

- IPTABLES_STATUS_NUMERIC : ドメインまたはホスト名ではなく数値の IP アドレスを出力します。このディレクティブは、以下の値を受け入れます。
  - yes : デフォルト値はです。ステータス出力内の IP アドレスのみを返します。
  - no : ステータス出力内でドメインまたはホスト名を返します。

#### 2.8.9.5. iptables および IP セット

ipset ユーティリティーは、Linux カーネルで *IP セット* を管理するために使用されます。IP セットは、IP アドレス、ポート番号、IP と MAC アドレスのペア、または IP アドレスとポート番号のペアを格納するフレームワークです。セットは、セットが非常に大きい場合でも、セットに対して非常に高速な一致が行われるようにインデックス化されます。IP セットを使用すると、より簡単で管理可能な設定が可能になり、iptables を使用する際にパフォーマンス上の利点が得られます。iptables のマッチおよびターゲットは、カーネルの所定セットを保護する参照を作成します。セットを参照する単一の参照がある場合は、セットを破棄することはできません。

ipset を使用すると、以下のように iptables コマンドを使用できます。セット

```
~]# iptables -A INPUT -s 10.0.0.0/8 -j DROP
~]# iptables -A INPUT -s 172.16.0.0/12 -j DROP
~]# iptables -A INPUT -s 192.168.0.0/16 -j DROP
```

は以下のように作成されます。そして、

```
~]# ipset create my-block-set hash:net
~]# ipset add my-block-set 10.0.0.0/8
~]# ipset add my-block-set 172.16.0.0/12
~]# ipset add my-block-set 192.168.0.0/16
```

以下のように **iptables** コマンドで参照されます。セットが設定時間よりも複数使用される

```
~]# iptables -A INPUT -m set --set my-block-set src -j DROP
```

場合。セットに多くのエントリーが含まれる場合は、処理時間を保存するエントリーが多数含まれます。

#### 2.8.9.5.1. ipset のインストール

**ipset** ユーティリティーをインストールするには、**root** で以下のコマンドを実行します。使用方法に関するメッセージが

```
~]# yum install ipset
```

表示されます。

```
~]$ ipset -h
ipset v6.11

Usage: ipset [options] COMMAND
```

#### 2.8.9.5.2. ipset コマンド

**ipset** コマンドの形式は以下のとおりです。

```
ipset [options] コマンド [command-options]
```

ここでの *command* は以下のいずれかになります。

```
Create | add | del | test | destroy | list | save | restore | flush | rename | swap | help | version | -
```

使用できる *オプション* は以下のとおりです。

```
-exist | -output [ plain | save | xml ] | -quiet | -resolve | -sorted | -name | -terse
```

create コマンドは、IP データのセットを格納するために新しいデータ構造を作成するために使用されます。この add コマンドは、セットに新しいデータを追加します。追加されたデータはセットの要素と呼ばれます。

この -exist オプションは、要素がすでに存在する場合はエラーメッセージを非表示にします。また、タイムアウト値の更新に特別なロールがあります。タイムアウトを変更するには、ipset add コマンドを使用して要素の全データを再指定し、必要に応じてタイムアウト値のみを変更し、-exist オプションを使用します。

test オプションは、セット内に要素がすでに存在する場合をテストするためのものです。

create コマンドの形式は以下のとおりです。

```
ipset create set-name type-name [create-options]
```

*set-name* はユーザーが選択する適切な名前です。*type-name* は、セットで構成されるデータを格納するために使用されるデータ構造の名前です。*type-name* の形式は以下のとおりです。

```
method:datatype[,datatype[,datatype]]
```

データの保存に許可される方法は以下のとおりです。

```
メールボックス | ハッシュ | リスト
```

使用できるデータタイプは次のとおりです。

```
ip | net | mac | port | iface
```

セット内のエントリーを追加、削除、またはテストする場合は、セット内のエントリー（要素）を構成するデータ構文と同じコンマ区切りのデータ構文を使用する必要があります。以下に例を示します。

```
ipset add set-name ipaddr,portnum,ipaddr
```

注記

セットに IPv4 アドレスと IPv 6 アドレスを同時に含めることはできません。セットが作成されると、IPv4 または IPv inet6 6 inet の場合はファミリーにバインドされ、デフォルトはになり inetます。

**例2.3 IP セットの作成**

**ソース IP アドレス、ポート、宛先 IP アドレスで構成される IP セットを作成するには、以下のコマンドを実行します。セットが作成され**

```
~]# ipset create my-set hash:ip,port,ip
```

**たら、以下のようにエントリーを追加できます。**

```
~]# ipset add my-set 192.168.1.2,80,192.168.2.2
~]# ipset add my-set 192.168.1.2,443,192.168.2.2
```

**セットタイプには、以下のオプションのパラメーターが共通です。この設定を使用するには、セットの作成時に指定する必要があります。**

- **timeout : create コマンドで指定される値は、作成されるセットのデフォルト値になります。add コマンドで値が指定された場合、これは要素の初期値以外の値になります。**

**例2.4 IP セットの一覧表示**

**特定の IP セットの内容を一覧表示するには my-set、以下のコマンドを発行します。**

```
~]# ipset list my-set
Name: my-set
Type: hash:ip,port,ip
Header: family inet hashsize 1024 maxelem 65536
Size in memory: 8360
References: 0
Members:
192.168.1.2,tcp:80,192.168.2.2
192.168.1.2,tcp:443,192.168.2.2
```

**すべてのセットの一覧を表示するには、set name を省略します。**

**例2.5 IP セットの要素のテスト**

大規模なセットのコンテンツの一覧表示には時間がかかります。要素の存在は、以下のようにテストできます。

```
~]# ipset test my-set 192.168.1.2,80,192.168.2.2
192.168.1.2,tcp:80,192.168.2.2 is in set my-set.
```

#### 2.8.9.5.3. IP セットの種類

**bitmap:ip**

IPv4 ホストアドレス、ネットワーク範囲、またはセットの作成時に netmask オプションが使用される場合に CIDR 表記の prefix-length を持つ IPv4 ネットワークアドレスを保存します。オプションで、タイムアウト値、カウンター値、およびコメントを保存できます。65536 エントリーまで保存できます。bitmap:ip セットを作成するコマンドの形式は以下のとおりです。

```
ipset create set-name range start_ipaddr-end_ipaddr|ipaddr/prefix-length[netmask prefix-length] [タイムアウト 値] [カウンター] [comment]
```

例2.6 接頭辞の長さを使用したアドレスの範囲の IP セットの作成

接頭辞の長さを使用してアドレスの範囲の IP セットを作成するには、以下のように bitmap:ip セットタイプを使用します。

```
~]# ipset create my-range bitmap:ip range 192.168.33.0/28
```

セットが作成されたら、以下のようにエントリーを追加できます。

```
~]# ipset add my-range 192.168.33.1
```

一覧のメンバーを確認します。

```
~]# ipset list my-range
Name: my-range
Type: bitmap:ip
```

```
Header: range 192.168.33.0-192.168.33.15
Size in memory: 84
References: 0
Members:
192.168.33.1
```

アドレスの範囲を追加するには、以下を行います。

```
~]# ipset add my-range 192.168.33.2-192.168.33.4
```

一覧のメンバーを確認します。

```
~]# ipset list my-range
Name: my-range
Type: bitmap:ip
Header: range 192.168.33.0-192.168.33.15
Size in memory: 84
References: 0
Members:
192.168.33.1
192.168.33.2
192.168.33.3
192.168.33.4
```

**例2.7 Netmask を使用したアドレス範囲の IP セットの作成**

ネットマスクを使用してアドレスの範囲の IP セットを作成するには、以下のように **bitmap:ip** セットタイプを使用します。セットが作成され

```
~]# ipset create my-big-range bitmap:ip range 192.168.124.0-192.168.126.0 netmask 24
```

たら、以下のようにエントリーを追加できます。

```
~]# ipset add my-big-range 192.168.124.0
```

アドレスの追加を試みると、そのアドレスが含まれる範囲が追加されます。

```
~]# ipset add my-big-range 192.168.125.150
~]# ipset list my-big-range
Name: my-big-range
Type: bitmap:ip
Header: range 192.168.124.0-192.168.126.255 netmask 24
Size in memory: 84
References: 0
Members:
192.168.124.0
192.168.125.0
```

**bitmap:ip,mac**

IPv4 アドレスと MAC アドレスをペアとして保存します。65536 エントリーまで保存できます。

```
ipset create my-range bitmap:ip,mac range start_ipaddr-end_ipaddr | ipaddr/prefix-length[タイムアウト 値] [カウンター] [comment]
```

**例2.8 IPv4 MAC アドレスペアの範囲の IP セットの作成**

IPv4 MAC アドレスペアの範囲に IP セットを作成するには、以下のように bitmap:ip,mac セットタイプを使用します。セットの作成時に MAC アドレスを指定する必要

```
~]# ipset create my-range bitmap:ip,mac range 192.168.1.0/24
```

はありません。

セットが作成されたら、以下のようにエントリーを追加できます。

```
~]# ipset add my-range 192.168.1.1,12:34:56:78:9A:BC
```

**bitmap:port**

ポートの範囲を保存します。65536 エントリーまで保存できます。

**ipset create *my-port-range* bitmap:port range *start_port-end_port*[タイムアウト *値*] [カウンター] [comment]**

設定された match および SET ターゲット netfilter カーネルモジュールは、保存された数字を TCP または UDP ポート番号として解釈します。プロトコルは、オプションでポートとともに指定できます。サービス名が使用され、その名前が TCP サービスとして存在しない場合に proto のみ指定する必要があります。

**例2.9 ポートの範囲の IP セットの作成**

ポートの範囲に IP セットを作成するには、以下のように bitmap:port セットタイプを使用します。セットが作成され

```
~]# ipset create my-permitted-port-range bitmap:port range 1024-49151
```

たら、以下のようにエントリーを追加できます。

```
~]# ipset add my-permitted-port-range 5060-5061
```

**hash:ip**

ホストまたはネットワークアドレスをハッシュの形式で保存します。デフォルトでは、ネットワークプレフィックスの長さを付けずに指定するアドレスはホストのアドレスです。ゼロの IP アドレスは保存できません。

**ipset create *my-addresses* hash:ip [ファミリー[ inet | inet6 ]] [hashsize *値*] [maxelem *値*] [netmask *prefix-length*] [タイムアウト *値*]**

が family 省略されたアドレスが IPv4 アドレスとして解釈される場合、inet ファミリーはデフォルトでになります。hashsize 値は、使用する初期ハッシュサイズで、デフォルトはに設定され 1024ます。maxelem 値は、セットに保存できる要素の最大数で、デフォルトではに設定され 65536ます。

netfilter ツールは、最も特殊なネットワーク接頭辞を検索します。これは、一致するアドレスの最小ブロックを探します。

例2.10 IP アドレスの IP セットの作成

IP アドレスの IP セットを作成するには、以下のように hash:ip セットタイプを使用します。セットが作成され

```
~]# ipset create my-addresses hash:ip
```

たら、以下のようにエントリーを追加できます。

```
~]# ipset add my-addresses 10.10.10.0
```

netmask や timeout などの追加オプションが必要な場合は、セットの作成時に指定する必要があります。たとえば、maxelem オプション

```
~]# ipset create my-busy-addresses hash:ip maxelem 24 netmask 28 timeout 100
```

はセット内の要素の合計数に制限されるため、メモリー領域が予約されます。

timeout オプションは、指定された秒数の要素がセットのみに存在することを意味します。例：

```
~]# ipset add my-busy-addresses timeout 100
```

以下の出力は、タイムアウト期間の終了時にセットから

```
[root@rhel6 ~]# ipset add my-busy-addresses 192.168.60.0 timeout 100
[root@rhel6 ~]# ipset list my-busy-addresses
Name: my-busy-addresses
Type: hash:ip
Header: family inet hashsize 1024 maxelem 24 netmask 28 timeout 100
Size in memory: 8300
References: 0
```

```
Members:
192.168.60.0 timeout 90
[root@rhel6 ~]# ipset list my-busy-addresses
Name: my-busy-addresses
Type: hash:ip
Header: family inet hashsize 1024 maxelem 24 netmask 28 timeout 100
Size in memory: 8300
References: 0
Members:
192.168.60.0 timeout 83
```

要素が削除されることを示しています。

その他の例は、ipset(8) man ページを参照してください。

#### 2.8.9.6. iptables および IPv6

iptables-ipv6 パッケージがインストールされている場合は、Red Hat Enterprise Linux の netfilter が次世代 IPv6 インターネットプロトコルをフィルタリングできます。IPv6 netfilter の操作に使用されるコマンドはです ip6tables。

このコマンドのディレクティブの大半は、に使用されるものと同じですが iptables、nat テーブルはまだサポートされていません。つまり、マスカレードやポート転送などの IPv6 ネットワークアドレス変換タスクはまだ実行できません。

のルール ip6tables は /etc/sysconfig/ip6tables ファイルに保存されます。init スクリプトによって保存される以前のルール ip6tables は /etc/sysconfig/ip6tables.save ファイルに保存されます。

ip6tables init スクリプトの設定オプションはに保存され /etc/sysconfig/ip6tables-config、各ディレクティブの名前はカウンターパートと若干異なり iptables ます。

たとえば、iptables-config ディレクティブの場合には、ip6tables-config ファイル IPTABLES_MODULES の同等のものはです IP6TABLES_MODULES。

#### 2.8.9.7. その他のリソース

ファイアウォールおよび Linux Netfilter サブシステムには、本章で説明できないさまざまな側面があります。詳細は、以下の資料を参照してください。

#### 2.8.9.7.1. 便利なファイアウォールの Web サイト

- http://www.netfilter.org/ - netfilter/iptables プロジェクトのホームには、特定の問題に対処する FAQ と iptables、Linux IP ファイアウォールの保守管理者（Linux IP ファイアウォールの担当者）によるさまざまな便利なガイドなどに関するさまざまな情報が含まれています。HOWTO のドキュメントには、基本的なネットワーク概念、カーネルパケットのフィルタリング、NAT 設定などが含まれます。

- http://www.tldp.org/: Linux ドキュメントプロジェクトには、ファイアウォールの作成および管理に関する便利なガイドがいくつか含まれています。

- http://www.iana.org/assignments/port-numbers - インターネット割り当て番号機関が割り当てた登録済みおよび共通サービスポートの公式リストです。

#### 2.8.9.7.2. 関連ドキュメント

- Bill McCarty 著の『Red Hat Linux ファイアウォール』。Red Hatwell は、Netfilter や Netfilter などのオープンソースパケットフィルタリング技術を使用して、ネットワークおよびサーバーのファイアウォールを構築する包括的なリファレンスです iptables。これには、ファイアウォールログの分析、ファイアウォールルールの開発、さまざまなグラフィカルツールを使用してファイアウォールのカスタマイズを行うトピックが含まれます。

- Kernel Ziegler『による Linux ファイアウォール』 (New Rinksville)- 2.2 カーネルと ipchains Netfilter との両方を使用してファイアウォールを構築するに関する多くの情報が含まれてい iptablesます。また、リモートアクセスの問題や侵入検出システムなどの追加のセキュリティートピックも取り上げます。

#### 2.8.9.7.3. インストールした IP テーブルに関するドキュメント

- man iptables : の説明 iptables と、ターゲット、オプション、および一致拡張機能に関する包括的な一覧が含まれます。

---

[3] システム BIOS はメーカーによって異なるため、いずれかのタイプのパスワード保護のみをサポートするものもあれば、いずれのタイプのパスワード保護もサポートしないものもあります。

[4] GRUB は暗号化されていないパスワードも使用できますが、MD5 ハッシュを使用してセキュリティーを強化することが推奨されます。

# 第3章 暗号化

保護する必要のあるデータには、移動しないデータと移動するデータという 2 つのタイプのデータがあります。これらの異なるタイプのデータは同様の技術を使用して保護されますが、実装は完全に異なる可能性があります。単一の保護的実装は、同じ情報が保存され、異なる時点で移動する可能性がある、すべての不正アクセスを防ぎます。

## 3.1. 復元中のデータ

保持するデータは、ハードドライブ、テープ、CD、DVD、ディスク、またはその他のメディアに保存されているデータです。この情報の脅威は、物理的に盗まれたことが原因です。写真のラップトップ、CD はメールを経由し、誤った場所に残されたバックアップテープはすべて、盗難によりデータが危険にさらされるイベントの例です。メディア上でデータを暗号化すると、アクセスされるデータの可能性が低くなります。

### 3.1.1. 完全なディスク暗号化

完全なディスクまたはパーティションの暗号化は、データを保護する最善の方法の 1 つです。各ファイルが保護されているだけでなく、これらのファイルの一部を含む可能性のある一時ストレージも保護されます。完全なディスク暗号化は、すべてのファイルを保護するため、ファイルを保護するものを選択する必要がなく、不明なものを選択する必要はありません。

Red Hat Enterprise Linux 6 は、LUKS 暗号化をネイティブにサポートします。LUKS は、ハードドライブのパーティションを一括暗号化し、コンピューターがオフの間はデータを保護します。また、これにより、シングルユーザーモードを使用してコンピューターにログインしようとする、攻撃者やアクセスの取得を試みる攻撃者からコンピューターを保護します。

LUKS などの完全なディスク暗号化ソリューションは、コンピューターがオフの時にのみデータを保護します。コンピューターの電源がオンになり、LUKS がディスクを復号すると、そのディスクのファイルは、通常、そのディスクにアクセスできるすべてのユーザーが利用できます。このコンピューターの電源がオンのときにファイルを保護するには、ファイルベースの暗号化などの別のソリューションと組み合わせて完全なディスク暗号化を使用します。また、コンピューター外のときに必ずコンピューターをロックするのを忘れないでください。パスフレーズで保護されるスクリーンパーサーは、非アクティブを数分後にアクティブにするよう設定されていることは、侵入を防ぎます。LUKS の詳細はを参照してください 「LUKS ディスクの暗号化」。

### 3.1.2. ファイルベースの暗号化

ファイルベースの暗号化は、CD、フラッシュドライブ、外部ハードドライブなど、モバイルストレージデバイスのファイルの内容を保護するために使用されます。ファイルベースの暗号化ソリューションによっては、コンピューターに物理的にアクセスできる攻撃者は、暗号化したファイルが一部の状況下で回復できるままになる可能性があります。お使いのコンピューターにアクセスできる可能性の

ある攻撃者からこれらのファイルのコンテンツを保護するには、完全なディスク暗号化などの別のソリューションと組み合わせて、ファイルベースの暗号化を使用します。

### 3.1.3. LUKS ディスクの暗号化

Linux Unified Key Setup-on-disk-format（または LUKS）を使用すると、Linux コンピューターのパーティションを暗号化できます。これは、モバイルコンピューターやリムーバブルメディアの場合に特に重要です。LUKS は、複数のユーザー鍵が、パーティションのバルク暗号化に使用されるマスター鍵を復号化できるようにします。

**LUKS の概要**

**LUKS の機能**

- LUKS は、ブロックデバイス全体を暗号化するため、リムーバブルストレージメディアやノート PC ディスクドライブなどのモバイルデバイスのコンテンツを保護するのに適しています。
- 暗号化されたブロックデバイスの基本的な内容は任意です。これにより、swap デバイスの暗号化に役立ちます。また、とりわけデータストレージ用にフォーマットしたブロックデバイスを使用する特定のデータベースに関しても有用です。
- LUKS は、既存のデバイスマッパーのカーネルサブシステムを使用します。
- LUKS は、パラフレーズの強化を提供し、辞書攻撃から保護します。
- LUKS デバイスには複数のキースロットが含まれているため、ユーザーはバックアップキー/パスフレーズを追加できます。

**LUKS が行わ ない こと**

- LUKS は、多くのユーザーが同じデバイスにアクセスする鍵をそれぞれ所有することが必要となるアプリケーションには適していません。
- LUKS は、ファイルレベルの暗号化を必要とするアプリケーションには適していません。

#### 3.1.3.1. Red Hat Enterprise Linux の LUKS 実装

Red Hat Enterprise Linux 6 は、LUKS を使用してファイルシステムの暗号化を行います。デフォルトではインストール時に、ファイルシステムを暗号化するオプションが指定されていません。ハードドライブを暗号化するオプションを選択すると、コンピューターを起動するたびにパスフレーズの入力が求められます。このパスフレーズは、パーティションの復号に使用されるバルク暗号鍵の「ロックを解除」します。デフォルトのパーティションテーブルの変更を選択すると、暗号化するパーティションを選択できます。この設定は、パーティションテーブル設定で行われます。

LUKS（を参照 cryptsetup --help）に使用されるデフォルトの暗号は aes-cbc-essiv:sha256 です。インストールプログラムである Anaconda は、XTS モード aes-xts-plain64 でデフォルトで AES 暗号を使用することに注意してください。LUKS のデフォルトの鍵サイズは 256 ビットです。Anaconda（XTS モード）を使用した LUKS のデフォルトの鍵サイズは 512 ビットです。

**警告**

デフォルトの暗号化属性を変更すると、システムのパフォーマンスに影響があり、さまざまなセキュリティーリスクにシステムが公開される可能性があります。暗号化に関する詳しい知識がなくても、システムのデフォルトの暗号化属性を変更しないでください。また、使用される暗号の組み合わせの機能を理解することはできません。

Red Hat では、デフォルトの暗号の使用を強く推奨します。デフォルトとして設定された暗号以外に暗号を使用する必要がある場合は、--cipher および --key-size オプションを使用してパーティションを初期化できます。コマンドの構文は以下のとおりです。

```
cryptsetup --verify-passphrase --cipher <cipher>-<mode>-<iv> --key-size <key-size> luksFormat
<device>
```

<cipher> -<mode>- <iv> は、 使用される暗号を表す文字列です。文字列は、ブロック暗号、ブロック暗号モード、および初期ベクトル(IV)の 3 つの部分で構成されます。

ブロック暗号は、データブロック上で動作し、一括データの暗号化と復号を可能にする決定的なアルゴリズムです。Red Hat Enterprise Linux で利用可能なブロック暗号は次のとおりです。

- AES - Advanced Encryption Standard。128 ビット、192 ビット、および 256 ビットの長さを持つ暗号化キーを使用した 128 ビット対称ブロック暗号です。詳細は、FIPS PUB 197 を参照してください。

- Twofish - 範囲の暗号化キーを 128 ビットから 256 ビットまでで操作する 128 ビットのブロック暗号。

- serpent - 128 ビット、192- ビット、および 256 ビットの暗号化鍵を使用する 128 ビットブロック暗号。

- cast5 - 範囲の暗号鍵（40 ビットから 128 ビット）に対応する 64 ビットの Feistel 暗号。詳細は RFC 2144 を参照してください。

- cast6 - 128 ビット、160 ビット、192- ビット、224 ビット、または 256 ビット暗号鍵を使用する 128 ビットの Feistel 暗号鍵。詳細は RFC 2612 を参照してください。

ブロック暗号モードは、データを安全に暗号化または復号化するために一括データにブロック暗号を繰り返し適用する方法を示します。以下のモードを使用できます。

- CBC - Cipher Block Chaining - 詳細は NIST SP 800-38A を参照してください。

- XTS - XEX Tweakable Block Cipher with Ciphertext Bonaling。詳細は IEEE 1619 または NIST SP 800-38E を参照してください。

- CTR - Counter( NIST SP 800-38A )を参照してください。

- ECB: 電子コードブック（詳しくは NIST SP 800-38A を参照してください）。

- CFB - 暗号フィードバック。詳細は NIST SP 800-38A を参照してください。

- エラーメッセージ - 出力フィードバック。詳細は NIST SP 800-38A を参照してください。

最初のベクトルは、暗号文のランダム化に使用されるデータのブロックです。IV は、同じプレーンテキストの繰り返し暗号化によって、異なる暗号文の出力を提供するようになります。IV を同じ暗号鍵で再利用することはできません。CBC モードの暗号については、IV を 予測できない必要があり ます。そうしないと、システムが特定の基準攻撃に対して脆弱になる可能性があります（詳細は、LUKS/cryptsetup FAQ を参照してください）。Red Hat は、以下の IV を AES で使用することを推奨します。

- ESSIV - 暗号化された Salt-Sector Initialization Vector - この IV は CBC モードの暗号に使用する必要があります。デフォルトのハッシュ sha256 を使用する必要があります。

- plain64（または plain）- IV セクターオフセット - この IV は、XTS モードの暗号に使用する必要があります。

使用される暗号化キーの長さを指定することもできます。キーのサイズは、ブロック暗号モードとブロック暗号モードの使用される組み合わせによって異なります。キーの長さを指定しないと、LUKS は指定の組み合わせのデフォルト値を使用します。たとえば、CBC モードで AES に 128 ビットキーを使用する場合は、LUKS は AES-128 実装を使用してパーティションを暗号化し、XTS モードで AES-256 実装に 512 ビットキーを指定すると、AES-256 実装が使用されます。XTS モードは 2 つの鍵で動作します。1 つ目はツイータブル暗号化で決定され、2 番目は通常の暗号化の場合は 2 番目であることに注意してください。

#### 3.1.3.2. ディレクトリーの手動暗号化

**警告**

この手順では、暗号化しているパーティションのデータをすべて削除します。WILL はすべての情報が失われます。この手順を開始する前に、データを外部ソースにバックアップしてください。

1. root で次のコマンドを実行します。

```
telinit 1
```

2. 既存の /home をアンマウントします。

```
umount /home
```

3. 前の手順のコマンドで失敗した場合は、を使用して /home fuser にカーソルを合わせ、これを強制終了します。

```
fuser -mvk /home
```

4. /home がマウントされていないことを確認します。

```
grep home /proc/mounts
```

5. パーティションにランダムデータを入力します。

```
shred -v --iterations=1 /dev/VG00/LV_home
```

このコマンドは、デバイスの連続書き込み速度で続行され、完了するのに時間がかかる場合があります。暗号化されていないデータが使用されているデバイスに残されないようにし、暗号化したデータが含まれるデバイスの部分をランダムなデータだけでなく、混乱させないようにすることが重要な手順です。

6. パーティションを初期化します。

```
cryptsetup --verbose --verify-passphrase luksFormat /dev/VG00/LV_home
```

7. 新規暗号化したデバイスを開きます。

```
cryptsetup luksOpen /dev/VG00/LV_home home
```

8. デバイスが存在することを確認します。

```
ls -l /dev/mapper | grep home
```

9. ファイルシステムを作成します。

```
mkfs.ext3 /dev/mapper/home
```

10. **ファイルシステムをマウントします。**

```
mount /dev/mapper/home /home
```

11. **ファイルシステムが表示されることを確認します。**

```
df -h | grep home
```

12. **以下を /etc/crypttab ファイルに追加します。**

```
home /dev/VG00/LV_home none
```

13. **/etc/fstab ファイルを編集し、/home の古いエントリーを削除し、以下の行を追加します。**

```
/dev/mapper/home /home ext3 defaults 1 2
```

14. **デフォルトの SELinux セキュリティーコンテキストを復元します。**

```
/sbin/restorecon -v -R /home
```

15. **マシンを再起動します。**

```
shutdown -r now
```

16. **のエントリー /etc/crypttab により、システムの起動時に luks パスフレーズが要求されます。**

17. **root としてログインし、バックアップを復元します。**

**これで、コンピューターの電源が切れている間は、すべてのデータに対して暗号化されたパーティションが安全に休止状態になりました。**

#### 3.1.3.3. 既存のデバイスへの新規パスフレーズの追加

以下のコマンドを使用して、既存のデバイスに新しいパスフレーズを追加します。

```
cryptsetup luksAddKey <device>
```

認証既存のパスのいずれかを要求すると、新しいパスフレーズを入力するように求められます。

#### 3.1.3.4. 既存デバイスからのパスフレーズの削除

以下のコマンドを使用して、既存のデバイスからパスフレーズを削除します。

```
cryptsetup luksRemoveKey <device>
```

削除するパスフレーズの入力が求められてから、認証用に残りのパスフレーズを入力するよう求められます。

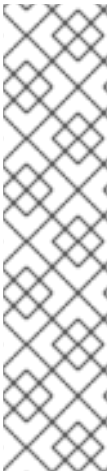
#### 3.1.3.5. Anaconda での暗号化ブロックデバイスの作成

システムのインストール時に、暗号化されたデバイスを作成できます。これにより、暗号化されたパーティションでシステムを簡単に設定することができます。

ブロックデバイスの暗号化を有効にするには、パーティション、ソフトウェア RAID アレイ、または論理ボリュームを作成するときに自動パーティションを設定するチェックボックスを選択する場合は、システム の 暗号 化チェックボックスを選択します。パーティション設定が完了すると、暗号化パスフレーズの入力が求められます。このパスフレーズは、暗号化されたデバイスへのアクセスに必要です。LUKS デバイスが存在し、インストールプロセスの初期段階でそのデバイスに正しいパスフレーズを提供している場合は、パスフレーズの入力ダイアログボックスにもチェックボックスが含まれます。このチェックボックスをチェックすると、既存の暗号化済みブロックデバイスの利用可能なスロットに新しいパスフレーズを追加することが分かります。

注記

自動パーティション 設定画面で システムを暗号化 するチェックボックスを選択してからカスタムレイアウトを作成 しても、ブロックデバイスは自動的に暗号化されません。



注記

kickstart ファイルを使用すると、新しい暗号化したブロックデバイスごとに、個別のパスフレーズを設定できます。また、Anaconda のデフォルト暗号である aes-xts-plain64 が適切でない場合、Kickstart では異なるタイプの暗号化を指定できます。暗号化するデバイスの依存関係では、、、autopart part partition logvol、および raid ディレクティブ --cipher=*<cipher-string>* とともにを指定できます。このオプションは、オプションとともに使用する必要があります。この --encrypted オプションを使用しないと、何も影響しません。*<cipher-string>* 形式および可能な暗号の組み合わせの詳細は、を参照してください 「Red Hat Enterprise Linux の LUKS 実装」。キックスタート設定の詳細は、『Red Hat Enterprise Linux 6 インストールガイド』を参照してください。

#### 3.1.3.6. その他のリソース

Red Hat Enterprise Linux で、LUKS またはハードドライブの暗号化に関する追加情報は、以下のいずれかのリンクを参照してください。

- LUKS ホームページ
- LUKS/cryptsetup FAQ
- LUKS - Linux Unified Key Setup 64- article
- HOWTO: 2 番目のハードドライブと pvmove を使用した暗号化された物理ボリューム (PV)の作成

### 3.2. MOTION のデータ

移動するデータは、ネットワーク経由で送信されるデータです。移動中のデータに対する脅威は傍受と変更です。ユーザー名とパスワードは、傍受され、他のユーザーが個人の偽装や機密情報へのアクセスを取得するため、保護なしにネットワーク上で送信しないでください。ネットワークセッションを暗号化すると、移動するデータのセキュリティーレベルが向上します。

攻撃者がそのデータを格納しているコンピューターに近づける必要がないため、移動中のデータは攻撃者に対して特に脆弱です。暗号化トンネルは、通信パスに基づいてデータを保護できます。

#### 3.2.1. 仮想プライベートネットワーク

仮想プライベートネットワーク(VPN)は、全ポートでコンピューターまたはコンピューターのネットワーク間で暗号化されたトンネルを提供します。VPN が設定されていると、クライアントからのネットワークトラフィックは、暗号化されたトンネルを介してサーバーに転送されます。つまり、クライアントは、VPN 経由で接続するサーバーと同じネットワークに論理的に配置されます。VPN は非常に一般的であり、使いやすく、設定が簡単です。

### 3.2.2. セキュアなシェル

*Secure Shell*（）SSH）は、セキュアなチャンネルで別のシステムと通信するために使用される強力なネットワークプロトコルです。SSH を介した転送は暗号化され、傍受から保護されます。暗号化ログインを使用して、従来のユーザー名とパスワードよりも優れた認証方法を提供することもできます。「暗号化ログイン」を参照してください。

SSH は、アクティベートが非常に簡単です。sshd デーモンを起動すると、システムは接続を許可し、接続プロセス中に正しいユーザー名とパスワードが提供されると、システムへのアクセスを許可します。SSH サービスの標準 TCP ポートはです 22。ただし、/etc/ssh/sshd_config 設定ファイルを変更してサービスを再起動すると、これを変更できます。このファイルには、SSH のその他の設定オプションも含まれます。

デフォルトでは、sshd サービスはシステムの起動時に自動的に起動します。root で以下のコマンドを実行して、デーモンのステータスをクエリーします。

```
~]# service sshd status
```

sshd サービスを再起動する必要がある場合は、root で以下のコマンドを発行します。

```
~]# service sshd restart
```

システムサービスの管理に関する詳細は、『Red Hat Enterprise Linux 6 デプロイメントガイド』の「サービス『とデーモン』」の章を参照してください。

Secure Shell（）SSH）は、コンピューター間で暗号化されたトンネルも提供しますが、単一のポートのみを使用します。ポート転送は SSH トンネル上で行うことができ、トラフィックはそのトンネルを通過するため暗号化されますが、ポート転送を使用する方法は fluid としてではありません。VPN (「仮想プライベートネットワーク」).

#### 3.2.2.1. 暗号化ログイン

SSH は、コンピューターへのログインに暗号鍵の使用をサポートします。これは、パスワードのみを使用するよりも安全です。このメソッドと他の認証方法を組み合わせる場合は、マルチファクター認

証と見なされます。複数 「複数の認証方法」 の認証方法の使用に関する詳細は、を参照してください。

認証に暗号鍵を使用できるようにするには、/etc/ssh/sshd_config ファイルの PubkeyAuthentication 設定ディレクティブをに設定する必要があり yesます。これはデフォルト設定であることに注意してください。ログインにパスワードを使用するのを無効 no にするには、PasswordAuthentication ディレクティブをに設定します。

SSH キーは、ssh-keygen コマンドを使用して生成できます。追加の引数なしで呼び出されると、2048 ビットが作成されます。 RSA キーセット。キーは、デフォルトで ~/.ssh ディレクトリーに保管されます。-b スイッチを使用してキーのビット強度を変更できます。通常、2048 ビットの鍵で十分です。SSH キーの 生成に関する詳細 は、『 Red Hat Enterprise Linux 6 デプロイメントガイド』の「キーペアの生成」の章を参照してください。

~/.ssh ディレクトリーには、2 つのキーが表示されるはずです。ssh-keygen コマンドの実行時にデフォルトを指定した場合は、生成されたファイルには id_rsa という名前が付けられ、プライベートキーと公開鍵がそれぞれ id_rsa.pub 含まれます。秘密鍵を、ファイルの所有者以外のすべてのユーザーが読み取りできないようにすることで、常に秘密鍵を公開から保護する必要があります。ただし、公開鍵は、ログインするシステムに転送する必要があります。ssh-copy-id コマンドを使用すると、鍵をサーバーに転送できます。

```
~]$ ssh-copy-id -i [user@]server
```

また、このコマンドにより、公開鍵が サーバー 上の ~/.ssh/authorized_key ファイルに自動的に追加されます。sshd デーモンは、サーバーへのログインを試みる際にこのファイルをチェックします。

パスワードおよびその他の認証メカニズムと同様に、SSH キーは定期的に変更する必要があります。これを行う場合は、authorized_key ファイルから未使用の鍵を削除するようにしてください。

#### 3.2.2.2. 複数の認証方法

複数の認証方法（マルチファクター認証）を使用すると、権限のないアクセスに対する保護レベルが向上します。そのため、攻撃を防ぐためにシステムを強化する際に考慮する必要があります。マルチファクター認証を使用するシステムにログインしようとするユーザーは、アクセスを付与するために指定されたすべての認証方法を正常に完了する必要があります。

/etc/ssh/sshd_config ファイルの AuthenticationMethods 設定ディレクティブを使用して、使用する認証方法を指定します。このディレクティブを使用して、必要な認証方法の一覧を複数定義できることに注意してください。その場合、ユーザーは少なくとも一覧のいずれかですべてのメソッドを完了する必要があります。一覧は空白で区切る必要があり、一覧内の個々の認証メソッド名はカンマで区切る必要があります。以下に例を示します。

```
AuthenticationMethods publickey,gssapi-with-mic publickey,keyboard-interactive
```

上記の **AuthenticationMethods** ディレクティブを使用して設定された **sshd** デーモンは、ユーザーが正常にログインしようとしている場合に限り、**gssapi-with-mic** または **publickey** 認証を行ってください **keyboard-interactive**。要求された各認証方法は、**/etc/ssh/sshd_config** ファイルで対応する設定ディレクティブ（など **PubkeyAuthentication**）を使用して明示的に有効にする必要があります。利用可能な認証方法の一般的 **ssh(1)** なリストは、の 『**AUTHENTICATION**』 セクションを参照してください。

#### 3.2.2.3. SSH のセキュリティー保護のその他の方法

**プロトコルのバージョン**

**Red Hat Enterprise Linux** で提供される **SSH** プロトコルの実装は、プロトコルの **SSH-1** および **SSH-2** の両方をサポートしますが、可能な場合は後者のみを使用してください。**SSH-2** バージョンには、古い **SSH-1** に対する多くの改善が含まれています。また、多くの高度な設定オプションは、**SSH-2** を使用する場合にのみ利用できます。

**SSH** プロトコルが使用されている認証および通信を保護するエクステントを最大化するために、**SSH -2** を使用することが推奨されます。**sshd** デーモンがサポートするプロトコルのバージョンまたはバージョンは、**/etc/ssh/sshd_config** ファイルの **Protocol configuration** ディレクティブを使用して指定できます。デフォルト設定はです **2**。

**鍵のタイプ**

**ssh-keygen** コマンドにより **SSH-2** のペアが生成されます。 **RSA -t** オプションを使用して、デフォルトでキーを生成するように指示できます。 **DSA** または **ECDSA** 鍵も。**The ECDSA (elliptic Curve Digital Signature Algorithm)**は、同じ対称鍵の長さで優れたパフォーマンスを提供します。また、短いキーも生成します。

**デフォルト以外のポート**

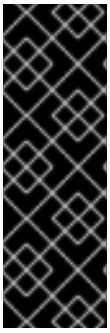
デフォルトでは、**sshd** デーモンは **22** ネットワークポートをリッスンします。ポートを変更すると、自動化したネットワークスキャンに基づく攻撃にシステムがさらされる可能性が減るため、あいまいさによりセキュリティーが向上します。ポートは、**/etc/ssh/sshd_config** 設定ファイルの **Port** ディレクティブを使用して指定できます。また、デフォルト以外のポートを使用できるように、デフォルトの **SELinux** ポリシーを変更する必要もあります。これを行うには、**root** で以下のコマンドを入力して、**ssh_port_t SELinux** タイプを変更します。

```
~]# semanage -a -t ssh_port_t -p tcp port_number
```

上記のコマンドで、***port_number*** を、**Port** ディレクティブで指定された新しいポート番号に置き換えます。

**root ログインなし**

特定のユースケースで root ユーザーとしてログインする必要がない場合は、/etc/ssh/sshd_config ファイルで設定ディレクティブを PermitRootLogin 設定することを検討 no してください。root ユーザーとしてログインする可能性を無効にすることで、管理者は通常のユーザーとしてログインして root 権限を取得すると、どのユーザーがどの特権コマンドを実行するかを監査できます。



重要

本セクションでは、SSH 設定を保護する最も一般的な方法に注意を促します。必ずしも、この提案された対策の一覧は完全または限定的であると見なされません。sshd デーモンの動作を修正する sshd_config(5) ために利用可能なすべての設定ディレクティブの説明は、を参照してください。これは、ssh(1) の基本的な SSH の概念の説明です。

### 3.3. OPENSSL INTEL AES-NI ENGINE

Intel Advanced Encryption Standard(AES)New Instructions(AES-NI)エンジンは特定の Intel プロセッサーで利用でき、非常に高速なハードウェア暗号化と復号化を可能にします。

注記

AES-NI エンジンをサポートする Intel プロセッサーの一覧 は、Intel の ARK を参照して ください。

AES-NI エンジンは、検出されたプロセッサーがサポートされているプロセッサーの一部である場合に自動的に有効になります。プロセッサーがサポートされていることを確認するには、以下の手順に従います。

1. プロセッサーに AES 命令セットがあることを確認します。

```
~]# grep -m1 -o aes /proc/cpuinfo
aes
```

2. root で以下のコマンドを実行し、その出力を比較します。後続のコマンドのパフォーマンスが大幅に向上する場合は、AES-NI が有効化されていることを示しています。以下の出力は簡潔にするために短いことに注意してください。

```
~]# openssl speed aes-128-cbc
The 'numbers' are in 1000s of bytes per second processed.
type             16 bytes     64 bytes    256 bytes   1024 bytes   8192 bytes
aes-128 cbc      99696.17k   107792.98k   109961.22k   110559.91k   110742.19k
```

```
~]# openssl speed -evp aes-128-cbc
The 'numbers' are in 1000s of bytes per second processed.
type             16 bytes     64 bytes    256 bytes   1024 bytes   8192 bytes
aes-128-cbc     800450.23k   873269.82k   896864.85k   903446.19k   902752.94k
```

**OpenSSH の速度をテストするには、以下のようなコマンドを実行します。**

```
~]# dd if=/dev/zero count=100 bs=1M | ssh -c aes128-cbc localhost "cat >/dev/null"
root@localhost's password:
100+0 records in
100+0 records out
104857600 bytes (105 MB) copied, 4.81868 s, 21.8 MB/s
```

**AES-NI エンジンの詳細は、Intel® Advanced Encryption Standard Instructions(AES-NI) を参照してください。**

### 3.4. RANDOM NUMBER GENERATOR の使用

**簡単に破損できない安全な暗号鍵を生成できるようにするには、ランダムな数字のソースが必要です。通常は、数字がランダムなほど、一意の鍵を取得する可能性が高くなります。通常、乱数を生成するエントロピーは、コンピューティング環境から「noise」またはハードウェア *乱数ジェネレーター* を使用して取得されます。**

**rng-tools パッケージの一部である rngd デーモンは、環境侵害とハードウェア乱数ジェネレーターの両方を使用してエントロピーを抽出できます。デーモンは、ランダム性のソースによって提供されたデータが十分にランダムなものかどうかをチェックしてから、カーネルの random-number エントロピープールに保存します。生成されるランダムな数字は /dev/random、およびの /dev/urandom 文字デバイスから利用できます。**

**/dev/random との違いは、以前のデバイス /dev/urandom がブロックデバイスであることです。つまり、エントロピーの量が、正しくランダムな出力を生成するのに不十分であると判断すると、数字を提供しなくなります。逆に /dev/urandom、ブロック以外のソースで、カーネルのエントロピープールを再読み込みし、擬似アンダークラウド番号を無制限に提供でき、エントロピーが少なくなります。/dev/urandom したがって、は長期暗号鍵の作成に使用しないでください。**

**rng-tools パッケージをインストールするには、root で以下のコマンドを実行します。**

```
~]# yum install rng-tools
```

**rngd デーモンを起動するには、root で以下のコマンドを実行します。**

```
~]# service rngd start
```

デーモンのステータスをクエリーするには、以下のコマンドを使用します。

```
~]# service rngd status
```

オプションのパラメーターを指定して **rngd** デーモンを起動するには、直接実行します。たとえば、（以外の **/dev/hwrandom**）**random-number** 入力の代替ソースを指定するには、以下のコマンドを使用します。

```
~]# rngd --rng-device=/dev/hwrng
```

上記のコマンドは、ランダムな数字が読み取られるデバイス **/dev/hwrng** として、で **rngd** デーモンを起動します。同様に、**-o**（または **--random-device**）オプションを使用して、ランダムな出力（デフォルト以外の **/dev/random**）のカーネルデバイスを選択できます。以下を参照してください。 **rngd(8)** 利用可能なすべてのオプションの一覧の **man** ページです。

**rng-tools** パッケージには、データのランダム性を確認するために使用できる **rngtest** ユーティリティーも含まれます。の出力のランダム性レベルをテストするには **/dev/random**、以下のように **rngtest** ツールを使用します。

```
~]$ cat /dev/random | rngtest -c 1000
rngtest 2
Copyright (c) 2004 by Henrique de Moraes Holschuh
This is free software; see the source for copying conditions.  There is NO warranty; not even for
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

rngtest: starting FIPS tests...
rngtest: bits received from input: 20000032
rngtest: FIPS 140-2 successes: 1000
rngtest: FIPS 140-2 failures: 0
rngtest: FIPS 140-2(2001-10-10) Monobit: 0
rngtest: FIPS 140-2(2001-10-10) Poker: 0
rngtest: FIPS 140-2(2001-10-10) Runs: 0
rngtest: FIPS 140-2(2001-10-10) Long run: 1
rngtest: FIPS 140-2(2001-10-10) Continuous run: 0
rngtest: input channel speed: (min=308.697; avg=623.670; max=730.823)Kibits/s
rngtest: FIPS tests speed: (min=51.971; avg=137.737; max=167.311)Mibits/s
rngtest: Program run time: 31461595 microseconds
```

**rngtest** ツールの出力で表示される多くの障害は、テストされたデータのランダム性が最適で、依存しないことを示しています。以下を参照してください。 **rngtest(1) rngtest** ユーティリティーで利用可能なオプションの一覧の **man** ページです。

## 3.5. GNU PRIVACY GUARD(GPG)

GnuPG(GPG)は、ファイルまたは電子メールメッセージの署名および暗号化を可能にする PGP のオープンソースバージョンです。これは、メッセージまたはファイルの整合性を維持し、ファイルまたは電子メール内に含まれる情報の機密性を保護するのに役立ちます。電子メールでは、GPG はデュアル保護を提供します。Data at Rest の保護だけでなく、メッセージをネットワーク経由で送信した後にデータの Motion 保護機能も提供します。これら 「Motion のデータ」 の概念の詳細については 「復元中のデータ」、およびを参照してください。

GPG は、自身を特定して通信を認証するために使用されます。通信には、認識しないユーザーを含めて通信を認証します。GPG では、GPG 署名の電子メールを読むすべてのユーザーが、その信頼性を検証することができます。つまり、GPG により、実際に署名した通信がユーザーからの通信であることを確実にできます。GPG は、サードパーティーがコードを変更したり、対話の傍受やメッセージを変更できないのを防ぐのに役立ちます。

### 3.5.1. GNOME での GPG 鍵の作成

GNOME で GPG キーを作成するには、以下の手順に従います。

1. Seahorse ユーティリティーをインストールします。これにより、GPG キー管理が容易になります。

```
~]# yum install seahorse
```

2. キーを作成するには、Applications → Accessories メニューから Passwords and Encryption Keys、アプリケーション Seahorse を起動するメニューを選択します。

3. File メニューでを選択し New、PGP Key を選択します。次に、をクリックし Continueます。

4. フルネーム、メールアドレス、およびユーザーを記述するオプションのコメントを入力します（例： john C.anda、jsmith@example.com、Software Engineer）。をクリックし Createます。キーパスフレーズの入力を求めるダイアログが表示されます。強固なパスフレーズを選択してくださいが、覚えやすいものもあります。をクリック OK し、キーが作成されます。

警告

パスフレーズを忘れると、データを復号できなくなります。

GPG キー ID を見つけるには、新たに作成された 鍵の横にあるキー ID 列を参照してください。ほとんどの場合、鍵 ID を要求する場合は、にあるよう 0x にキー ID の前にが追加され 0x6789ABCDます。秘密鍵のバックアップを作成し、安全な場所に保存する必要があります。

### 3.5.2. KDE での GPG キーの作成

KDE で GPG キーを作成するには、以下の手順に従います。

1. メインメニューから KGpg プログラムを起動するには、を選択し Applications → Utilities → Encryption Toolます。以前に KGpg を使用したことがない場合、プログラムは、独自の GPG キーペアを作成するプロセスを開始します。

2. 新しいキーペアの作成を求めるダイアログボックスが表示されます。名前、メールアドレス、およびオプションのコメントを入力します。キーの有効期限や、キーの強度（ビットの数）およびアルゴリズムを選択することもできます。

3. 次のダイアログボックスにパスフレーズを入力します。この時点で、キーがメイン KGpg ウィンドウに表示されます。

警告

パスフレーズを忘れると、データを復号できなくなります。

GPG キー ID を見つけるには、新たに作成された 鍵の横にあるキー ID 列を参照してください。ほとんどの場合、鍵 ID を要求する場合は、にあるよう 0x にキー ID の前にが追加され 0x6789ABCDます。秘密鍵のバックアップを作成し、安全な場所に保存する必要があります。

### 3.5.3. コマンドラインで GPG 鍵の作成

1. 以下のシェルコマンドを使用します。

```
~]$ gpg2 --gen-key
```

このコマンドは、公開鍵と秘密鍵で構成されるキーペアを生成します。その他のユーザーは公開鍵を使用して通信を認証または復号化します。特にメーリングリストなど、お客様から正式な通信を受信したい場合に、公開鍵を可能な限り広く配布します。

2. 一連のプロンプトにより、プロセスが実行されます。**Enter** キーを押して、必要であればデフォルト値を割り当てます。最初のプロンプトでは、希望する鍵の選択が求められます。

```
Please select what kind of key you want:
(1) RSA and RSA (default)
(2) DSA and Elgamal
(3) DSA (sign only)
(4) RSA (sign only)
Your selection?
```

ほとんどの場合で、デフォルト値が正しい選択になります。**RSA/RSA** キーを使用すると、通信に署名するだけでなく、ファイルを暗号化できます。

3. キーサイズを選択します。

```
RSA keys may be between 1024 and 4096 bits long.
What keysize do you want? (2048)
```

ここでも、ほとんどのユーザーにはデフォルトの **2048** で十分で、セキュリティーレベルは非常に強固です。

4. キーの有効期限が切れるタイミングを選択します。デフォルトを使用する代わりに有効期限を選択することが推奨され **none**ます。たとえば、キーのメールアドレスが無効になると、その公開鍵の使用を停止するように、有効期限が他のユーザーに通知されます。

```
Please specify how long the key should be valid.
0 = key does not expire
d = key expires in n days
w = key expires in n weeks
```

```
m = key expires in n months
y = key expires in n years
key is valid for? (0)
```

**1y の値を入力して（例：）、キーは 1 年間有効になります。（設定を変更する場合は、キーの生成後にこの有効期限を変更できます。）**

5. **gpg2 アプリケーションが署名情報を要求する前に、以下のプロンプトが表示されます。**

```
Is this correct (y/N)?
```

**y を入力してプロセスを完了します。**

6. **GPG キーの名前とメールアドレスを入力します。このプロセスは、実際の個人としてユーザーを認証することにあります。このため、実際の名前を含めます。偽のメールアドレスを選択すると、他のユーザーが公開鍵を見つけることがより困難になります。これにより、通信の認証が困難になります。この GPG キーを使用してメーリングリストで自己操作を行います。たとえば、そのリストで使用するメールアドレスを入力します。**

**comment フィールドを使用してエイリアスやその他の情報を追加します。（一部のユーザーはさまざまな目的で異なるキーを使用し、「Office」や「オープンソースプロジェクト」などのコメントで各キーを特定します。）**

7. **確認プロンプトで文字を入力し、すべてのエントリーが正しい場合は続行 O するか、他のオプションを使用して問題を解決します。最後に、秘密鍵のパスフレーズを入力します。gpg2 プログラムはパスフレーズの入力を 2 回入力して、エラーが発生しないように要求します。**

8. **最後に、鍵をできるだけ一意にするランダムなデータを gpg2 生成します。マウスを移動し、ランダムな鍵を入力するか、このステップ中にシステムに他のタスクを実行して処理を迅速化します。この手順が完了すると、鍵が完了し、使用できる状態になります。**

```
pub  1024D/1B2AFA1C 2005-03-31 John Q. Doe <jqdoe@example.com>
Key fingerprint = 117C FE83 22EA B843 3E86  6486 4320 545E 1B2A FA1C
sub  1024g/CEA4B22E 2005-03-31 [expires: 2006-03-31]
```

9. **キーフィンガープリントは、キーの短い「署名」です。これにより、改ざんなしで、実際に公開鍵を受信したことを他のユーザーに確認することができます。このフィンガープリントを書き留める必要はありません。フィンガープリントをいつでも表示するには、メールアドレスを置き換えて、次のコマンドを使用します。**

```
~]$ gpg2 --fingerprint jqdoe@example.com
```

~]# gpg --fingerprint [user's e-mail address]

「GPG key ID」は、公開鍵を識別する 8 16 進法で構成されます。上記の例では、GPG キー ID はです 1B2AFA1C。ほとんどの場合、鍵 ID を要求する場合は、にあるよう 0x にキー ID の前にが追加され 0x6789ABCDます。

**警告**

パスフレーズを忘れると、鍵は使用できないため、その鍵を使用して暗号化したデータはすべて失われます。

### 3.5.4. 公開鍵の暗号化について

1. 重要 - 公開鍵の暗号化
2. HowStuffWorks - Encryption

## 3.6. STUNNEL の使用

stunnel プログラムは、クライアントとサーバー間の暗号化ラッパーです。設定ファイルで指定されたポートでリッスンし、クライアントとの通信を暗号化し、通常のポートでリッスンする元のデーモンにデータを転送します。これにより、それ自体がいずれの暗号化にも対応していないサービスのセキュリティーを保護することや、セキュリティー上の理由から回避したい暗号化を使用するサービスのセキュリティーを向上させることができます(CVE-2014-3566)。設定により SSLv3 を無効にできないコンポーネントの解決は、「Resolution for POODLE SSLv3.0 vulnerability(CVE-2014-3566)」を参照してください。2.4.39 よりも古いOpenLDAP (Red Hat Enterprise Linux 6.6)および CUPS は、独自の設定で SSL を無効にする方法を提供しないコンポーネントの例です。

### 3.6.1. stunnel のインストール

root で以下のコマンドを実行して stunnel パッケージをインストールします。

```
~]# yum install stunnel
```

### 3.6.2. stunnel を TLS Wrapper として設定

**stunnel を設定するには、以下の手順に従います。**

1. **stunnel に有効な証明書が必要になりますが、それを使用するサービスに関係なく必要です。適切な証明書がない場合は、*認証局* に適用して取得するか、自己署名のセマンティクスを作成できます。**

**警告**

**実稼働環境で実行しているサーバーには、認証局が署名した証明書を常に使用してください。自己署名証明書は、テスト目的またはプライベートネットワークにのみ適しています。**

**stunnel 用の自己署名証明書を作成するには、/etc/pki/tls/certs/ ディレクトリーを入力し、root で以下のコマンドを入力します。**

```
certs]# make stunnel.pem
```

**すべての質問に回答して、プロセスを完了します。**

2. **証明書がある場合、stunnel 用の設定ファイルを作成します。これは、すべての行でオプションまたはサービス定義の開始を指定するテキストファイルです。また、ファイルにコメントと空の行を維持し、その信頼性を向上させることもできます。コメントはセミコロンで始まる場合です。**

**stunnel RPM パッケージには、設定ファイルを保存できる /etc/stunnel/ ディレクトリーが含まれています。stunnel には特別なファイル名または拡張子は必要ありませんが、を使用し /etc/stunnel/stunnel.confます。以下のコンテンツは、stunnel を TLS ラッパーとして設定します。**

```
cert = /etc/pki/tls/certs/stunnel.pem
; Allow only TLS, thus avoiding SSL
sslVersion = TLSv1
chroot = /var/run/stunnel
setuid = nobody
```

```
setgid = nobody
pid = /stunnel.pid
socket = l:TCP_NODELAY=1
socket = r:TCP_NODELAY=1

[service_name]
accept = port
connect = port
TIMEOUTclose = 0
```

以下の行にが含まれる行を **sslVersion = TLSv1** 置き換えると、**SSL** を回避することができます。

```
options = NO_SSLv2
options = NO_SSLv3
```

オプションの目的は以下のとおりです。

- **CERT : 証明書へのパス**
- **sslVersion : SSL のバージョン - SSL と TLS は 2 つの独立した暗号化プロトコルですが、TLS ここで使用することができます。**
- **chroot : stunnel プロセスが実行され、セキュリティーを強化するために変更したルートディレクトリー。**
- **setuid,setgid - stunnel プロセス nobody が実行するユーザーおよびグループは、制限されたシステムアカウントです。**
- **pid : stunnel がプロセス ID を保存するファイル（と相対的に）。 chroot**
- **Socket : ローカルおよびリモートのソケットオプション - この場合 *は Nagle のアルゴリズム* を無効にしてネットワークレイテンシーを向上させます。**
- **[*service_name*] : サービス定義の開始 - この行で使用されているオプションは指定のサービスにのみ適用されますが、上記のオプションは stunnel にグローバルに影響します。**

- accept : リッスンするポート

- connect : 接続するポート。これは、セキュリティー保護するサービスに使用するポートでなければなりません。

- TIMEOUTclose : クライアントから *close_notify* アラートを待機する秒数。stunnel が全く待機しないように 0 指示します。

- options : OpenSSL ライブラリーオプション

> **例3.1 OpenLDAP のセキュリティー保護**
>
> stunnel を 2.4.39 よりも古い OpenLDAP の TLS ラッパーとして設定するには、以下の値を使用します。
>
> ```
> [openldap]
> accept = 636
> connect = 389
> ```
>
> 636 はセキュアな LDAP の標準ポートです 389。は、OpenLDAP デーモンがリッスンするポートです。

> **例3.2 CUPS のセキュリティー保護**
>
> 同様に、stunnel を CUPS の TLS ラッパーとして設定するには、以下の値を使用します。
>
> ```
> [cups]
> accept = 632
> connect = 631
> ```
>
> の代わりに 632、任意の空きポートを使用できます。631 は、CUPS が通常使用するポートです。

3. chroot ディレクトリーを作成し、setuid オプションで指定されるユーザーに書き込みアクセス権限を付与します。これを行うには、root で以下のコマンドを実行します。

```
~]# mkdir /var/run/stunnel
~]# chown nobody:nobody /var/run/stunnel
```

これにより、**stunnel** は **PID** ファイルを作成します。

4. 新しいポートへのアクセスを許可しないファイアウォール設定を使用している場合は、適切に変更します。詳細 「その他のポート」 は 「ファイアウォール」 を参照してください。

5. 設定ファイルと **chroot** ディレクトリーを作成し、指定したポートにアクセスできることを確認すると、**stunnel** の使用を開始する準備が整います。

### 3.6.3. stunnel の開始、停止、および再起動

**stunnel** を起動するには、**root** で以下のコマンドを実行します。

```
~]# stunnel /etc/stunnel/stunnel.conf
```

デフォルトでは、**stunnel** は **/var/log/secure** を使用して出力をログに記録します。

**stunnel** を終了するには、**root** で以下のコマンドを実行してプロセスを強制終了します。

```
~]# kill `cat /var/run/stunnel/stunnel.pid`
```

**stunnel** の実行中に設定ファイルを編集する場合は、**stunnel** を終了し、変更を有効にするために再び起動します。

## 3.7. TLS 設定のハードニング

**TLS (Transport Layer Security)**は、ネットワーク通信のセキュリティーを保護するために使用される暗号化プロトコルです。優先する *鍵交換プロトコル*、*認証方法*、および *暗号化アルゴリズム* を設定してシステムのセキュリティー設定を強化する場合は、サポートされるクライアントの範囲が広ければ広いほど、セキュリティーレベルが低くなることを認識しておく必要があります。反対に、セキュリティー設定によりクライアントとの互換性が制限され、システムからロックアウトされるユーザーが少なくなることがあります。可能な限り厳密な設定を目指し、互換性に必要な場合に限り、設定を緩めるようにしてください。

ほとんどのデプロイメントでは、Red Hat Enterprise Linux に含まれるライブラリーが提供するデフォルト設定が十分に安全であることに注意してください。TLS 実装は、可能な場合は安全なアルゴリズムを使用しますが、レガシーのクライアントまたはサーバーへの接続は妨げません。セキュアなアルゴリズムまたはプロトコルをサポートしないレガシーなクライアントまたはサーバーが、接続が期待できない、または許可されないレガシーなセキュリティー要件がある環境では、このセクションで説明する強化された設定を適用します。

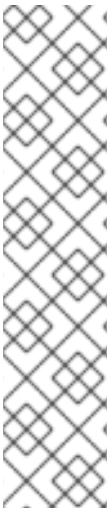
### 3.7.1. 有効にするアルゴリズムの選択

選択および設定が必要なコンポーネントが複数あります。以下のそれぞれは、設定の堅牢性（つまり、クライアントでのサポートレベル）や、ソリューションがシステムに持つ計算要件に直接影響します。

プロトコルのバージョン

最新バージョンの TLS は、最高のセキュリティーメカニズムを提供します。古いバージョンの TLS（または SSL）のサポートが含まれるようなような理由がない限り、システムは最新バージョンの TLS のみを使用して接続をネゴシエートできるようにし ます。

SSL バージョン 2 または 3 を使用するネゴシエーションを許可しないでください。これらのバージョンにはいずれも重大なセキュリティー脆弱性があります。TLS バージョン 1.0 以降を使用するネゴシエーションのみを許可します。TLS 1.2 の現行バージョンは常に推奨する必要があります。



注記

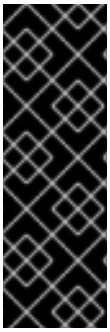
現在、TLS の全バージョンのセキュリティーは、TLS 拡張機能の使用、特定の暗号（下記参照）の使用などによって異なることに注意してください。すべての TLS 接続ピアは、セキュアな再ネゴシエーションインデックス(RFC 5746)を実装する必要があります。圧縮をサポートしない。また、CBCモード暗号（Lucky Thir 攻撃）に対するタイミング攻撃の緩和策を実装する必要があります。TLS v1.0 クライアントは、追加のレコード分割（BEAST 攻撃に対する回避策）を実装する必要があります。TLS v1.2 は、*認証された暗号化と関連するデータ*（）をサポートします。AEAD） AES-GCM、AES-CCM、Camellee- GCM などのモード暗号。既知の問題はありません。上記の軽減策はすべて、Red Hat Enterprise Linux に含まれる暗号化ライブラリーに実装されています。

プロトコルバージョンの概要と推奨される使用方法は 表3.1「プロトコルのバージョン」 を参照してください。

表3.1 プロトコルのバージョン

| プロトコルのバージョン | 使用に関する推奨事項 |
| --- | --- |
| SSL v2 | 使用しないでください。深刻なセキュリティー上の脆弱性があります。 |
| SSL v3 | 使用しないでください。深刻なセキュリティー上の脆弱性があります。 |
| TLS v1.0 | 必要に応じて相互運用性の目的で使用します。相互運用性を保証する方法で軽減できない既知の問題があるため、デフォルトでは軽減策が有効になっていません。最新の暗号スイートには対応しません。 |
| TLS v1.1 | 必要に応じて相互運用性の目的で使用します。既知の問題はありませんが、Red Hat Enterprise Linux のすべての TLS 実装に含まれるプロトコルの修正に依存します。最新の暗号スイートには対応しません。 |
| TLS v1.2 | 推奨されるバージョン。最新の AEAD 暗号スイートに対応します。 |

Red Hat Enterprise Linux の一部のコンポーネントは、TLS v 1.1 または v 1.2 のサポートを提供しますが、TLS v 1.0 を使用するように設定されています。これは、最新バージョンの TLS をサポートしない外部サービスとの最も高いレベルの相互運用性を実現することが目的です。相互運用性の要件に応じて、利用可能な TLS の最大値を有効にします。



重要

SSL v3 の使用は推奨されません。ただし、セキュアでないと見なされて一般的に使用できない場合は、SSL v3 を有効にしたままにする必要があります。暗号化に対応していないサービスを使用している場合や、古くなった暗号化モードのみを使用するサービスを使用する場合でも、s stunnel を使用して通信を安全に暗号化する 「stunnel の使用」 方法はを参照してください。

128 ビット未満のセキュリティーしか提供しない暗号化スイートでは直ちにセキュリティーが保護されなくなるというわけではありませんが、使用できる期間が短いため考慮すべきではありません。128 ビット以上のセキュリティーを使用するアルゴリズムは、少なくとも数年間は改ざんできないことが期待されるため、強く推奨されます。3DES 暗号は 168 ビットを使用していることを公開していますが、実際には 112 ビットのセキュリティーを提供していることに注意してください。

PFS(Perfect *)フォワード Secrecy(PFS)* をサポートする暗号スイートを常に優先します。これにより、サーバーキーが危険にさらされた場合でも、暗号化されたデータの機密性が確保されます。これにより、高速 RSA 鍵交換は除外されますが、ECDHE および DHE を使用できます。この 2 つでは、ECDHE の方が高速であるため、推奨される選択肢となります。

ECDSA 証明書で ECDHE 鍵交換を使用すると、トランザクションは純粋な RSA 鍵交換よりもさらに高速になります。レガシークライアントに対応するには、サーバー上に証明書と鍵のペアを 2 つ（新しいクライアント用の ECDSA 鍵 と、レガシー用の RSA 鍵）インストールできます。

公開鍵の長さ

RSA 鍵を使用する場合は、SHA-256 以上で署名された鍵の長さが 3072 ビット以上推奨されます。これは、実際に 128 ビットのセキュリティーに対して十分な大きさです。

警告

システムのセキュリティーレベルは、チェーン内で最も弱いリンクと同じであることに注意してください。たとえば、強力な暗号化だけではすぐれたセキュリティーは保証されません。鍵と証明書も同様に重要で、*認証局 (CA)*が鍵の署名に使用するハッシュ機能と鍵も重要になります。

### 3.7.2. TLS の実装の使用

Red Hat Enterprise Linux には、TLS のフル機能実装が同梱されています。このセクションで

は、OpenSSL および GnuTLSの設定を説明します。個別 「特定のアプリケーションの設定」 のアプリケーションで TLS サポートを設定する方法は、を参照してください。

利用可能な TLS 実装は、TLSでセキュア化された通信の確立および使用時に統合されたすべての要素を定義するさまざまな *暗号スイート* に対応します。

別の実装に含まれるツールを使用して、の推奨事項を検討しながら、ユースケースに最適なセキュリティーを提供する暗号スイートを一覧表示および指定し 「有効にするアルゴリズムの選択」 ます。生成される暗号スイートを使用すると、個々のアプリケーションが接続をネゴシエートし、セキュアな接続をネゴシエートする方法を設定できます。

重要

使用する TLS 実装の更新またはアップグレード、またはその実装を使用するアプリケーションの更新またはアップグレード後に必ず設定を確認してください。新しいバージョンでは、有効にしたくない新しい暗号スイートが導入され、現在の設定が無効にならない可能性があります。

#### 3.7.2.1. OpenSSL での暗号スイートの使用

OpenSSL は、SSL プロトコルおよび TLS プロトコルをサポートするツールキットおよび暗号ライブラリーです。Red Hat Enterprise Linux では、設定ファイルがにあり /etc/pki/tls/openssl.cnfます。この設定ファイルの形式は、に記載されています。 config(1).

OpenSSL のインストールでサポートされる暗号スイートの一覧を取得するには、以下のように ciphers サブ openssl コマンドとともにコマンドを実行します。

```
~]$ openssl ciphers -v 'ALL:COMPLEMENTOFALL'
```

出力を絞り込むために、他のパラメーター（ OpenSSL ドキュメントの *暗号文字列* および *キーワード* として参照される）を ciphers サブコマンドに渡します。特別なキーワードは、特定の条件を満たすスイートのみを一覧表示するために使用できます。たとえば、として定義されたスイートのみを一覧表示するには、 HIGH グループ、以下のコマンドを使用します。

```
~]$ openssl ciphers -v 'HIGH'
```

以下を参照してください。 暗号化(1) 利用可能なキーワードおよび暗号文字列の一覧の man ページです。

で説明されている推奨事項を満たす暗号スイートの一覧を取得するには「有効にするアルゴリズムの選択」、以下のようなコマンドを使用します。

```
~]$ openssl ciphers -v 'kEECDH+aECDSA+AES:kEECDH+AES+aRSA:kEDH+aRSA+AES' |
column -t
ECDHE-ECDSA-AES256-GCM-SHA384  TLSv1.2  Kx=ECDH  Au=ECDSA  Enc=AESGCM(256)
Mac=AEAD
ECDHE-ECDSA-AES256-SHA384      TLSv1.2  Kx=ECDH  Au=ECDSA  Enc=AES(256)
Mac=SHA384
ECDHE-ECDSA-AES256-SHA         SSLv3    Kx=ECDH  Au=ECDSA  Enc=AES(256)    Mac=SHA1
ECDHE-ECDSA-AES128-GCM-SHA256  TLSv1.2  Kx=ECDH  Au=ECDSA  Enc=AESGCM(128)
Mac=AEAD
ECDHE-ECDSA-AES128-SHA256      TLSv1.2  Kx=ECDH  Au=ECDSA  Enc=AES(128)
Mac=SHA256
ECDHE-ECDSA-AES128-SHA         SSLv3    Kx=ECDH  Au=ECDSA  Enc=AES(128)    Mac=SHA1
ECDHE-RSA-AES256-GCM-SHA384    TLSv1.2  Kx=ECDH  Au=RSA    Enc=AESGCM(256)
Mac=AEAD
ECDHE-RSA-AES256-SHA384        TLSv1.2  Kx=ECDH  Au=RSA    Enc=AES(256)    Mac=SHA384
ECDHE-RSA-AES256-SHA           SSLv3    Kx=ECDH  Au=RSA    Enc=AES(256)    Mac=SHA1
ECDHE-RSA-AES128-GCM-SHA256    TLSv1.2  Kx=ECDH  Au=RSA    Enc=AESGCM(128)
Mac=AEAD
ECDHE-RSA-AES128-SHA256        TLSv1.2  Kx=ECDH  Au=RSA    Enc=AES(128)    Mac=SHA256
ECDHE-RSA-AES128-SHA           SSLv3    Kx=ECDH  Au=RSA    Enc=AES(128)    Mac=SHA1
DHE-RSA-AES256-GCM-SHA384      TLSv1.2  Kx=DH    Au=RSA    Enc=AESGCM(256)
Mac=AEAD
DHE-RSA-AES256-SHA256          TLSv1.2  Kx=DH    Au=RSA    Enc=AES(256)    Mac=SHA256
DHE-RSA-AES256-SHA             SSLv3    Kx=DH    Au=RSA    Enc=AES(256)    Mac=SHA1
DHE-RSA-AES128-GCM-SHA256      TLSv1.2  Kx=DH    Au=RSA    Enc=AESGCM(128)
Mac=AEAD
DHE-RSA-AES128-SHA256          TLSv1.2  Kx=DH    Au=RSA    Enc=AES(128)    Mac=SHA256
DHE-RSA-AES128-SHA             SSLv3    Kx=DH    Au=RSA    Enc=AES(128)    Mac=SHA1
```

上記のコマンドは、すべての安全ではない暗号を省略します。これにより、一時的 *な楕円曲線 Diffie-Hellman*鍵交換と **ECDSA** 暗号が優先され、**RSA** 鍵交換は省略されます（これにより、フォワード暗号が確保されます）。

これはより厳密な設定であるため、実際のシナリオで条件を緩和し、幅広いクライアントとの互換性を確保する必要がある場合があります。

#### 3.7.2.2. GnuTLS での暗号スイートの使用

**GnuTLS** は、**SSL** プロトコルおよび **TLS** プロトコル、および関連するテクノロジーを実装する通信ライブラリーです。

**注記**

**Red Hat Enterprise Linux での GnuTLS インストールでは、ほとんどのユースケースに対して十分なセキュリティーを提供する最適なデフォルト設定値が提供されます。特別なセキュリティー要件を満たさない限り、提供されるデフォルトを使用することが推奨されます。**

**-l（または --list）オプションを指定して gnutls-cli コマンドを使用して、対応している暗号スイートの一覧を表示します。**

```
~]$ gnutls-cli -l
```

**-l オプションで表示される暗号スイートの一覧を絞り込むには、1 つ以上のパラメーター（GnuTLS ドキュメントの *優先度文字列* および *キーワード*）を --priority オプションに渡します。利用可能な優先度文字列の一覧は、http://www.gnutls.org/manual/gnutls.html#Priority-Strings で GnuTLS のドキュメント を 参照してください。たとえば、以下のコマンドを発行して、最低 128 ビットのセキュリティーを提供する暗号スイートの一覧を取得します。**

```
~]$ gnutls-cli --priority SECURE128 -l
```

**で説明されている推奨事項を満たす暗号スイートの一覧を取得するには 「有効にするアルゴリズムの選択」、以下のようなコマンドを使用します。**

```
~]$ gnutls-cli --priority SECURE256:+SECURE128:-VERS-TLS-ALL:+VERS-TLS1.2:-RSA:-DHE-
DSS:-CAMELLIA-128-CBC:-CAMELLIA-256-CBC -l
Cipher suites for SECURE256:+SECURE128:-VERS-TLS-ALL:+VERS-TLS1.2:-RSA:-DHE-DSS:-
CAMELLIA-128-CBC:-CAMELLIA-256-CBC
TLS_ECDHE_ECDSA_AES_256_GCM_SHA384                0xc0, 0x2c      TLS1.2
TLS_ECDHE_ECDSA_AES_256_CBC_SHA384                0xc0, 0x24      TLS1.2
TLS_ECDHE_ECDSA_AES_256_CBC_SHA1                  0xc0, 0x0a      SSL3.0
TLS_ECDHE_ECDSA_AES_128_GCM_SHA256                0xc0, 0x2b      TLS1.2
TLS_ECDHE_ECDSA_AES_128_CBC_SHA256                0xc0, 0x23      TLS1.2
TLS_ECDHE_ECDSA_AES_128_CBC_SHA1                  0xc0, 0x09      SSL3.0
TLS_ECDHE_RSA_AES_256_GCM_SHA384                  0xc0, 0x30      TLS1.2
TLS_ECDHE_RSA_AES_256_CBC_SHA1                    0xc0, 0x14      SSL3.0
TLS_ECDHE_RSA_AES_128_GCM_SHA256                  0xc0, 0x2f      TLS1.2
TLS_ECDHE_RSA_AES_128_CBC_SHA256                  0xc0, 0x27      TLS1.2
TLS_ECDHE_RSA_AES_128_CBC_SHA1                    0xc0, 0x13      SSL3.0
TLS_DHE_RSA_AES_256_CBC_SHA256                    0x00, 0x6b      TLS1.2
TLS_DHE_RSA_AES_256_CBC_SHA1                      0x00, 0x39      SSL3.0
TLS_DHE_RSA_AES_128_GCM_SHA256                    0x00, 0x9e      TLS1.2
TLS_DHE_RSA_AES_128_CBC_SHA256                    0x00, 0x67      TLS1.2
TLS_DHE_RSA_AES_128_CBC_SHA1                      0x00, 0x33      SSL3.0

Certificate types: CTYPE-X.509
Protocols: VERS-TLS1.2
Compression: COMP-NULL
```

```
Elliptic curves: CURVE-SECP384R1, CURVE-SECP521R1, CURVE-SECP256R1
PK-signatures: SIGN-RSA-SHA384, SIGN-ECDSA-SHA384, SIGN-RSA-SHA512, SIGN-ECDSA-
SHA512, SIGN-RSA-SHA256, SIGN-DSA-SHA256, SIGN-ECDSA-SHA256
```

上記のコマンドは、出力を 128 ビット以上のセキュリティーで暗号に制限し、強力なセキュリティーを優先します。また、RSA 鍵交換と DSS 認証も禁止します。

これはより厳密な設定であるため、実際のシナリオで条件を緩和し、幅広いクライアントとの互換性を確保する必要がある場合があります。

### 3.7.3. 特定のアプリケーションの設定

アプリケーションによって TLS の独自の設定メカニズムが提供されます。本セクションでは、最も一般的に使用されるサーバーアプリケーションが使用する TLS関連の設定ファイルを説明し、一般的な設定の例を説明します。

いずれの設定を選択しても、サーバーアプリケーションが強制的に *サーバー側が指定した順序* で暗号を利用することを確認し、使用される暗号化スイートの選択がサーバでの設定順に行われるように設定してください。

#### 3.7.3.1. Apache HTTP サーバーの設定

Apache HTTP Server は、TLS のニーズに OpenSSL ライブラリーと NSS ライブラリーの両方を使用できます。TLS ライブラリーの選択に応じて、mod_ssl モジュールまたは mod_ nss モジュール（詳細なパッケージで提供される）をインストールする必要があります。たとえば、OpenSSL mod_ssl モジュールを提供するパッケージをインストールするには、root で以下のコマンドを実行します。

```
~]# yum install mod_ssl
```

mod_ssl パッケージは、/etc/httpd/conf.d/ssl.conf 設定ファイルをインストールします。このファイルは、Apache HTTP Server の TLS関連の設定を変更するために使用できます。同様に、mod_nss パッケージは /etc/httpd/conf.d/nss.conf 設定ファイルをインストールします。

httpd-manual パッケージをインストールして、TLS 設定を含む Apache HTTP Server の完全なドキュメントを取得します。/etc/httpd/conf.d/ssl.conf 設定ファイルで利用可能なディレクティブは、で詳細に説明されてい /usr/share/httpd/manual/mod/mod_ssl.html ます。各種設定の例は、にあり /usr/share/httpd/manual/ssl/ssl_howto.html ます。

設定 /etc/httpd/conf.d/ssl.conf ファイルの設定を修正する場合は、少なくとも以下の 3 つのディレ

クティブを確認してください。

**SSLProtocol**

このディレクティブを使用して、許可する TLS （または SSL）のバージョンを指定します。

**SSLCipherSuite**

優先する暗号化スイートを指定する、もしくは許可しないスイートを無効にするディレクティブです。

**SSLHonorCipherOrder**

コメントを解除して、このディレクティブを on に設定して、接続先のクライアントが指定した暗号の順序に従います。

以下に例を示します。

```
SSLProtocol all -SSLv2 -SSLv3
SSLCipherSuite HIGH:!aNULL:!MD5
SSLHonorCipherOrder on
```

上記の設定は最小で、で説明されている推奨事項に従って大幅に強化でき 「有効にするアルゴリズムの選択」 ます。

mod_nss モジュールを設定して使用するには、/etc/httpd/conf.d/nss.conf 設定ファイルを変更します。mod_nss モジュールは mod_ssl から派生するものであるため、設定ファイルの構造や利用可能なディレクティブなど、多くの機能を共有するためです。mod_nss ディレクティブには、では NSS なくの接頭辞が指定されてい SSLます。mod_nss に 適用できない mod_ ssl 設定ディレクティブのリストなど、mod_ nss に関する情報は https://git.fedorahosted.org/cgit/mod_nss.git/plain/docs/mod_nss.html を参照してください。

### 3.7.4. 追加情報

TLS 設定と関連トピックの詳細は、以下に挙げるリソースを参照してください。

インストールされているドキュメント

- config(1) : /etc/ssl/openssl.conf 設定ファイルの形式を説明します。

- 暗号化(1) : 利用可能な OpenSSL キーワードおよび暗号文字列の一覧が含まれます。

- /usr/share/httpd/manual/mod/mod_ssl.html : Apache HTTP Server の mod_ssl モジュールが使用する /etc/httpd/conf.d/ssl.conf 設定ファイルで利用可能なディレクティブの詳細を説明します。

- /usr/share/httpd/manual/ssl/ssl_howto.html - Apache HTTP Server の mod_ssl モジュールが使用する /etc/httpd/conf.d/ssl.conf 設定ファイルの実際の設定例を取り上げます。

オンラインドキュメント

- Red Hat Enterprise Linux 6 Security-Enhanced Linux - Red Hat Enterprise Linux 6 の『Security-Enhanced Linux』 ガイドでは、SELinux の基本的な原則について説明しています。

- http://tools.ietf.org/html/draft-ietf-uta-tls-bcp-00: TLS および DTLS をセキュアにする推奨設定

# 第4章 情報セキュリティーの一般的な原則

以下の一般的な原則は、優れたセキュリティープラクティスの概要を示しています。

- 中間者攻撃や盗聴を防ぐために、ネットワーク経由で送信されるすべてのデータを暗号化します。パスワードなどの認証情報を暗号化することが重要です。

- インストールして実行しているサービス量を最小限に抑えます。

- セキュリティー強化ソフトウェアおよびツール（Mandatory Access Control(MAC)向け Security-Enhanced Linux(SELinux)、パケットフィルタリング(firewall)の Netfilter iptables、ファイルの暗号化に GNU Privacy Guard(GPG)など）を使用します。

- 可能な場合は、各ネットワークサービスを別のシステムで実行し、危険にさらされたサービスを使用して他のサービスを危険にさらすリスクを最小限に抑えます。

- ユーザーアカウントを維持します。強力なパスワードポリシーを作成して強制します。未使用のユーザーアカウントを削除します。

- システムおよびアプリケーションログを定期的に確認します。デフォルトでは、セキュリティー関連のシステムログは /var/log/secure およびに書き込まれ /var/log/audit/audit.logます。注記： 専用のログサーバーへのログを送信すると、攻撃者は検出を避けるためにローカルログを簡単に変更できないようにします。

- 絶対的に必要でない限り、root ユーザーとしてログインしないでください。必要に応じて、管理者は sudo を使用して root としてコマンドを実行することが推奨されます。を実行できるユーザーは、で指定 sudo されてい /etc/sudoersます。visudo ユーティリティーを使用してを編集し /etc/sudoersます。

# 第5章 セキュアなインストール

セキュリティーの開始は、CD または DVD をディスクドライブに初めて挿入し、Red Hat Enterprise Linux をインストールします。最初からシステムのセキュリティーを設定することで、追加のセキュリティー設定を実装することがより簡単になります。

## 5.1. ディスクパーティション

Red Hat は /boot、、、、および用に別々のパーティションを作成することを推奨 / /home /tmp/し /var/tmp/ます。ルートパーティション(/)が破損すると、データが完全に失われます。異なるパーティションを使用すると、データはより保護されます。このパーティションをターゲットにして、頻繁にバックアップを作成することもできます。各パーティションの目的は異なり、それぞれのパーティションに対応します。

/boot : このパーティションは、システムの起動時にシステムが最初に読み込むパーティションです。Red Hat Enterprise Linux でシステムを起動するのに使用されるブートローダーとカーネルイメージはこのパーティションに保存されます。このパーティションは暗号化しないでください。このパーティションがに含まれ、/ そのパーティションが暗号化されているか、使用できない場合は、システムを起動できなくなります。

/home : 別のパーティションでは / なくユーザーデータ(/home)を保存すると、パーティションが満杯になり、オペレーティングシステムが不安定になる可能性があります。また、システムを次のバージョンの Red Hat Enterprise Linux にアップグレードする場合は、インストール時に上書きされないため、/home パーティションにデータを保持できるので、これは非常に簡単です。

/tmp および /var/tmp/ : /tmp および /var/tmp/ ディレクトリーはいずれも、長期保存の必要がないデータを保管するために使用されます。ただし、このいずれかのディレクトリーでデータがあふれると、ストレージ領域がすべて使用することができます。このディレクトリーが内に保存されると、システムが不安定になり /、クラッシュする可能性があります。そのため、このディレクトリーは個別のパーティションに移動することが推奨されます。

## 5.2. LUKS パーティション暗号化の使用

インストールプロセス時に、パーティションを暗号化するオプションがユーザーに提示されます。ユーザーは、パスフレーズを入力する必要があります。このパスフレーズは、パーティションのデータを保護するために使用されるバルク暗号鍵を解除する鍵として使用されます。

# 第6章 ソフトウェアメンテナンス

セキュアなシステムを維持するには、ソフトウェアのメンテナンスが非常に重要です。攻撃者が既知のホールを使用してシステムに侵入しないように、利用可能になった直後にソフトウェアにパッチを適用することが必須です。

## 6.1. 最小ソフトウェアのインストール

コンピューターの各ソフトウェアに脆弱性が含まれる可能性があるため、インストールするパッケージのみをインストールすることが推奨されます。インストールを DVD メディアから行う場合は、インストール時にインストールするパッケージを正確に選択することができます。別のパッケージが必要になる場合は、後でいつでもシステムに追加できます。

最小インストールの詳細は、『『Red Hat Enterprise Linux 6 インストールガイド』の「『パッケージグループの選択」』 セクションを参照してください』。--nobase オプションを使用すると、キックスタートファイルを介して最小限のインストールを実行することもできます。詳細は、『『Red Hat Enterprise Linux 6 インストールガイド』の「『パッケージの選択」』 セクションを参照してください』。

## 6.2. セキュリティー更新のプランニングおよび設定

すべてのソフトウェアにバグが含まれています。多くの場合、このようなバグは、システムを悪意のあるユーザーに公開できる脆弱性が発生する可能性があります。パッチが適用されないシステムは、コンピューターの侵入の一般的な原因です。セキュリティーパッチを適時にインストールして、これらの脆弱性を悪用しないように計画している必要があります。

ホームユーザーの場合は、できるだけ早くセキュリティー更新をインストールする必要があります。セキュリティー更新の自動インストールを設定すると、記憶する必要がなくなりますが、設定またはシステム上の他のソフトウェアとの競合が発生するリスクが若干あります。

ビジネスまたは上級のホームユーザーの場合、インストール用にセキュリティー更新をテストし、スケジュールする必要があります。パッチリリースとそのシステムへのインストールの間の時間帯に、システムを保護するには、追加の制御を使用する必要があります。このコントロールは、脆弱性の正確な脆弱性によって異なりますが、ファイアウォールルールの追加、外部ファイアウォールの使用、またはソフトウェア設定の変更などが含まれます。

## 6.3. 自動更新の調整

Red Hat Enterprise Linux は、すべての更新を日次スケジュールに適用するように設定されています。システムの更新のインストール方法を変更する場合は、ソフトウェアアップデート設定を使用して

これを行う必要があります。スケジュール、適用する更新のタイプを変更したり、利用可能な更新を通知したりできます。

GNOME では、更新の制御はにあります System → Preferences → Software Updates。KDE では、.. にあります Applications → Settings → Software Updates。

### 6.4. 既知のリポジトリーからの署名パッケージのインストール

ソフトウェアパッケージは、リポジトリーにより公開されます。既知のリポジトリーはすべてパッケージの署名をサポートします。パッケージ署名は公開鍵技術を使用して、署名の適用以降、リポジトリーによって公開されたパッケージが変更されていないことを証明します。これにより、パッケージの作成後、ダウンロードする前に、悪意のある変更が加えられた可能性のあるソフトウェアのインストールに対する保護がいくつか提供されます。

多くのリポジトリー、信頼できないリポジトリー、または未署名のパッケージを持つリポジトリーを使用する場合に、システムに悪意のあるコードまたは脆弱なコードを導入するリスクが高まります。リポジトリーを yum/software 更新に追加する場合は注意してください。

# 第7章 システム監査

Linux の Audit システムは、システムのセキュリティー関連情報を追跡する方法を提供します。事前設定されたルールに基づき、Audit は、ログエントリーを生成し、システムで発生しているイベントに関する情報をできるだけ多く記録します。この情報は、ミッションクリティカルな環境でセキュリティーポリシーの違反者と、違反者によるアクションを判断する上で必須のものです。Audit は、追加のセキュリティー機能をシステムに提供するのではありません。システムで使用されるセキュリティーポリシーの違反を発見するために使用できます。このような違反は、SELinux などの別のセキュリティー対策で防ぐことができます。

以下は、Audit がログファイルに記録できる情報の概要です。

- イベントの日時、タイプ、結果
- サブジェクトとオブジェクトの機密性のラベル
- イベントを開始したユーザーの ID とイベントの関連性
- Audit 設定の全修正および Audit ログファイルへのアクセス試行
- SSH、Kerberos、およびその他の認証メカニズムの全使用
- など、信頼されるデータベースへの変更 /etc/passwd
- システムからの情報のインポート、およびシステムへの情報のエクスポートの試行
- ユーザー ID、サブジェクトおよびオブジェクトラベルなどの属性に基づく include または exclude イベント

Audit システムの使用は、多くのセキュリティー関連の認定における要件でもあります。Audit は、以下の認定またはコンプライアンスガイドの要件に合致するか、それを超えるように設計されています。

- Controlled Access Protection Profile (CAPP)
- Labeled Security Protection Profile (LSPP)
- Rule Set Base Access Control (RSBAC)
- NISPOM (National Industrial Security Program Operating Manual)
- Federal Information Security Management Act (FISMA)
- PCI DSS (Payment Card Industry Data Security Standard)
- セキュリティー技術実装ガイド (Security Technical Implementation Guide (STIG))

Audit は以下でも認定されています。

- National Information Assurance Partnership (NIAP) および Best Security Industries (BSI) による評価
- Red Hat Enterprise Linux 5 の LSPP/CAPP/RSBAC/EAL4 以降の認定
- Red Hat Enterprise Linux 6 における OSPP/EAL4 以降(Operating System Protection Profile / Evaluation Assurance Level 4+)の認定

使用例

ファイルアクセスの監視

Audit は、ファイルまたはディレクトリーがアクセス、修正、実行されているか、またはファイルの属性が変更されたかを追跡できます。これはたとえば、重要なファイルへのアクセスを検出し、これらのファイルが破損した場合に監査証跡を入手可能とする際に役に立ちます。

システムコールの監視

Audit は、一部のシステムコールが使用されるたびにログエントリーを生成するように設定できます。これを使用すると、settimeofday や clock_adjtime、その他の時間関連のシステムコールを監視することで、システム時間への変更を追跡できます。

ユーザーが実行したコマンドの記録

Audit はファイルが実行されたかどうかを追跡できるため、特定のコマンドの実行を録画するために複数のルールを定義できます。たとえば、/bin ディレクトリー内のすべての実行可能ファイルにルールを定義できます。これにより作成されるログエントリーをユーザー ID で検索すると、ユーザーごとに実行されたコマンドの監査証跡を生成できます。

セキュリティーイベントの記録

pam_faillock 認証モジュールは、失敗したログイン試行を記録できます。Audit で失敗したログイン試行も記録するように設定すると、ログインを試みたユーザーに関する追加情報が提供されます。

イベントの検索

Audit は ausearch ユーティリティーを提供します。これを使用すると、ログエントリーをフィルターにかけ、いくつもの条件に基づく完全な監査証跡を提供できます。

サマリーレポートの実行

aureport ユーティリティーを使用すると、記録されたイベントのデイリーレポートを生成できます。システム管理者は、このレポートを分析し、疑わしいアクティビティーをさらに調べることができます。

ネットワークアクセスの監視

iptables ユーティリティーおよび ebtables ユーティリティーは、Audit イベントを発生するように設定できるため、システム管理者がネットワークアクセスを監視できるようになります。

注記

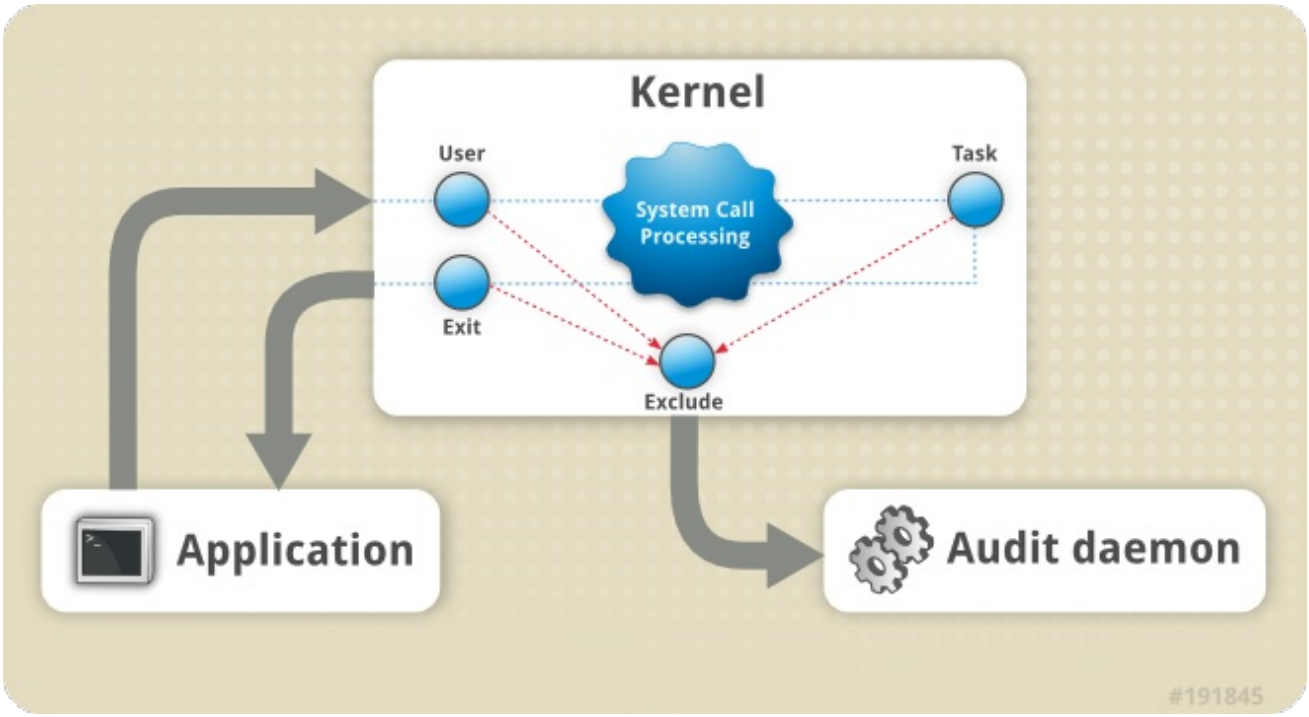
システムのパフォーマンスは、Audit が収集する情報量によって影響される可能性があります。

## 7.1. AUDIT システムのアーキテクチャー

Audit システムは、ユーザー空間アプリケーションおよびユーティリティーと、カーネル側のシステ

ムコール処理という 2 つの主要部分で構成されます。カーネルコンポーネントは、ユーザー空間アプリケーションからシステムコールを受け、これを user、task、または exit のいずれかのフィルターで振り分けます。システムコールがこれらのフィルターのいずれかを通過したら、除外 フィルターを介して送信されます。図7.1「Audit システムのアーキテクチャー」 これは、Audit ルール設定に基づいて、詳細な処理を行うために Audit デーモンに送信します。

図7.1 Audit システムのアーキテクチャー




[D]

ユーザー空間の Audit デーモンは、カーネルから情報を収集し、ログファイルエントリーをログファイルに作成します。他のユーザー空間ユーティリティーは、Audit デーモン、カーネルの Audit コンポーネント、または Audit ログファイルと相互作用します。

- audisp - Audit dispatcher デーモンは Audit デーモンと対話し、詳細な処理のためにイベントを他のアプリケーションに送信します。このデーモンの目的は、リアルタイムの分析プログラムが Audit イベントと対話できるようにプラグインメカニズムを提供することです。

- auditctl - Audit 制御ユーティリティーはカーネル Audit コンポーネントと相互作用し、イベント生成プロセスの多くの設定およびパラメーターを制御します。

- 残りの Audit ユーティリティーは、Audit ログファイルのコンテンツを入力として取り、ユーザーの要件に基づいて出力を生成します。たとえば、aureport ユーティリティーは、記録された全イベントのレポートを生成します。

## 7.2. AUDIT パッケージのインストール

Audit システムを使用するには、audit パッケージがシステムにインストールされている必要があります。audit パッケージ（audit および audit-libs）は、Red Hat Enterprise Linux 6 にデフォルトでインストールされます。これらのパッケージがインストールされていない場合は、root ユーザーとして以下のコマンドを実行してインストールします。

```
~]# yum install audit
```

## 7.3. 監査 サービスの設定

Audit デーモンは /etc/audit/auditd.conf 設定ファイルで設定できます。このファイルは、Audit デーモンの動作を変更する設定パラメーターで構成されます。空の行やハッシュ記号(#)の後に続くテキストは無視されます。以下を参照してください。 auditd.conf(5) man ページは、すべての設定パラメーターとその説明の完全リストを提供します。

### 7.3.1. CAPP 環境での auditd の設定

デフォルトの auditd 設定は、ほとんどの環境に適しています。ただし、ご使用の環境が Common Criteria 証明書の一部である Controlled Access Protection Profile (CAPP)で設定されている基準を満たす必要がある場合は、Audit デーモンを以下の設定で設定する必要があります。

- Audit ログファイル（通常は /var/log/audit/）を保持するディレクトリーは、別のパーティションに存在する必要があります。これにより、その他のプロセスがこのディレクトリー内の領域を使用しないようにし、Audit デーモンの残りの領域を正確に検出します。

- 1 つの Audit ログファイルの最大サイズを指定する *max_log_file* パラメーターは、Audit ログファイルを保持するパーティションで利用可能な領域をすべて使用するように設定する必要があります。

- に設定されている制限に *max_log_file* 達すると実行するアクションを指定する *max_log_file_action* パラメーターは、Audit ログファイルが上書きされないよう keep_logs に設定する必要があります。

- *space_left* パラメーターで設定したアクションがトリガーされるディスク上に残される空き領域の量を指定する *space_left_action* パラメーターは、管理者がディスク領域に対応および解放するのに十分な時間を設定する必要があります。この *space_left* 値は、Audit ログファイルが生成される速度によって異なります。

- *space_left_action* パラメーターを exec 適切な通知方法に設定 email することが推奨されます。

- *admin_space_left* パラメーターで設定したアクションがトリガーされるディスクの最小領域の最小量を指定する *admin_space_left_action* パラメーターは、管理者が実行するアクションのログを記録するのに十分な領域を残す値に設定する必要があります。

- *admin_space_left_action* パラメーターは、システムを single シングルユーザーモードにし、管理者がディスク領域を解放できるように設定する必要があります。

- *disk_full_action* パラメーター。Audit ログファイルを保持するパーティションに空き領域がない場合にトリガーされる動作を指定します。このパラメーターは、halt またはに設定する必要があり singleます。これにより、Audit がイベントをログに記録できなくなると、システムは、シングルユーザーモードでシャットダウンまたは動作します。

- *disk_error_action*。Audit ログファイルを保持するパーティションでエラーが検出された場合に発生するアクションを指定する、、、、または、ハードウェアの誤作動の処理に関するローカルのセキュリティーポリシーに応じて halt、、syslog singleまたはを設定する必要があります。

- *flush* 設定パラメーターは sync またはに設定する必要があり dataます。このパラメーターにより、すべての Audit イベントデータがディスクのログファイルと完全に同期されます。

残りの設定オプションは、ローカルのセキュリティーポリシーに合わせて設定します。

### 7.4. 監査 サービスの起動

auditd が適切に設定されたら、サービスを起動して Audit 情報を収集し、ログファイルに保存します。root ユーザーとして以下のコマンドを実行して auditd を起動します。

```
~]# service auditd start
```

必要に応じて、root ユーザーで次のコマンドを使用して、システムの起動時に auditd が起動するように設定できます。

```
~]# chkconfig auditd on
```

その他のアクションは、service auditd *action* コマンドを使用して auditd で実行できます。ここでの *アクション* は以下のいずれかになります。

- stop : auditd を停止します。
- restart : auditd を再起動します。
- reload または force-reload : /etc/audit/auditd.conf ファイルから auditd の設定を再読み込みします。
- rotate : /var/log/audit/ ディレクトリー内のログファイルをローテーションします。
- resume : Audit イベントのログが以前に一時停止された後に再開します。たとえば、Audit ログファイルが含まれるディスクパーティションに十分な空き領域がない場合などです。
- condrestart または try-restart : auditd がすでに実行している場合にのみ再起動します。
- status : auditd の稼働状況を表示します。

## 7.5. 監査ルールの定義

Audit システムは、ログファイルでキャプチャーされる内容を定義する一連のルールで動作します。指定できる Audit ルールには、以下の 3 つのタイプがあります。

- コントロールルール： Audit システムの動作と、その設定の一部の変更を許可します。
- ファイルシステムルール（ファイル監視とも呼ばれる）は、特定のファイルまたはディレクトリーへのアクセスの監査を許可します。
- システムコールルール - 指定したプログラムが作成するシステムコールのログを許可します。

Audit ルールは、auditctl ユーティリティーを使用してコマンドラインで指定できます（これらのルールは再起動すると持続しない、または /etc/audit/audit.rules ファイルで記述されることに注意して

ください）。以下の 2 つのセクションでは、Audit ルールを定義する両方の方法についての概要を説明します。

### 7.5.1. auditctl ユーティリティーを使用した Audit ルールの定義

注記

Audit サービスと Audit ログファイルと対話するコマンドはすべて root 権限が必要です。これらのコマンドは必ず root ユーザーとして実行してください。

auditctl コマンドを使用すると、Audit システムの基本的な機能を制御し、どの Audit イベントをログに記録するかを決定するルールを定義できます。

コントロールルールの定義

以下は、Audit システムの動作を変更できるようにする制御ルールの一部です。

**-b**

カーネル内の既存の Audit バッファーの最大量を設定します。以下に例を示します。

```
~]# auditctl -b 8192
```

**-f**

以下のように、重大なエラーが検出されたときに実行されるアクションを設定します。

```
~]# auditctl -f 2
```

上記の設定は、重大なエラーが発生した場合にカーネルパニックをトリガーします。

**-e**

Audit システムを有効および無効にするか、設定をロックします。以下に例を示します。

```
~]# auditctl -e 2
```

上記のコマンドは、Audit 設定をロックします。

**-r**

**1 秒あたりに生成されたメッセージのレートを設定します。以下に例を示します。**

```
~]# auditctl -r 0
```

**上記の設定では、生成されるメッセージに対するレート制限は設定されません。**

**-s**

**Audit システムのステータスを報告します。以下に例を示します。**

```
~]# auditctl -s
AUDIT_STATUS: enabled=1 flag=2 pid=0 rate_limit=0 backlog_limit=8192 lost=259 backlog=0
```

**-l**

**現在読み込み済みの Audit ルールを一覧表示します。以下に例を示します。**

```
~]# auditctl -l
LIST_RULES: exit,always watch=/etc/localtime perm=wa key=time-change
LIST_RULES: exit,always watch=/etc/group perm=wa key=identity
LIST_RULES: exit,always watch=/etc/passwd perm=wa key=identity
LIST_RULES: exit,always watch=/etc/gshadow perm=wa key=identity
 ⋮
```

**-D**

**現在読み込まれている Audit ルールをすべて削除します。以下に例を示します。**

```
~]# auditctl -D
No rules
```

**ファイルシステムルールの定義**

**ファイルシステムルールを定義するには、以下の構文を使用します。**

```
auditctl -w path_to_file -p permissions -k key_name
```

**詳細は以下のようになります。**

- *path_to_file* は、監査されるファイルまたはディレクトリーです。
- パーミッション はログに記録されるパーミッションです。
  - r : ファイルまたはディレクトリーへの読み取りアクセス
  - w : ファイルまたはディレクトリーへの書き込みアクセス。
  - x : ファイルまたはディレクトリーへのアクセスを実行します。
  - a : ファイルまたはディレクトリーの属性を変更します。
- *key_name* は、特定のログエントリーを生成したルールまたは一連のルールの特定に役立つオプションの文字列です。

**例7.1 ファイルシステムのルール**

すべての書き込みアクセスと、/etc/passwd ファイルのすべての属性変更をログに記録するルールを定義するには、以下のコマンドを実行します。

```
~]# auditctl -w /etc/passwd -p wa -k passwd_changes
```

**-k** オプションの後に続く文字列は任意であることに注意してください。

すべての書き込みアクセスと、/etc/selinux/ ディレクトリー内の全ファイルに対するすべての属性変更をログに記録するルールを定義するには、以下のコマンドを実行します。

```
~]# auditctl -w /etc/selinux/ -p wa -k selinux_changes
```

/sbin/insmod コマンドの実行（Linux カーネルにモジュールを挿入する）をログに記録するルールを定義するには、以下のコマンドを実行します。

```
~]# auditctl -w /sbin/insmod -p x -k module_insertion
```

**システムコールルールの定義**

**システムコールルールを定義するには、以下の構文を使用します。**

```
auditctl -a action,filter -S system_call -F field=value -k key_name
```

**詳細は以下のようになります。**

- ***action* および *filter* は、特定のイベントがログに記録されるタイミングを指定します。*アクション* は、always またはのいずれかです never。*filter* は、イベントに適用されるカーネルルールマッチングフィルターを指定します。rule-matching フィルターは task、、exit user、およびのいずれかになります exclude。これらのフィルターの詳細は、の最初を参照してください 「Audit システムのアーキテクチャー」。

- ***system_call* は、名前でシステムコールを指定します。すべてのシステムコールの一覧は、/usr/include/asm/unistd_64.h ファイルにあります。-S オプションの後に指定した各システムコールは、1 つのルールにグループ化できます。

- ***field=value* は、指定したアーキテクチャー、グループ ID、プロセス ID などに基づいてイベントに一致するようにさらにルールを変更する追加のオプションを指定します。利用可能なすべてのフィールドタイプとその値の一覧は、を参照してください。 auditctl(8) の man ページ。

- ***key_name* は、特定のログエントリーを生成したルールまたは一連のルールの特定に役立つオプションの文字列です。

**例7.2 システムコールルール**

**adjtimex または settimeofday システムコールがプログラムで使用され、システムが 64 ビットアーキテクチャーを使用するたびにログエントリーを作成するルールを定義するには、以下のコマンドを実行します。**

```
~]# auditctl -a always,exit -F arch=b64 -S adjtimex -S settimeofday -k time_change
```

**ID が 500 以上のシステムユーザーがファイルを削除したり、名前を変更するたびにログエントリーを作成するルールを定義するには（この -F auid!=4294967295 オプションを使用して、ログイン UID が設定されていないユーザーを除外する場合）、以下のコマンドを実行します。**

```
~]# auditctl -a always,exit -S unlink -S unlinkat -S rename -S renameat -F auid>=500 -F
auid!=4294967295 -k delete
```

また、システムコールルール構文を使用してファイルシステムルールを定義することもできます。以下のコマンドは、**-w /etc/shadow -p wa** ファイルシステムルールに類似するシステムコールのルールを作成します。

```
~]# auditctl -a always,exit -F path=/etc/shadow -F perm=wa
```

### 7.5.2. ファイルでの永続監査ルールおよび制御の /etc/audit/audit.rules 定義

再起動後も持続する **Audit** ルールを定義するには、**/etc/audit/audit.rules** ファイルに含める必要があります。このファイルは、同じ **auditctl** コマンドライン構文を使用してルールを指定します。空の行やハッシュ記号(#)の後に続くテキストは無視されます。

また、**auditctl** コマンドは、**-R** オプションを使用して、指定したファイルからルールを読み込むために使用することもできます。以下に例を示します。

```
~]# auditctl -R /usr/share/doc/audit-version/stig.rules
```

**コントロールルールの定義**

ファイルには、**Audit** システムの動作を変更する以下の制御ルール（、**-b**、、**-D -e -f**、および）のみを含めることができます **-r**。これらのオプションの詳細は、を参照してください 「コントロールルールの定義」。

**例7.3 の制御ルール audit.rules**

```
# Delete all previous rules
-D

# Set buffer size
-b 8192

# Make the configuration immutable -- reboot is required to change audit rules
-e 2

# Panic when a failure occurs
-f 2

# Generate at most 100 audit messages per second
-r 100
```

ファイルシステムおよびシステムコールルールの定義

ファイルシステムおよびシステムコールルールは、auditctl 構文を使用して定義されます。の例は、以下のルールファイルで表示 「auditctl ユーティリティーを使用した Audit ルールの定義」 できます。

例7.4 のファイルシステムおよびシステムコールルール audit.rules

```
-w /etc/passwd -p wa -k passwd_changes
-w /etc/selinux/ -p wa -k selinux_changes
-w /sbin/insmod -p x -k module_insertion

-a always,exit -F arch=b64 -S adjtimex -S settimeofday -k time_change
-a always,exit -S unlink -S unlinkat -S rename -S renameat -F auid>=500 -F auid!=4294967295 -k delete
```

事前設定されたルールファイル

/usr/share/doc/audit-*version*/ ディレクトリーでは、audit パッケージは、さまざまな証明書規格に従って、事前設定されたルールファイルのセットを提供します。

- nispom.rules - National Industrial Security Program Operating Manual の第 8 章で指定されている要件を満たす監査ルール設定
- capp.rules : Common Criteria 証明書に含まれる Controlled Access Protection Profile (CAPP)によって設定されている要件を満たす監査ルール設定。
- lspp.rules : Common Criteria 証明書に含まれる Labeled Security Protection Profile (LSPP)で設定されている要件を満たす監査ルール設定。
- stig.rules : セキュリティー技術実装ガイド(STIG)で設定されている要件を満たす監査ルール設定。

これらの設定ファイルを使用するには、元の /etc/audit/audit.rules ファイルのバックアップを作成し、任意の設定ファイルを /etc/audit/audit.rules ファイルにコピーします。

```
~]# cp /etc/audit/audit.rules /etc/audit/audit.rules_backup
~]# cp /usr/share/doc/audit-version/stig.rules /etc/audit/audit.rules
```

## 7.6. AUDIT ログファイルについて

デフォルトでは、Audit システムはログエントリーを /var/log/audit/audit.log ファイルに保存します。ログローテーションが有効になると、ローテーションされた audit.log ファイルは同じディレクトリーに保存されます。

以下の Audit ルールは、/etc/ssh/sshd_config ファイルの読み取りまたは修正の試行をすべてログに記録します。

```
-w /etc/ssh/sshd_config -p warx -k sshd_config
```

auditd デーモンが実行している場合は、以下のコマンドを実行して Audit ログファイルに新しいイベントを作成します。

```
~]# cat /etc/ssh/sshd_config
```

audit.log ファイルのこのイベントは、以下のようになります。

```
type=SYSCALL msg=audit(1364481363.243:24287): arch=c000003e syscall=2 success=no exit=-13
a0=7fffd19c5592 a1=0 a2=7fffd19c4b50 a3=a items=1 ppid=2686 pid=3538 auid=500 uid=500
gid=500 euid=500 suid=500 fsuid=500 egid=500 sgid=500 fsgid=500 tty=pts0 ses=1 comm="cat"
exe="/bin/cat" subj=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023 key="sshd_config"
type=CWD msg=audit(1364481363.243:24287):  cwd="/home/shadowman"
type=PATH msg=audit(1364481363.243:24287): item=0 name="/etc/ssh/sshd_config" inode=409248
dev=fd:00 mode=0100600 ouid=0 ogid=0 rdev=00:00 obj=system_u:object_r:etc_t:s0
```

上記のイベントは、3 つのレコード（type= キーワードで始まる）で構成されており、タイムスタンプとシリアル番号を共有します。各レコードは、空白またはコンマで区切られた複数の *name=value* ペアで構成されます。上記のイベントの詳細な分析は以下のようになります。

**1 つ目のレコード**

**type=SYSCALL**

type フィールドには、レコードのタイプが含まれます。この例では、SYSCALL 値は、カーネルへのシステムコールによりこのレコードがトリガーされたことを示しています。

可能なすべてのタイプ値とその説明は、を参照してください 「監査レコードタイプ」。

**msg=audit(1364481363.243:24287):**

msg フィールドは以下を記録します。

- フォーム内のレコードのタイムスタンプと一意の ID audit(*time_stamp*:*ID*)。複数のレコードが同じ Audit イベントの一部として生成されている場合は、同じタイムスタンプおよび ID を共有できます。

- カーネルまたはユーザー空間アプリケーションが提供するさまざまなイベント固有の *name=value* ペア。

**arch=c000003e**

arch フィールドには、システムの CPU アーキテクチャーに関する情報が含まれます。値は 16 進数表記 c000003eでエンコードされます。ausearch コマンドで Audit レコードを検索する場合は、-i または --interpret オプションを使用して、16 進数の値を人間が判読できる値に自動的に変換します。この c000003e 値はとして解釈され x86_64ます。

**syscall=2**

syscall フィールドは、カーネルに送信されたシステムコールのタイプを記録します。の値は 2、/usr/include/asm/unistd_64.h ファイルで人間が判読できる値と一致します。この場合、2 は open システムコールです。ausyscall ユーティリティーでは、システムコール番号を、人間が判読できる値に変換できます。ausyscall --dump コマンドを使用して、システムコールの一覧とその数字を表示します。詳細はを参照してください。 ausyscall(8) の man ページ。

**success=no**

success フィールドは、その特定のイベントで記録されたシステムコールが成功したかどうかを記録します。この例では、呼び出しが成功しませんでした。

**exit=-13**

exit フィールドには、システムコールが返した終了コードを指定する値が含まれます。この値は、システムコールにより異なります。この値は、次のコマンドで人間が判読できるものに変換できます ausearch --interpret --exit -13 （Audit ログに終了コードで失敗したイベントが含まれていることを前提とします -13）。

**a0=7fffd19c5592, a1=0, a2=7fffd19c5592, a3=a**

a0 to a3 フィールドは、このイベントにおけるシステムコールの最初の 4 つの引数（16 進数表記でエンコード）を記録します。この引数は、使用されるシステムコールにより異なります。ausearch ユーティリティーで解釈できます。

**items=1**

items フィールドには、イベント内のパスレコードの数が含まれます。

**ppid=2686**

この ppid フィールドは、親プロセス ID(PPID)を記録します。この場合、2686 は bash プロセスの PPID です。

**pid=3538**

この pid フィールドは、プロセス ID(PID)を記録します。この例で 3538 は、は cat プロセスの PID です。

**auid=500**

この auid フィールドは、loginuid である Audit ユーザー ID を記録します。この ID は、ログイン時にユーザーに割り当てられ、ユーザーのアイデンティティーが変更される場合でも（たとえば、su - john コマンドでユーザーアカウントを切り替えることで）すべてのプロセスによって継承されます。

**uid=500**

uid フィールドは、解析しているプロセスを開始したユーザーのユーザー ID を記録します。ユーザー ID は、次のコマンドでユーザー名に変換できます ausearch -i --uid *UID*。ここでは、500 はユーザーのユーザー ID です shadowman。

**gid=500**

gid フィールドは、解析しているプロセスを開始したユーザーのグループ ID を記録します。

**euid=500**

euid フィールドは、解析しているプロセスを開始したユーザーの実効ユーザー ID を記録します。

**suid=500**

suid フィールドは、解析しているプロセスを開始したユーザーのセットユーザー ID を記録します。

**fsuid=500**

fsuid フィールドは、解析しているプロセスを開始したユーザーのファイルシステムユーザー ID

を記録します。

**egid=500**

egid フィールドは、解析しているプロセスを開始したユーザーの実効グループ ID を記録します。

**sgid=500**

sgid フィールドは、解析しているプロセスを開始したユーザーのセットグループ ID を記録します。

**fsgid=500**

fsgid フィールドは、解析しているプロセスを開始したユーザーのファイルシステムグループ ID を記録します。

**tty=pts0**

tty フィールドは、解析しているプロセスが開始したターミナルを記録します。

**ses=1**

ses フィールドは、解析しているプロセスが開始したセッションのセッション ID を記録します。

**comm="cat"**

comm フィールドは、解析しているプロセスを開始するために使用したコマンドのコマンドライン名を記録します。この場合、cat コマンドを使用してこの Audit イベントをトリガーしました。

**exe="/bin/cat"**

exe フィールドは、解析しているプロセスを開始するために使用した実行可能ファイルへのパスを記録します。

**subj=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023**

subj フィールドは、解析しているプロセスの実行時にラベル付けされた SELinux コンテキストを記録します。

**key="sshd_config"**

key フィールドは、Audit ログでこのイベントを生成したルールに関連付けられた管理者定義の文字列を記録します。

2 つ目のレコード

type=CWD

2 つ目のレコードでは、type フィールドの値は、CWD 現在の作業ディレクトリーです。このタイプは、最初のレコードで指定されたシステムコールを開始したプロセスの作業ディレクトリーを記録するために使用されます。

この記録の目的は、相対パスが関連する PATH レコードにキャプチャーされた場合に、現在のプロセスの位置を記録することです。これにより、絶対パスを再構築できます。

msg=audit(1364481363.243:24287)

msg フィールドは、最初のレコードと同じタイムスタンプと ID の値を保持します。

cwd="/home/shadowman"

cwd フィールドには、システムコールが開始したディレクトリーへのパスが含まれます。

3 つ目のレコード

type=PATH

3 つ目のレコードでは、type フィールドの値はです PATH。Audit イベントには、システムコールに引数として渡されたすべてのパスの PATHタイプのレコードが含まれます。この Audit イベントでは、引数として 1 つのパス(/etc/ssh/sshd_config)だけが使用されていました。

msg=audit(1364481363.243:24287):

msg フィールドは、1 番目と 2 つ目のレコードと同じタイムスタンプと ID の値を保持します。

item=0

item フィールドは、SYSCALL タイプレコードで参照されるアイテムの合計数のうち、現在のレコードがどのアイテムであるかを示します。この数はゼロベースで、の値は最初の項目であることを 0 意味します。

name="/etc/ssh/sshd_config"

name フィールドは、システムコールに引数として渡されたファイルまたはディレクトリーのパスを記録します。この場合、これは /etc/ssh/sshd_config ファイルです。

**inode=409248**

inode フィールドには、このイベントで記録されたファイルまたはディレクトリーに関連する inode 番号が含まれます。以下のコマンドは、409248 inode 番号に関連付けられたファイルまたはディレクトリーを表示します。

```
~]# find / -inum 409248 -print
/etc/ssh/sshd_config
```

**dev=fd:00**

dev フィールドは、このイベントで記録されたファイルまたはディレクトリーを含むデバイスのマイナー ID とメジャー ID を指定します。この場合、値は /dev/fd/0 デバイスを表します。

**mode=0100600**

mode フィールドは、ファイルまたはディレクトリーのパーミッションを数値表記で記録します。この場合、はと解釈 0100600 できます。つまり -rw-------、root ユーザーのみが /etc/ssh/sshd_config ファイルに読み取りおよび書き込み権限が付与されます。

**ouid=0**

ouid フィールドは、オブジェクトの所有者のユーザー ID を記録します。

**ogid=0**

ogid フィールドは、オブジェクトの所有者のグループ ID を記録します。

**rdev=00:00**

rdev フィールドには、特定ファイルにのみ記録されたデバイス識別子が含まれます。ここでは、記録されたファイルは通常のファイルであるため、このフィールドは使用されません。

**obj=system_u:object_r:etc_t:s0**

obj フィールドは、実行時に、記録されたファイルまたはディレクトリーにラベル付けする SELinux コンテキストを記録します。

上記で分析した Audit イベントには、イベントに含まれる可能性のあるすべてのフィールドのサブセットのみが含まれます。すべてのイベントフィールドとその説明は、を参照してください 「監査イベントフィールド」。すべてのイベントタイプとその説明は、を参照してください 「監査レコードタイプ」。

**例7.5 追加の audit.log イベント**

以下の Audit イベントは、auditd デーモンの起動に成功したことを示しています。ver フィールドは、開始した Audit デーモンのバージョンを表示します。

```
type=DAEMON_START msg=audit(1363713609.192:5426): auditd start, ver=2.2 format=raw
kernel=2.6.32-358.2.1.el6.x86_64 auid=500 pid=4979 subj=unconfined_u:system_r:auditd_t:s0
res=success
```

以下の Audit イベントは、root ユーザーとしてログインするために UID が 500 のユーザーの失敗を記録します。

```
type=USER_AUTH msg=audit(1364475353.159:24270): user pid=3280 uid=500 auid=500 ses=1
subj=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023 msg='op=PAM:authentication
acct="root" exe="/bin/su" hostname=? addr=? terminal=pts/0 res=failed'
```

## 7.7. 監査ログファイルの検索

ausearch ユーティリティーを使用すると、Audit ログファイルで特定のイベントを検索できます。デフォルトでは、ausearch は /var/log/audit/audit.log ファイルを検索します。ausearch *options* -if *file_name* コマンドを使用して、別のファイルを指定できます。1 つの ausearch コマンドで複数のオプションを指定することは、AND 演算子の使用と同じです。

**例7.6 を使用 ausearch した Audit ログファイルの検索**

/var/log/audit/audit.log ファイルで失敗したログイン試行を検索するには、以下のコマンドを使用します。

```
~]# ausearch --message USER_LOGIN --success no --interpret
```

アカウント、グループ、およびロールの変更をすべて検索するには、以下のコマンドを使用します。

```
~]# ausearch -m ADD_USER -m DEL_USER -m ADD_GROUP -m USER_CHAUTHTOK -m
DEL_GROUP -m CHGRP_ID -m ROLE_ASSIGN -m ROLE_REMOVE -i
```

ユーザーのログイン ID(auid)を使用して、特定ユーザーが実行したログに記録されたすべてのアクションを検索するには、以下のコマンドを使用します。

```
~]# ausearch -ua 500 -i
```

最大からこれまでに障害が発生したシステムコールをすべて検索するには、以下のコマンドを使用します。

```
~]# ausearch --start yesterday --end now -m SYSCALL -sv no -i
```

すべての ausearch オプションの全一覧については、を参照してください。 ausearch(8) の man ページ。

## 7.8. 監査レポートの作成

aureport ユーティリティーを使用すると、Audit ログファイルに記録されたイベントに関するサマリーおよび列レポートを生成できます。デフォルトでは、/var/log/audit/ ディレクトリー内のすべての audit.log ファイルは、レポートを作成するためにクエリーされます。aureport *options* -if *file_name* コマンドを使用して、レポートを実行する別のファイルを指定できます。

**例7.7 の使用 aureport による Audit レポートの生成**

過去 3 日（現在の例を除く）でログに記録されたイベントのレポートを生成するには、以下のコマンドを使用します。

```
~]# aureport --start 04/08/2013 00:00:00 --end 04/11/2013 00:00:00
```

すべての実行可能ファイルイベントのレポートを生成するには、以下のコマンドを使用します。

```
~]# aureport -x
```

上記の実行可能ファイルイベントレポートの概要を生成するには、以下のコマンドを使用します。

```
~]# aureport -x --summary
```

全ユーザーで失敗したイベントのサマリーレポートを生成するには、以下のコマンドを使用しま

す。

```
~]# aureport -u --failed --summary -i
```

システムユーザーごとに失敗したすべてのログイン試行の概要レポートを生成するには、以下のコマンドを使用します。

```
~]# aureport --login --summary -i
```

すべてのファイルアクセスイベントを検索する ausearch クエリーからレポートを生成するには 500、以下のコマンドを使用します。

```
~]# ausearch --start today --loginuid 500 --raw | aureport -f --summary
```

クエリーされたすべての Audit ファイルのレポートと、含まれるイベントの時間範囲を生成するには、以下のコマンドを使用します。

```
~]# aureport -t
```

すべての aureport オプションの全一覧については、を参照してください。 aureport(8) の man ページ。

## 7.9. 監査用の PAM の設定

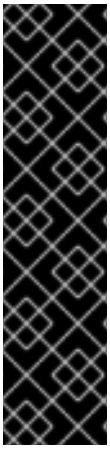
### 7.9.1. pam_tty_audit の設定

Red Hat Enterprise Linux の監査システムは、pam_tty_audit PAM モジュールを使用して、指定したユーザーの TTY 入力の監査を有効または無効にします。監査されたユーザーがログインすると、ユーザーは正確なキー入力を /var/log/audit/audit.log ファイルに pam_tty_audit 記録します。モジュールは auditd デーモンと連携するため、設定前に有効にされていることを確認し pam_tty_auditます。詳細は、「監査 サービスの起動」を参照してください。

TTY 監査のユーザー名を指定する場合は、以下の形式で /etc/pam.d/system-auth および enable オプションを使用して disable および /etc/pam.d/password-auth ファイルを変更します。

```
session required pam_tty_audit.so disable=username,username2 enable=username
```

オプションにコンマで区切られたユーザー名を 1 つ以上指定できます。disable または enable オプションを指定すると、同じユーザー名と一致する以前のオプションが上書きされます。TTY 監査を有効にすると、そのユーザーが開始したすべてのプロセスが継承されます。特に、ユーザーがデーモンを再起動する場合は、TTY 監査を有効にし、これらのユーザーの監査が明示的に無効にされていない限り、他のユーザーであっても TTY 入力を監査します。したがって、PAM を使用するほとんどのデーモンでは、を最初のオプション disable=* として使用することが推奨されます。



重要

デフォルトで pam_tty_audit は、TTY が パスワードエントリーモードの場合、はキーログを記録しません。以下の方法でオプションと他の log_passwd オプションを追加して、ロギングを再度有効にできます。

```
session required pam_tty_audit.so disable=username,username2
enable=username log_passwd
```

モジュールを有効にすると、入力は auditd デーモンによって書き込まれた /var/log/audit/audit.log ファイルに記録されます。TTY の監査が最初にバッファーに保存し、レコードを定期的に書き込み、監査されたユーザーがログアウトすると、入力がすぐにログに記録されないことに注意してください。audit.log ファイルには、バックスペース、削除、戻りキー、コントロールキーなど、指定したユーザーが入力する すべて のキーが含まれます。の内容は人間が判読できる形式ですが、aureport ユーティリティーの使用が容易である可能性があります。これにより、TTY レポートが読みやすくなる形式で提供 audit.log されます。以下のコマンドは、root で使用できます。

```
~]# aureport --tty
```

以下の例は、すべての端末で root ユーザーのアクション pam_tty_audit を追跡し、入力を確認する方法を示しています。

例7.8 pam_tty_audit で root アクションのログを記録する設定

/etc/pam.d/system-auth および /etc/pam.d/password-auth ファイルの session セクションに以下の行を入力します。

```
session    required     pam_tty_audit.so disable=* enable=root
```

aureport --tty コマンドを使用してログを表示します。root ユーザーが TTY コンソールの約 11:00 o'clock にログインし、pwd コマンドを発行しようとしますが、その ls 代わりに削除して発行すると、レポートは以下のようになります。

```
~]# aureport --tty -ts today | tail
40. 08/28/2014 11:00:27 901 0 ? 76 bash "pwd",<backspace>,<backspace>
<backspace>,"ls",<ret>
41. 08/28/2014 11:00:29 903 0 ? 76 bash <^D>
```

詳細は、pam_tty_audit(8) man ページのを参照してください。

### 7.10. その他のリソース

Audit システムの詳細は、以下の資料を参照してください。

#### オンラインのリソース

- Linux Audit システムのプロジェクトページ： http://people.redhat.com/sgrubb/audit/
- 記事「『Investigating kernel Return Codes with the Linux Audit System』 in the Hack In the Box」を参照してください http://magazine.hackinthebox.org/issues/HITB-Ezine-Issue-005.pdf。

#### インストールされているドキュメント

audit パッケージが提供するドキュメンテーションは、/usr/share/doc/audit-*version*/ ディレクトリーにあります。

#### man ページ

- audispd.conf(5)
- auditd.conf(5)

- **ausearch-expression(5)**
- **audit.rules(7)**
- **audispd(8)**
- **auditctl(8)**
- **auditd(8)**
- **aulast(8)**
- **aulastlog(8)**
- **aureport(8)**
- **ausearch(8)**
- **ausyscall(8)**
- **autrace(8)**
- **auvirt(8)**

# 第8章 OPENSCAP を使用したコンプライアンスおよび脆弱性のスキャン

## 8.1. RED HAT ENTERPRISE LINUX におけるセキュリティーコンプライアンス

*コンプライアンス監査* は、指定したオブジェクトが、コンプライアンスポリシーに記載されているすべてのルールに従っているかどうかを判断するプロセスです。*コンプライアンスポリシー* は、コンピューティング環境で使用される必要な設定を指定するセキュリティー専門家が定義します（多くの場合、チェックリストの形式を取ります）。

コンプライアンスポリシーは組織により大幅に異なることがあり、同一組織内でもシステムが異なるとポリシーが異なる可能性があります。ポリシーは、システムの目的や、組織におけるシステム重要性により異なります。カスタマイズしたソフトウェア設定や導入の特徴によっても、カスタマイズしたポリシーのチェックリストが必要になってきます。

Red Hat Enterprise Linux は、完全に自動化されたコンプライアンス監査を可能にするツールを提供します。このツールは SCAP (Security Content Automation Protocol) 規格に基づいており、コンプライアンスポリシーの自動化に合わせるように設計されています。

Red Hat Enterprise Linux 6 でサポートされるセキュリティーコンプライアンスツール

- OpenSCAP - oscap コマンドラインユーティリティーは、ローカルシステムで構成スキャンと脆弱性スキャンを実行するように設計されています。これにより、セキュリティーコンプライアンスのコンテンツを検証し、スキャンおよび評価に基づいてレポートおよびガイドを生成します。
- Script Check Engine(SCE) - SCE は SCAP プロトコルの拡張機能で、コンテンツ作成者は Bash、Python、Ruby などのスクリプト言語を使用してセキュリティーコンテンツを記述できるようにします。SCE 拡張機能は、openscap-engine-sce パッケージで提供されます。
- SCAP Security Guide(SSG) - scap-security-guide パッケージは、Linux システム向けの最新のセキュリティーポリシーコレクションを提供します。

複数のリモートシステムで自動コンプライアンス監査を実行する必要がある場合は、Red Hat Satellite 用の OpenSCAP ソリューションを利用できます。詳細は 「Red Hat Satellite での OpenSCAP の使用」 およびを参照してください 「その他のリソース」。

注記

Red Hat は、Red Hat Enterprise Linux 6 ディストリビューションに加えて、デフォルトのコンプライアンスポリシーを提供しないことに注意してください。この理由については、で説明してい 「コンプライアンスポリシーの定義」 ます。

## 8.2. コンプライアンスポリシーの定義

セキュリティーまたはコンプライアンスポリシーは、ゼロからほとんど記述されません。ISO 27000 の標準シリーズ、派生作業、およびその他のソースは、セキュリティーポリシーテンプレートと、で始まることが便利なプラクティスの推奨事項を提供します。ただし、組織は情報セキュリティープログラムを構築するため、ニーズに合わせてポリシーテンプレートを変更する必要があります。ポリシーテンプレートは、会社環境との関連性に基づいて選択し、テンプレートを調整する必要があります。テンプレートには、組織に適用できないビルドイン想定が含まれるか、またはテンプレートが明示的に特定の決定を行う必要があるためです。

Red Hat Enterprise Linux の監査機能は、SCAP(Security Content Automation Protocol)標準規格に基づいています。SCAP は相互運用可能な仕様の合成で、形式を標準化し、ソフトウェアの不具合とセキュリティー設定情報がマシンも通信します。SCAP は、自動化された設定、脆弱性およびパッチの確認、技術的な制御コンプライアンスアクティビティー、およびセキュリティーの測定に対応している多目的な仕様のフレームワークです。

つまり、SCAP はセキュリティーポリシーを表現するベンダーに依存しない方法です。したがって、最新の企業で広く使用されています。SCAP の仕様は、セキュリティーコンテンツの形式により、既知で標準化されたエコシステムが作られますが、一方で、スキャナーやポリシーエディターの導入は義務化されていません。このような状態では、企業がいくつものセキュリティーベンダーを用いていても、組織がセキュリティーポリシー (SCAP コンテンツ) を構築するのは一度で済みます。

最新バージョンの SCAP には、基礎となるいくつかの標準が含まれています。これらのコンポーネントは、以下のように SCAP 内の機能に従ってグループに分けられます。

SCAP コンポーネント

- 言語 - このグループは、コンプライアンスポリシーを表現するための標準的なボキャブラリーと規則を定義する SCAP 言語で構成されます。
  - *eXtensible Configuration Checklist Description Format(XCCDF)* - セキュリティーガイダンスを表現、整理、および管理するために設計された言語。
  - *Open Vulnerability and Assessment Language(OVAL)* - スキャンされたシステムの状態に関する論理アサーションを実行するために開発された言語。

- *Open Checklist Interactive Language(OCIL)* - ユーザーをクエリーし、指定の質問へのユーザー応答を解釈する標準的な方法を提供するために設計された言語。

- *アセット識別(AI)*: セキュリティーアセットを特定するためのデータモデル、メソッド、およびガイダンスを提供するために開発された言語。

- アセットレポート*形式(ARF)- 収集されたセキュリティーアセットおよびアセット* とセキュリティーレポート間の関係に関する情報のトランスポート形式を表現するために設計された言語。

- 重要 - このグループには、命名形式と、関係する特定のセキュリティー関連の領域のアイテムの公式リストまたはディクショナリーを定義する SCAP 標準が含まれています。

  - *Common Configuration Enumtion(CCE)*: アプリケーションおよびオペレーティングシステムのセキュリティー関連の設定要素。

  - *Common Platform Enumtion(CPE)*: 情報テクノロジー(IT)システム、プラットフォーム、およびソフトウェアパッケージの特定に使用される構造化命名スキーム。

  - *Common Vulnerabilities and Exposures(CVE)* - 公開されているソフトウェアの脆弱性と脆弱性のコレクションへの参照。

- メトリクス: このグループは、セキュリティーリスクを特定し、評価するためのフレームワークで構成されます。

  - *Common Configuration Scoring System(CCSS)*: セキュリティー関連の設定要素を評価し、ユーザーが適切な応答ステップに優先順位を設定するのに役立つスコアを割り当てるメトリックシステムです。

  - *Common Vulnerability Scoring System(CVB)* - ソフトウェア脆弱性を評価し、ユーザーがセキュリティーリスクを優先させるのに役立つスコアを割り当てるためのメトリックシステムです。

- インテグリティー - SCAP コンテンツおよびスキャン結果の整合性を維持する SCAP 仕

様。

- *TMSAD(Security Automation Data)- 署名、ハッシュ、鍵情報、およびアイデンティティー情報をセキュリティー自動化ドメイン内の XML ファイルのコンテキストで表すための既存の仕様の使用を説明する推奨事項。*

各 SCAP コンポーネントには、独自の XML ベースのドキュメント形式と、その XML ネームスペースがあります。SCAP で表現されるコンプライアンスポリシーは、単一の OVAL 定義 XML ファイル、データストリームファイル、単一 zip アーカイブ、またはポリシーチェックリストを表す XCCDF ファイルを含む個別の XML ファイルのいずれかの形式を取ることができます。

### 8.2.1. XCCDF ファイル形式

XCCDF 言語は、情報交換、ドキュメント生成、組織および状況に合わせて調整、自動化コンプライアンステスト、コンプライアンススコアリングをサポートするように設計されています。言語には主に説明があり、セキュリティースキャンを実行するコマンドは含まれていません。ただし、XCCDF ドキュメントは他の SCAP コンポーネントを参照できるため、関連する評価ドキュメント（OVAL、OCIL）を除き、すべてのターゲットプラットフォームで移植可能なコンプライアンスポリシーを作成できます。

コンプライアンスポリシーを表す一般的な方法は、XML ファイルの 1 つが XCCDF チェックリストです。この XCCDF ファイルは、通常、複数の OVAL、OCIL、および Script Check Engine(SCE)ファイルを指します。さらに、ファイルセットには、CPE ディクショナリーファイルと、このディクショナリーのオブジェクトを定義する OVAL ファイルが含まれます。

XCCDF は XML ベースの言語であるため、XML 要素および属性を大幅に定義および使用します。以下のリストは、主な XCCDF 要素を簡単に紹介します。XCCDF の詳細は、NIST Interagency Report 7275 Revision 4 を参照してください。

**XCCDF ドキュメントの主な XML 要素**

- <xccdf:Benchmark>: これは、XCCDF ドキュメント全体を囲むルート要素です。また、タイトル、説明、作成者の一覧、最新の変更日、チェックリスト受け入れステータスなどのチェックリストメタデータが含まれる場合もあります。
- <xccdf:Rule>: これはチェックリストの要件を表し、その説明を保持する主要な要素です。指定のルールとのコンプライアンスを検証または強制するアクションを定義する子要素が含まれる場合や、ルール自体を変更する場合があります。
- <xccdf:Value>: このキー要素は、ベンチマーク内の他の XCCDF 要素の属性を表現するた

めに使用されます。

- <xccdf:Group>: この要素は、<xccdf: Rule>、<xccdf: Value>、および <xccdf:Group> 要素を収集し、同じコンテキストまたは要件ドメインで構造を構成するために XCCDF ドキュメントを構成するために使用されます。

- <xccdf:Profile>: この要素は、XCCDF ベンチマークの名前付きの調整に使用します。ベンチマークが複数の異なる調整を保持できるようにします。<xccdf:Profile> は、<xccdf:select> や <xccdf:refine-rule> などの複数のセレクター要素を使用して、どの要素が有効であるかを判断します。

- <xccdf:Tailoring> - この要素は、コンプライアンスポリシーの手動による調整 が望ましい場合もあります。

- <xccdf:TestResult>: この要素は、ターゲットシステム上の所定のベンチマークのスキャン結果を保持します。<xccdf:TestResult> は、特定のスキャンのコンプライアンスポリシーを定義するために使用したプロファイルを参照し、スキャンに関連するターゲットシステムに関する重要な情報も含まれるはずです。

- <xccdf:rule-result> - これは、ベンチマークからターゲットシステムに特定のルールを適用する結果を保持するために使用される <xccdf:TestResult> の子要素です。

- <xccdf:fix> - これは <xccdf:Rule> の子要素で、指定のルールに準拠しないターゲットシステムの修復に適しています。システムにルールを準拠させるために、ターゲットシステムで実行するコマンドまたはスクリプトを含めることができます。

- <xccdf:check>: これは、指定のルールを評価する方法を定義する外部ソースを参照する <xccdf:Rule> の子要素です。

- <xccdf:select>: これは、選択したルールまたはルールのグループをポリシーから除外するために使用されるセレクター要素です。

- <xccdf:set-value>: これは、他のプロパティーを変更せずに指定された <xccdf:Value> 要素の現在の値を上書きするために使用されるセレクター要素です。

- <xccdf:refine-value>: ポリシーのテーラリング中に特定の <xccdf:Value> 要素の制約を指定するために使用されるセレクター要素です。

- **<xccdf:refine-rule>: 選択したルールのプロパティーを上書きすることができるこのセレクター要素です。**

**例8.1 XCCDF ドキュメントの例**

```
<?xml version="1.0" encoding="UTF-8"?>
<Benchmark xmlns="http://checklists.nist.gov/xccdf/1.2"
    id="xccdf_com.example.www_benchmark_test">
  <status>incomplete</status>
  <version>0.1</version>
  <Profile id="xccdf_com.example.www_profile_1">
    <title>Profile title is compulsory</title>
    <select idref="xccdf_com.example.www_group_1"
        selected="true"/>
    <select idref="xccdf_com.example.www_rule_1"
        selected="true"/>
    <refine-value idref="xccdf_com.example.www_value_1"
        selector="telnet service"/>
  </Profile>
  <Group id="xccdf_com.example.www_group_1">
    <Value id="xccdf_com.example.www_value_1">
      <value selector="telnet_service">telnet-server</value>
      <value selector="dhcp_servide">dhcpd</value>
      <value selector="ftp_service">tftpd</value>
    </Value>
    <Rule id="xccdf_com.example.www_rule_1">
      <title>The telnet-server Package Shall Not Be Installed </title>
      <rationale>
        Removing the telnet-server package decreases the risk
        of the telnet service's accidental (or intentional) activation
      </rationale>
      <fix platform="cpe:/o:redhat:enterprise_linux:6"
          reboot="false"
          disruption="low"
          system="urn:xccdf:fix:script:sh">
        yum -y remove
        <sub idref="xccdf_com.example.www_value_1"/>
      </fix>
      <check system="http://oval.mitre.org/XMLSchema/oval-definitions-5">
        <check-export value-id="xccdf_com.example.www_value_1"
            export-name="oval:com.example.www:var:1"/>
        <check-content-ref href="examplary.oval.xml"
            name="oval:com.example.www:def:1"/>
      </check>
      <check system="http://open-scap.org/page/SCE">
        <check-import import-name="stdout"/>
        <check-content-ref href="telnet_server.sh"/>
      </check>
    </Rule>
```

```
  </Group>
</Benchmark>
```

### 8.2.2. OVAL ファイル形式

セキュリティー検査言語 OVAL (Open Vulnerability Assessment Language) は、SCAP に不可欠で最も古いコンポーネントです。OVAL 規格の主な目的は、セキュリティー製品間の相互運用性を有効にすることです。これは、以下の 3 つのドメインを標準化することで実現されます。

1. ターゲットシステム設定の表現。
2. ターゲットシステムの分析で特定のマシン状態が存在する。
3. 指定されたマシンの状態と観察されたマシン状態の比較の結果を報告します。

その他のツールやカスタムスクリプトとは異なり、OVAL 言語は、宣言型でリソースの望ましい状態を記述します。OVAL 言語コードは直接実行されることはありませんが、*スキャナー* と呼ばれる OVAL インタープリターツールを用いて実行します。OVAL が宣言型であるため、評価されるシステムの状態が誤って変更されません。これは、セキュリティースキャナーが最も高い権限で実行される場合があるためです。

OVAL 仕様は、公開コメントとマーケティングのために開かれており、IT 業界が MITRE に提供され、プロモーションの非プロモーション組織が多数公開されています。OVAL 仕様は継続的に進化し、さまざまな資料はバージョン番号によって区別されます。現在のバージョン 5.10.1 は 2012 年 1 月にリリースされました。

他のすべての SCAP コンポーネントと同様に、OVAL は XML に基づいています。OVAL 標準は、いくつかのドキュメント形式を定義します。これらはそれぞれ異なる種類の情報が含まれ、異なる目的を提供します。

**OVAL ドキュメント形式**

- *OVAL Definitions* 形式は、システムスキャンに直接使用される OVAL ファイル形式です。OVAL Definitions ドキュメントでは、ターゲットシステムの望ましい状態を説明します。
- *OVAL 変数の形式は*、*OVAL* Definitions ドキュメントの修正に使用される変数を定義しま

す。OVAL 変数のドキュメントは通常、実行時にターゲットシステムのセキュリティーコンテンツを調整するために、OVAL Definitions ドキュメントとともに使用されます。

- *OVAL System Characteristics* 形式は、評価されるシステムに関する情報を保持します。OVAL System Characteristics ドキュメントは、通常、OVAL Definitions ドキュメントで定義されている期待される状態と比較するために、システムの実際の状態を比較するために使用されます。

- *OVAL Results* は、システム評価の結果を報告するのに使用される最も包括的な OVAL 形式です。OVAL Results ドキュメントには、一般的に評価される OVAL 定義のコピー、バインドされた OVAL 変数、OVAL システム特性、およびシステムの特性と定義の比較に基づいて計算されるテストの結果が含まれます。

- *OVAL Directives* 形式は、特定の詳細を含めるまたは除外することで、OVAL Result ドキュメントの冗長性を調整するために使用されます。

- *OVAL Common Model* 形式には、その他の複数の OVAL スキームで使用されるコンストラクトおよびエミュレーションの定義が含まれています。これは、複数のドキュメントでの重複を避けるために、OVAL 定義を再利用するために使用されます。

OVAL Definitions ドキュメントは、*定義*、*テスト*、*オブジェクト*、*状態*、*変数 の 5 つの基本的なセクションで各要件が定義* されている設定要件のセットで構成されています。define セクション内の要素は、指定の定義を満たすためにテストを実行するかどうかを記述します。test 要素のリンクオブジェクトと状態をまとめます。システム評価中に、指定のオブジェクト要素によって示される評価されたシステムのリソースが指定の state 要素と一致する場合に、テストが渡されるとみなされます。variables セクションは、State セクションから要素を調整するのに使用できる外部変数を定義します。これらのセクション以外には、OVAL Definitions ドキュメントには、通常 *ジェネレーター* と *署名* セクションが含まれています。*generator* セクションは、ドキュメントの作成元と、そのコンテンツに関するさまざまな追加情報を保持します。

OVAL ドキュメントの基本セクションの各要素は、以下の形式の識別子によって明確に識別されます。

```
oval:namespace:type:ID
```

*namespace* は識別子を定義する名前空間で、*type* は定義要素の場合は *def*、tests 要素の場合は *tst*、オブジェクト要素の場合は *obj*、および変数 *要素* の場合は *var*、*ID* は識別子の整数値になります。

例8.2 OVAL Definitions ドキュメントの例

```
<?xml version="1.0" encoding="utf-8"?>
<oval_definitions
  xmlns:lin-def="http://oval.mitre.org/XMLSchema/oval-definitions-5#linux"
  xmlns:oval="http://oval.mitre.org/XMLSchema/oval-common-5"
  xmlns="http://oval.mitre.org/XMLSchema/oval-definitions-5"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
 <generator>
  <oval:product_name>vim</oval:product_name>
  <oval:schema_version>5.10.1</oval:schema_version>
  <oval:timestamp>2012-11-22T15:00:00+01:00</oval:timestamp>
 </generator>
 <definitions>
  <definition class="inventory"
    id="oval:org.open-scap.cpe.rhel:def:6"
    version="1">
   <metadata>
    <title>Red Hat Enterprise Linux 6</title>
    <affected family="unix">
     <platform>Red Hat Enterprise Linux 6</platform>
    </affected>
    <reference ref_id="cpe:/o:redhat:enterprise_linux:6"
      source="CPE"/>
    <description>
     The operating system installed on the system is Red Hat Enterprise Linux 6
    </description>
   </metadata>
   <criteria>
    <criterion comment="Red Hat Enterprise Linux 6 is installed"
      test_ref="oval:org.open-scap.cpe.rhel:tst:6"/>
   </criteria>
  </definition>
 </definitions>
 <tests>
  <lin-def:rpminfo_test check_existence="at_least_one_exists"
    id="oval:org.open-scap.cpe.rhel:tst:6"
    version="1"
    check="at least one"
    comment="redhat-release is version 6">
   <lin-def:object object_ref="oval:org.open-scap.cpe.redhat-release:obj:1"/>
   <lin-def:state state_ref="oval:org.open-scap.cpe.rhel:ste:6"/>
  </lin-def:rpminfo_test>
 </tests>
 <objects>
  <lin-def:rpmverifyfile_object id="oval:org.open-scap.cpe.redhat-release:obj:1"
    version="1">
   <!-- This object represents rpm package which owns /etc/redhat-release file -->
   <lin-def:behaviors nolinkto='true'
     nomd5='true'
     nosize='true'
     nouser='true'
     nogroup='true'
     nomtime='true'
     nomode='true'
     nordev='true'
     noconfigfiles='true'
```

```
        noghostfiles='true' />
      <lin-def:name operation="pattern match"/>
      <lin-def:epoch operation="pattern match"/>
      <lin-def:version operation="pattern match"/>
      <lin-def:release operation="pattern match"/>
      <lin-def:arch operation="pattern match"/>
      <lin-def:filepath>/etc/redhat-release</lin-def:filepath>
    </lin-def:rpmverifyfile_object>
  </objects>
  <states>
    <lin-def:rpminfo_state id="oval:org.open-scap.cpe.rhel:ste:6"
        version="1">
      <lin-def:name operation="pattern match">^redhat-release</lin-def:name>
      <lin-def:version operation="pattern match">^6[^\d]</lin-def:version>
    </lin-def:rpminfo_state>
  </states>
</oval_definitions>
```

### 8.2.3. データストリーム形式

**SCAP データストリームは SCAP バージョン 1.2 以降に使用され、XCCDF チェックリストで表現されるコンプライアンスポリシーの定義に使用できる XCCDF、OVAL、およびその他のコンポーネントファイルのバンドルを表します。また、SCAP コンポーネントに従って指定のデータストリームをファイルに分割できるようにするインデックスおよびカタログも含まれます。**

**データストリームは、コンテンツの表と <ds:component> 要素のリスト別に形成されるヘッダーで構成される XML 形式を使用します。これらの各要素には、XCCDF、OVAL、CPE などの SCAP コンポーネントが含まれます。データストリームファイルには、同じタイプの複数のコンポーネントが含まれる可能性があるため、所属組織に必要なすべてのセキュリティーポリシーがカバーされます。**

**例8.3 データストリームヘッダーの例**

```
<ds:data-stream-collection xmlns:ds="http://scap.nist.gov/schema/scap/source/1.2"
  xmlns:xlink="http://www.w3.org/1999/xlink"
  xmlns:cat="urn:oasis:names:tc:entity:xmlns:xml:catalog"
  id="scap_org.open-scap_collection_from_xccdf_ssg-rhel6-xccdf-1.2.xml"
  schematron-version="1.0">
 <ds:data-stream id="scap_org.open-scap_datastream_from_xccdf_ssg-rhel6-xccdf-1.2.xml"
    scap-version="1.2" use-case="OTHER">
 <ds:dictionaries>
  <ds:component-ref id="scap_org.open-scap_cref_output--ssg-rhel6-cpe-dictionary.xml"
      xlink:href="#scap_org.open-scap_comp_output--ssg-rhel6-cpe-dictionary.xml">
    <cat:catalog>
      <cat:uri name="ssg-rhel6-cpe-oval.xml"
          uri="#scap_org.open-scap_cref_output--ssg-rhel6-cpe-oval.xml"/>
    </cat:catalog>
  </ds:component-ref>
 </ds:dictionaries>
```

```
    <ds:checklists>
      <ds:component-ref id="scap_org.open-scap_cref_ssg-rhel6-xccdf-1.2.xml"
          xlink:href="#scap_org.open-scap_comp_ssg-rhel6-xccdf-1.2.xml">
        <cat:catalog>
          <cat:uri name="ssg-rhel6-oval.xml"
              uri="#scap_org.open-scap_cref_ssg-rhel6-oval.xml"/>
        </cat:catalog>
      </ds:component-ref>
    </ds:checklists>
    <ds:checks>
      <ds:component-ref id="scap_org.open-scap_cref_ssg-rhel6-oval.xml"
          xlink:href="#scap_org.open-scap_comp_ssg-rhel6-oval.xml"/>
      <ds:component-ref id="scap_org.open-scap_cref_output--ssg-rhel6-cpe-oval.xml"
          xlink:href="#scap_org.open-scap_comp_output--ssg-rhel6-cpe-oval.xml"/>
      <ds:component-ref id="scap_org.open-scap_cref_output--ssg-rhel6-oval.xml"
          xlink:href="#scap_org.open-scap_comp_output--ssg-rhel6-oval.xml"/>
    </ds:checks>
  </ds:data-stream>
  <ds:component id="scap_org.open-scap_comp_ssg-rhel6-oval.xml"
      timestamp="2014-03-14T16:21:59">
    <oval_definitions xmlns="http://oval.mitre.org/XMLSchema/oval-definitions-5"
        xmlns:oval="http://oval.mitre.org/XMLSchema/oval-common-5"
        xmlns:ind="http://oval.mitre.org/XMLSchema/oval-definitions-5#independent"
        xmlns:unix="http://oval.mitre.org/XMLSchema/oval-definitions-5#unix"
        xmlns:linux="http://oval.mitre.org/XMLSchema/oval-definitions-5#linux"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation="http://oval.mitre.org/XMLSchema/oval-common-5
        oval-common-schema.xsd
            http://oval.mitre.org/XMLSchema/oval-definitions-5
        oval-definitions-schema.xsd
            http://oval.mitre.org/XMLSchema/oval-definitions-5#independent
        independent-definitions-schema.xsd
            http://oval.mitre.org/XMLSchema/oval-definitions-5#unix
        unix-definitions-schema.xsd
            http://oval.mitre.org/XMLSchema/oval-definitions-5#linux
        linux-definitions-schema.xsd">
```

### 8.3. SCAP WORKBENCH の使用

**SCAP Workbench (scap-workbench)はグラフィカルユーティリティーで、1 台のローカルシステムまたはリモートシステムで構成スキャンと脆弱性スキャンを実行し、システムの修復を実行して、スキャン評価に基づくレポートを生成します。oscap コマンドラインユーティリティーとの比較は、SCAP Workbench には限定的な機能しかないことに注意してください。また、SCAP Workbench は、XCCDF およびデータストリームファイルの形式でのみセキュリティーコンテンツを処理できます。**

**以下のセクションでは、SCAP Workbench をインストール、開始、および使用し、これらのタスクに関連するシステムスキャン、修復、スキャンのカスタマイズ、および表示を行う方法を説明します。**

### 8.3.1. SCAP Workbench のインストール

システムに SCAP Workbench をインストールするには、root で以下のコマンドを実行します。

```
~]# yum install scap-workbench
```

このコマンドは、ユーティリティー自体を提供するパッケージなど、SCAP Workbench が適切に機能するために必要な scap-workbench パッケージをすべてインストールします。やパッケージなどの必要な依存関係は qt、openssh パッケージがすでにインストールされている場合は、利用可能な最新バージョンに自動的に更新されます。

SCAP Workbench の使用を開始する前に、一部のセキュリティーコンテンツをシステムにインストールするか、またはインポートする必要があります。たとえば、SCAP Security Guide(SSG)パッケージをインストールできます。これには scap-security-guide、現在 Linux システム向けの最も進化し、詳細なセキュリティーポリシーが含まれます。システムに SCAP セキュリティーガイドパッケージをインストールするには、root で以下のコマンドを実行します。

```
~]# yum install scap-security-guide
```

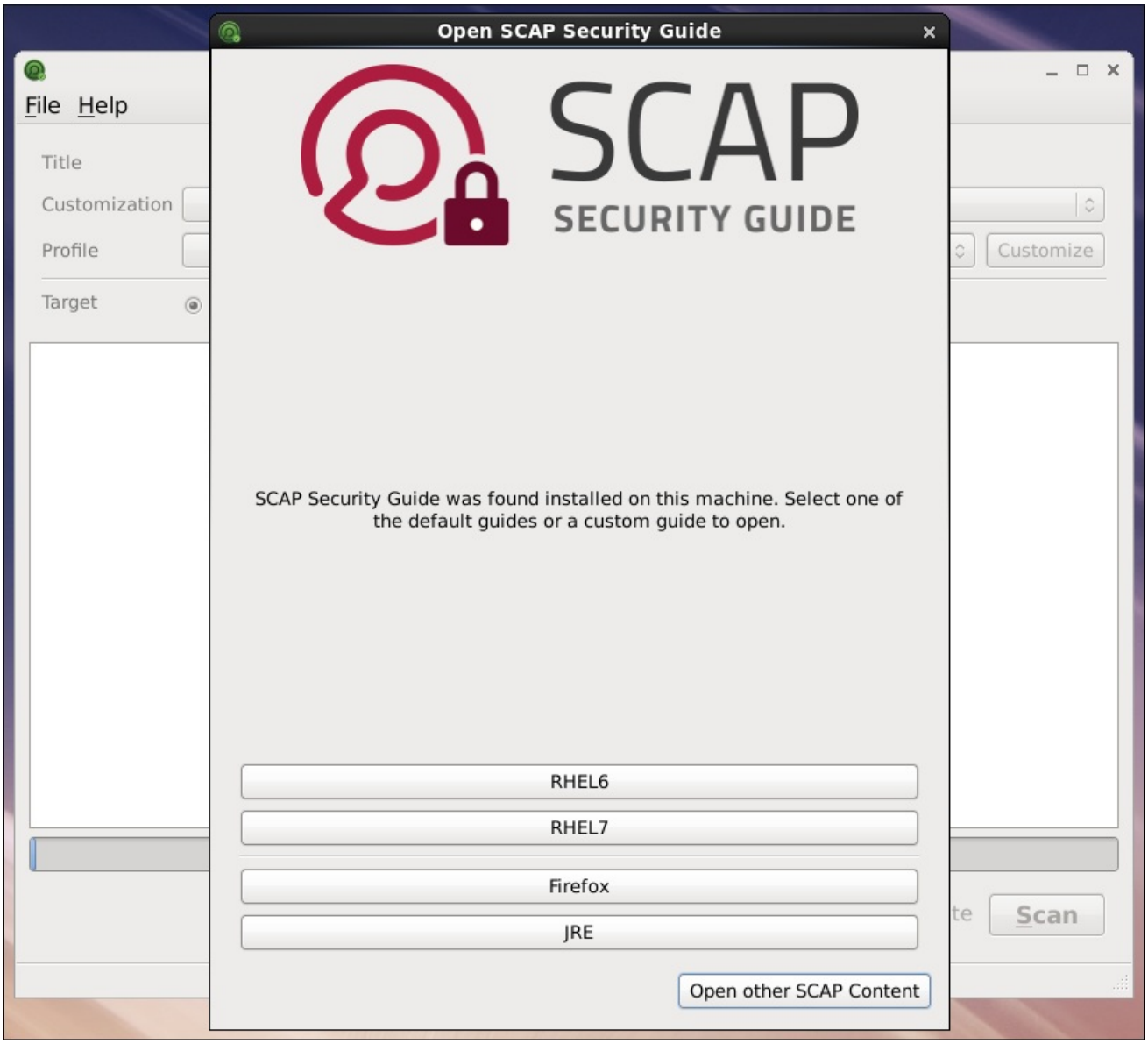
システムに scap-security-guide インストールした後、指定がない場合は SSG セキュリティーコンテンツが /usr/share/xml/scap/ssg/content/ ディレクトリー下にあり、他のセキュリティーコンプライアンス操作に進むことができます。

ニーズに適した既存の SCAP コンテンツのその他のソースを見つけるには、を参照してください 「その他のリソース」。

### 8.3.2. SCAP Workbench の実行

SCAP Workbench ユーティリティーおよび SCAP コンテンツの両方のインストールに成功すると、システムで SCAP Workbench を使用できるようになります。GNOME Classic デスクトップ環境から SCAP Workbench を実行するには、Super キーを押して アクティビティーの概要 に入り scap-workbench、と入力してを押し Enterます。Super キーはキーボード Windows や他のハードウェアによって外見が異なりますが、通常は Command キーの左側に表示され Spacebar ます。

**図8.1 Open SCAP Security Guide ウィンドウ**




[D]

ユーティリティーを起動すると、**Open SCAP Security Guide** ウィンドウが表示されます。一部のガイドを選択すると、**SCAP Workbench** ウィンドウが表示されます。このウィンドウは、複数のインタラクティブなコンポーネントで構成されており、システムのスキャンを開始する前に理解する必要があります。

**ファイル**

このメニュー一覧は、**SCAP** 関連のコンテンツをロードまたは保存するための複数のオプションを提供します。最初の **Open SCAP Security Guide** ウィンドウを表示するには、同じ名前のメニュー項目をクリックします。**Open Other Content** をクリックして、**XCCDF** 形式で別のカスタマイズファイルを読み込みます。カスタマイズを **XCCDF XML** ファイルとして保存するには、**Save Customization Only** 項目を使用します。**Save All** を使用すると、選択したディレクトリーまたは **RPM** パッケージのいずれかに **SCAP** ファイルを保存できます。

**カスタマイズ**

このボックスには、指定のセキュリティーポリシーに使用されているカスタマイズが表示されます。このボックスをクリックすると、システム評価に適用されるカスタムルールを選択できます。デフォルト値は （カスタマイズなし） です。つまり、使用されるセキュリティーポリシーは変更されません。選択したセキュリティープロファイルに変更を加えた場合、File メニューの Save Customization Only 項目をクリックすると、その変更を XML ファイルとして保存できます。

**profile**

このチェックボックスには、選択したセキュリティープロファイルの名前が含まれます。XCCDF ファイルまたはデータストリームファイルからセキュリティープロファイルを選択するには、このボックスをクリックします。選択したセキュリティープロファイルのプロパティーを継承する新規プロファイルを作成するには、Customize ボタンをクリックします。

**ターゲット**

2 つのラジオボタンを使用すると、評価するシステムがローカルマシンまたはリモートマシンであるかを選択できます。

**選択したルール**

このフィールドは、セキュリティーポリシーのサブジェクトとなるセキュリティールールの一覧を表示します。特定のセキュリティールールの拡張は、そのルールに関する詳細情報を提供します。

**ステータスバー**

これは、実行される操作のステータスを示すグラフィカルバーです。

**リモートリソースの取得**

このチェックボックスを使用すると、スキャナーに対して XML ファイルで定義されたリモート OVAL コンテンツをダウンロードするように指示できます。

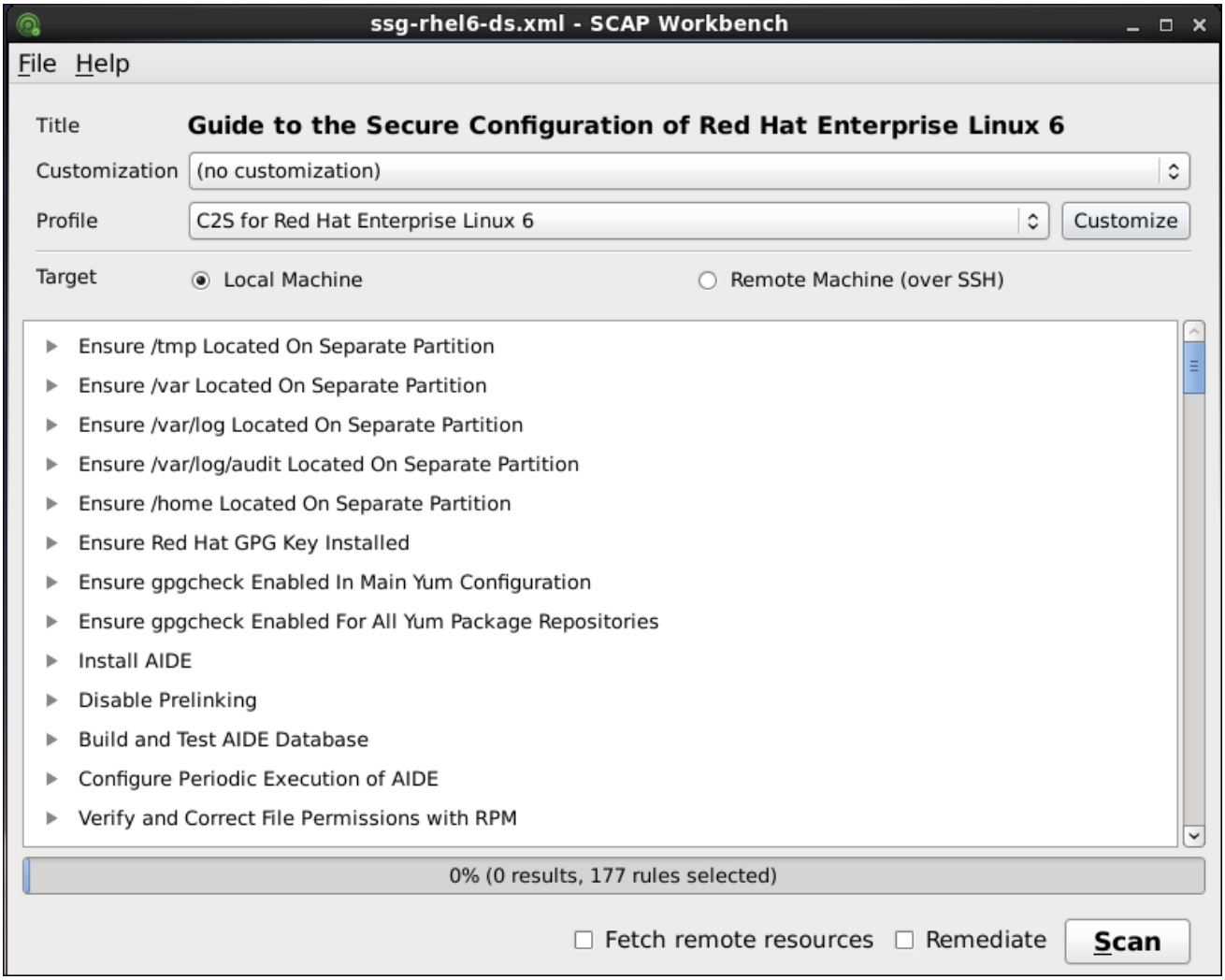
**修復**

このチェックボックスは、システムの評価中に修復機能を有効にします。このチェックボックスを選択すると、SCAP Workbench はポリシーで定義した状態と一致することに失敗するシステム設定の修正を試みます。

**scan**

このボタンを使用すると、指定したシステムの評価を開始できます。

**図8.2 SCAP Workbench Window**




[D]

### 8.3.3. システムのスキャン

**SCAP Workbench の主な機能は、所定の XCCDF ファイルまたはデータストリームファイルに従って、選択したシステムでセキュリティースキャンを実行することです。選択したセキュリティーポリシーに対してシステムを評価するには、以下の手順に従います。**

1. **Open SCAP Security Guide ウィンドウを使用するか、File メニュー Open Other Content で各 XCCDF、SCAP RPM、またはデータストリームファイルを使用してセキュリティーポリシーを選択します。**

**警告**

セキュリティーポリシーを選択すると、保存されなかったこれまでの変更が失われます。失われたオプションを再度適用するには、利用可能なプロファイルとカスタマイズ内容を再度選択する必要があります。以前の設定は、新しいセキュリティーポリシーでは適用されないことに注意してください。

2. ユースケースに固有のカスタマイズしたセキュリティーコンテンツで事前に配置されたファイルを使用するには、カスタマイズオプションボックスをクリックしてこのファイルをロード し ます。また、利用可能なセキュリティープロファイルを変更することで、カスタムの調整ファイルを作成することもできます。詳細は「セキュリティープロファイルのカスタマイズ」を参照してください。

   a. 現在のシステム評価用にカスタマイズを使用しない場合は、(no customization) オプションを選択します。以前の設定を選択していない場合は、このオプションがデフォルトオプションになります。

   b. 現在のシステム評価に使用する特定の調整ファイルを検索する (open customization file...) オプションを選択します。

   c. これまでいくつかのカスタマイズファイルを使用していた場合は、SCAP Workbench はこのファイルを記憶し、これを一覧に追加します。これにより、同じスキャンの繰り返し適用が容易になります。

3. プロファイルをクリックして、適切なセキュリティープロファイルを選択し ます。

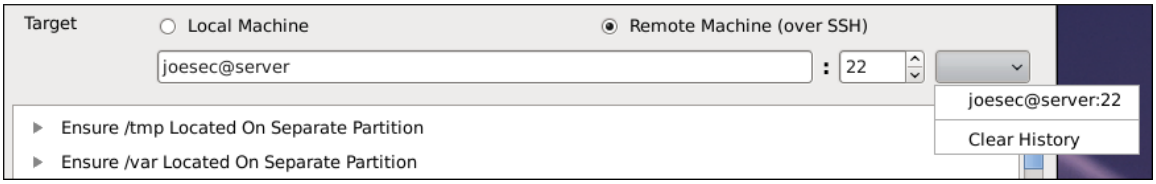
   a. 選択したプロファイルを変更するには、Customize ボタンをクリックします。プロファイルのカスタマイズに関する詳細は、を参照してください 「セキュリティープロファイルのカスタマイズ」。

4. ローカルマシンまたはリモートマシンのいずれかをスキャンするには、2 つの Target ラジオボタンを選択します。

   a. リモートシステムを選択した場合は、以下の例のようにユーザー名、ホスト名、およ

びポート情報を入力して指定します。リモートスキャンを使用していた場合は、最近スキャンされたマシンの一覧からリモートシステムを選択することもできます。

**図8.3 リモートシステムの指定**




[D]

5. **Remediate** チェックボックスを選択して、システム設定の自動修正を行うことができます。このオプションを有効にすると、**SCAP Workbench** は、ポリシーが適用したセキュリティールールに従ってシステム設定の変更を試みます。システムのスキャン中に関連するチェックが失敗すると、**SCAP Workbench** はシステムスキャンでシステム設定の変更を試みます。

**警告**

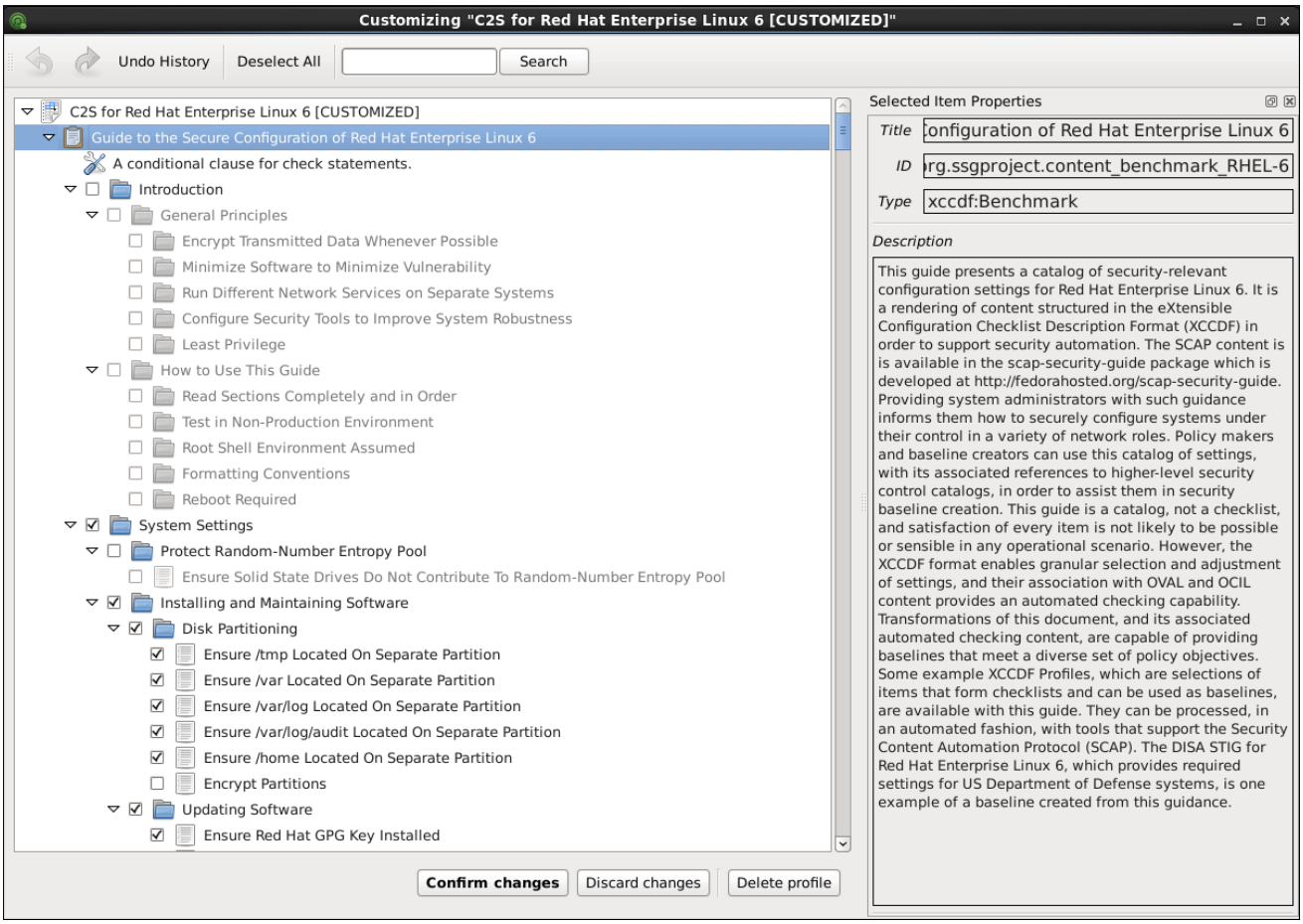
修正オプションを有効にしてシステム評価を実行すると、システムが機能しなくなることがあります。

6. **Scan** ボタンをクリックして、システムスキャンを開始します。

### 8.3.4. セキュリティープロファイルのカスタマイズ

セキュリティーポリシーに適したセキュリティープロファイルを選択すると、**Customize** ボタンをクリックしてさらに調整できます。これにより、各 **XCCDF** ファイルを実際に変更せずに、現在選択している **XCCDF** プロファイルを変更できる新しいカスタマイズウィンドウが開きます。

**図8.4 選択したセキュリティープロファイルのカスタマイズ**

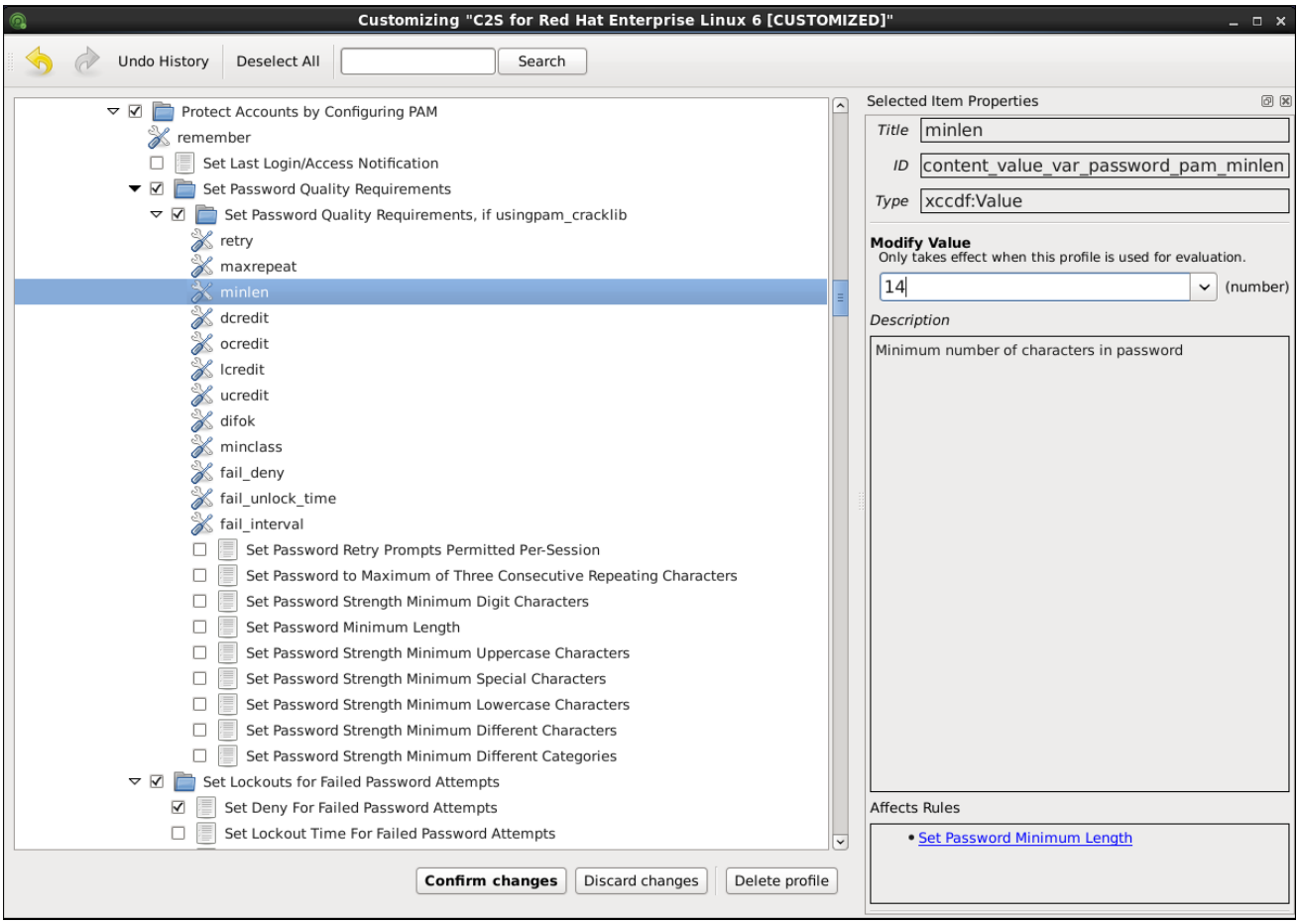



**[D]**

カスタマイズ ウィンドウには、選択したセキュリティープロファイルに関連する **XCCDF** 要素の完全なセットと、各要素とその機能に関する詳細情報が含まれます。この要素を有効または無効にするには、このウィンドウのメインフィールドで該当するチェックボックスを選択または選択解除します。カスタマイズ ウィンドウは、元に 戻す 機能 もサポートします。ウィンドウの左上隅にある矢印アイコンをクリックして、選択した内容を元に戻すことができます。

評価に使用する変数を変更することもできます。カスタマイズ ウィンドウで目的の項目を見つけ、右側に移動して **Modify value** フィールドを使用します。

**図8.5 カスタマイズウィンドウで選択したアイテムの値の設定**




[D]

プロファイルのカスタマイズが完了したら、Confirm Customization ボタンをクリックして変更を確認します。これで変更がメモリーにあり、SCAP Workbench が閉じられているり、新しい SCAP コンテンツの選択や別のカスタマイズオプションの選択など、特定の変更が行われても保持されません。変更を保存するには、SCAP Workbench ウィンドウの Save Customization ボタンをクリックします。この操作により、選択したディレクトリーに XCCDF カスタマイズファイルとしてセキュリティープロファイルへの変更を保存できます。このカスタマイズファイルは、他のプロファイルでさらに選択できます。

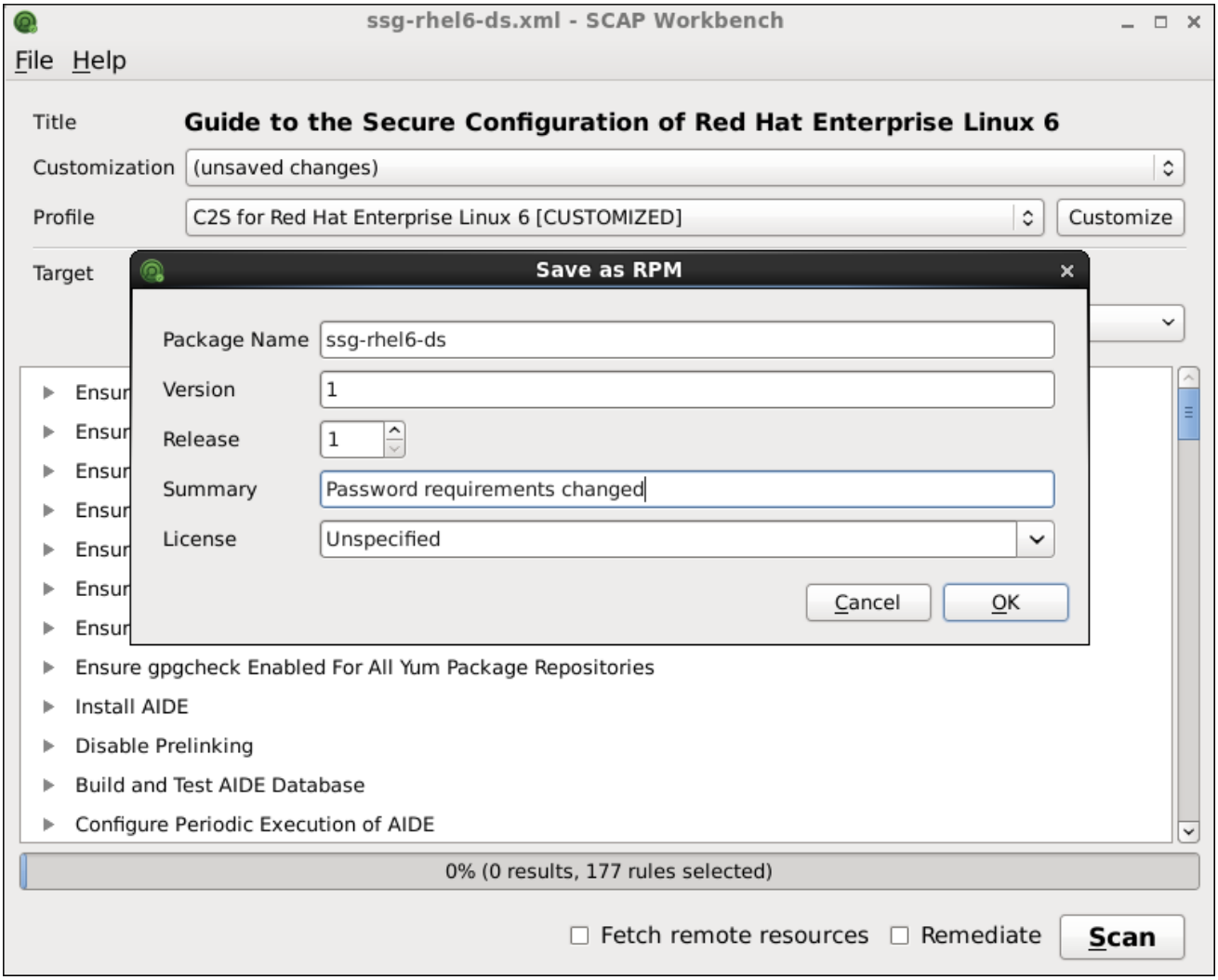
### 8.3.5. SCAP コンテンツの保存

SCAP Workbench を使用すると、システム評価で使用される SCAP コンテンツを保存することもできます。カスタマイズファイルを個別に保存することも（を参照 「セキュリティープロファイルのカスタマイズ」）、すべてのセキュリティーコンテンツを一度に保存するには、対象の Save content ボックスをクリックして Save into a directory または Save as RPM オプションを選択します。

Save into a directory オプションを選択すると、SCAP Workbench は、XCCDF ファイルまたはデータストリームファイル、ならびにカスタマイズファイルの両方を、指定した場所に保存します。これは、バックアップソリューションとして役に立ちます。

Save as RPM オプションを選択すると、SCAP Workbench に、XCCDF ファイルまたはデータストリームファイル、ならびにカスタマイズファイルを含む RPM パッケージを作成するように指示できます。これは、希望するセキュリティーコンテンツをリモートでスキャンできないシステムに配布したり、詳細な処理のためにコンテンツを配信するのに便利です。

**図8.6 現在の SCAP コンテンツを RPM パッケージとして保存**




[D]

### 8.3.6. スキャン結果の表示とスキャンレポートの生成

システムスキャンが完了すると、ボタンの代わりに 3 つの新しいボタン（、Clear Save Results Show Report、および）が表示され Scan ます。

**警告**

**Clear ボタンをクリックすると、スキャン結果を永続的に削除されます。**

スキャン結果を保存するには、XCCDF ファイル、ARF ファイル、または HTML ファイルの形式で保存するには、ダイアログボックスを Save Results クリックします。スキャンレポートを人間が判読できる形式で生成する HTML Report オプションを選択します。XCCDF 形式および ARF (データストリーム) 形式は、追加の自動処理に適しています。3 つのオプションはすべて繰り返し選択できます。

スキャン結果を保存せずに即座に表示する場合は、Show Report ボタンをクリックすると、デフォルトの Web ブラウザーで一時 HTML ファイルの形式でスキャン結果が表示されます。

### 8.4. OSCAPの使用

oscap コマンドラインユーティリティーを使用すると、ユーザーはローカルシステムのスキャン、セキュリティーコンプライアンスコンテンツの確認、ならびにスキャンおよび評価を基にしたレポートとガイドの生成が可能です。このユーティリティーは、OpenSCAP ライブラリーのフロントエンドとして機能し、その機能を処理する SCAP コンテンツのタイプに基づいてモジュール（サブコマンド）にグループ化します。

以下のセクションでは、oscap のインストール、最も一般的な操作を実行し、これらのタスクの関連する例を表示する方法を説明します。特定のサブコマンドの詳細は、oscap コマンドで --help オプションを使用します。

> **oscap [options]** ***module module_operation*** **[module_operation_options_and_arguments] --help**

***module*** は処理される SCAP コンテンツのタイプを表し、***module_operation*** は SCAP コンテンツ上の特定の操作のサブコマンドになります。

> **例8.4 特定の oscap 操作に関するヘルプの取得**
>
> ```
> ~]$ oscap ds sds-split --help
> oscap -> ds -> sds-split
>
> Split given SourceDataStream into separate files
>
> Usage: oscap [options] ds sds-split [options] SDS TARGET_DIRECTORY
> ```

```
SDS - Source data stream that will be split into multiple files.
TARGET_DIRECTORY - Directory of the resulting files.

Options:
  --datastream-id <id>          - ID of the datastream in the collection to use.
  --xccdf-id <id>               - ID of XCCDF in the datastream that should be evaluated.
```

**oscap の機能とそのオプションの完全なリストの詳細は、man ページの oscap(8) を参照してください。**

### 8.4.1. oscapのインストール

**oscap をシステムにインストールするには、root で以下のコマンドを実行します。**

```
~]# yum install openscap-scanner
```

**このコマンドを使用すると、oscap で、パッケージなど、適切に機能するために必要なパッケージをすべてインストールでき openscap ます。**

**独自のセキュリティーコンテンツを作成する場合は、Script Check Engine(SCE)を提供する openscap-engine-sce パッケージもインストールする必要があります。SCE は SCAP プロトコルの拡張機能で、コンテンツ作成者は Bash、Python、Ruby などのスクリプト言語を使用してセキュリティーコンテンツを記述できるようにします。openscap-engine-sce パッケージは、openscap-scanner パッケージと同じ方法でインストールできますが、Red Hat Enterprise Linux バリアントのオプションパッケージを使用して、リポジトリーまたはチャネルにアクセスできる必要があります。システムが Red Hat Subscription Management に登録されている場合は、『Red Hat Enterprise Linux 6 デプロイメントガイド』の Yum の章 で説明しているように、rhel-6-*variant*-optional-rpms リポジトリーを有効にします。*variant* は、サーバー や ワークステーション などの Red Hat Enterprise Linux バリアントです。システムが RHN Classic に登録されている場合は、システムを rhel-*architecture*（*variant*-6-optional チャンネル）にサブスクライブします https://access.redhat.com/site/solutions/9907。**

**必要に応じて、oscap のインストール後に、oscap のバージョンの機能、サポートする仕様、特定の oscap ファイルを保存する場所、使用可能な SCAP オブジェクトの種類、その他の有用な情報を確認できます。この情報を表示するには、以下のコマンドを入力します。**

```
~]$ oscap -V
OpenSCAP command line tool (oscap) 1.0.8
Copyright 2009--2014 Red Hat Inc., Durham, North Carolina.

==== Supported specifications ====
XCCDF Version: 1.2
OVAL Version: 5.10.1
```

```
CPE Version: 2.3
CVSS Version: 2.0
CVE Version: 2.0
Asset Identification Version: 1.1
Asset Reporting Format Version: 1.1

==== Capabilities added by auto-loaded plugins ====
SCE Version: 1.0 (from libopenscap_sce.so.8)

==== Paths ====
Schema files: /usr/share/openscap/schemas
Schematron files: /usr/share/openscap/xsl
Default CPE files: /usr/share/openscap/cpe
Probes: /usr/libexec/openscap

==== Inbuilt CPE names ====
Red Hat Enterprise Linux - cpe:/o:redhat:enterprise_linux
Red Hat Enterprise Linux 5 - cpe:/o:redhat:enterprise_linux:5
Red Hat Enterprise Linux 6 - cpe:/o:redhat:enterprise_linux:6
Red Hat Enterprise Linux 7 - cpe:/o:redhat:enterprise_linux:7
Fedora 16 - cpe:/o:fedoraproject:fedora:16
Fedora 17 - cpe:/o:fedoraproject:fedora:17
Fedora 18 - cpe:/o:fedoraproject:fedora:18
Fedora 19 - cpe:/o:fedoraproject:fedora:19
Fedora 20 - cpe:/o:fedoraproject:fedora:20
Fedora 21 - cpe:/o:fedoraproject:fedora:21
Red Hat Enterprise Linux Optional Productivity Applications - cpe:/a:redhat:rhel_productivity
Red Hat Enterprise Linux Optional Productivity Applications 5 - cpe:/a:redhat:rhel_productivity:5

==== Supported OVAL objects and associated OpenSCAP probes ====
system_info                 probe_system_info
family                      probe_family
filehash                    probe_filehash
environmentvariable         probe_environmentvariable
textfilecontent54           probe_textfilecontent54
textfilecontent             probe_textfilecontent
variable                    probe_variable
xmlfilecontent              probe_xmlfilecontent
environmentvariable58       probe_environmentvariable58
filehash58                  probe_filehash58
inetlisteningservers        probe_inetlisteningservers
rpminfo                     probe_rpminfo
partition                   probe_partition
iflisteners                 probe_iflisteners
rpmverify                   probe_rpmverify
rpmverifyfile               probe_rpmverifyfile
rpmverifypackage            probe_rpmverifypackage
selinuxboolean              probe_selinuxboolean
selinuxsecuritycontext      probe_selinuxsecuritycontext
file                        probe_file
interface                   probe_interface
password                    probe_password
process                     probe_process
runlevel                    probe_runlevel
shadow                      probe_shadow
uname                       probe_uname
```

```
xinetd                  probe_xinetd
sysctl                  probe_sysctl
process58               probe_process58
fileextendedattribute   probe_fileextendedattribute
routingtable            probe_routingtable
```

**oscap ユーティリティーを効果的に使用を開始する前に、一部のセキュリティーコンテンツをシステムにインストールするか、またはインポートする必要があります。各 Web サイトから SCAP コンテンツをダウンロードしたり、RPM ファイルまたはパッケージとして指定された場合は Yum パッケージマネージャーを使用して、指定した場所（既知のリポジトリー）からインストールできます。**

**たとえば、Linux システム向けの最新のセキュリティーポリシーを含む SCAP Security Guide(SSG)パッケージをインストールするには、以下のコマンドを実行します。**

```
~]# yum install scap-security-guide
```

**scap-security-guide パッケージをシステムにインストールし、指定しない限り、SSG セキュリティーコンテンツが /usr/share/xml/scap/ssg/content/ ディレクトリーで利用可能になり、他のセキュリティーコンプライアンス操作に進むことができます。**

**ニーズに適した既存の SCAP コンテンツの他のソースを確認するには、を参照してください 「その他のリソース」。**

**システムに SCAP コンテンツをインストールした後、oscap は、コンテンツのファイルパスを指定してコンテンツを処理できます。oscap ユーティリティーは SCAP バージョン 1.2 に対応しており、SCAP バージョン 1.1 および 1.0 と後方互換性があるので、特別な要件なしで SCAP コンテンツの以前のバージョンを処理することができます。**

### 8.4.2. SCAP コンテンツの表示

**SCAP 標準は、多数のファイル形式を定義します。oscap ユーティリティーは、多くの形式に準拠するファイルを処理または作成できます。SCAP コンテンツで指定のファイルをさらに処理するには、指定のファイルタイプとともに oscap を使用する方法を理解する必要があります。特定のファイルの使用方法が分からない場合は、ファイルを開くか、oscap の info モジュールを使用できます。これらを使用すると、ファイルを解析して、関連情報を人間が判読できる形式で抽出することができます。**

**以下のコマンドを実行して SCAP ドキュメントの内部構造を確認し、ドキュメントタイプ、仕様バージョン、ドキュメントのステータス、ドキュメントが公開された日付、ドキュメントが公開された日付、ドキュメントがファイルシステムにコピーされた日付などの役立つ情報を表示します。**

**oscap info *file***

*file* は、確認するセキュリティーコンテンツファイルのフルパスになります。以下の例は、**oscap info** コマンドの使用例を示しています。

**例8.5 SCAP コンテンツに関する情報の表示**

```
~]$ oscap info /usr/share/xml/scap/ssg/content/ssg-rhel6-ds.xml
Document type: Source Data Stream
Imported: 2014-08-28T15:41:34

Stream: scap_org.open-scap_datastream_from_xccdf_ssg-rhel6-xccdf-1.2.xml
Generated: (null)
Version: 1.2
Checklists:
        Ref-Id: scap_org.open-scap_cref_ssg-rhel6-xccdf-1.2.xml
                Profiles:
                        xccdf_org.ssgproject.content_profile_test
                        xccdf_org.ssgproject.content_profile_CS2
                        xccdf_org.ssgproject.content_profile_common
                        xccdf_org.ssgproject.content_profile_server
                        xccdf_org.ssgproject.content_profile_stig-rhel6-server-upstream
                        xccdf_org.ssgproject.content_profile_usgcb-rhel6-server
                        xccdf_org.ssgproject.content_profile_rht-ccp
                        xccdf_org.ssgproject.content_profile_CSCF-RHEL6-MLS
                        xccdf_org.ssgproject.content_profile_C2S
                Referenced check files:
                        ssg-rhel6-oval.xml
                                system: http://oval.mitre.org/XMLSchema/oval-definitions-5
Checks:
        Ref-Id: scap_org.open-scap_cref_ssg-rhel6-oval.xml
        Ref-Id: scap_org.open-scap_cref_output--ssg-rhel6-cpe-oval.xml
        Ref-Id: scap_org.open-scap_cref_output--ssg-rhel6-oval.xml
Dictionaries:
        Ref-Id: scap_org.open-scap_cref_output--ssg-rhel6-cpe-dictionary.xml
```

### 8.4.3. システムのスキャン

**oscap** の最も重要な機能は、ローカルシステムの設定スキャンと脆弱性スキャンを実行することです。各コマンドの一般的な構文を以下に示します。

```
oscap [options] module eval [module_operation_options_and_arguments]
```

**oscap** ユーティリティーは、**XCCDF (eXtensible Configuration Checklist Description Format)**ベンチマークと **OVAL (Open Vulnerability and Assessment Language)**定義の両方が表される **SCAP** コンテンツに対してシステムをスキャンできます。セキュリティーポリシーには、**OVAL** ファイルまたは **XCCDF** ファイルの形式を 1 つ持つことができ、各ファイルが異なるコンポーネント（**XCCDF**、**OVAL**、**CPE**、**CVE** など）を表します。スキャンの結果は、標準出力と **XML** ファイルの両方に出力で

きます。結果ファイルは、人間が判読できる形式でレポートを生成するために **oscap** でさらに処理できます。以下の例は、コマンドの最も一般的な使用方法を示しています。

**例8.6 SSG OVAL 定義を使用したシステムのスキャン**

すべての定義の評価中に **SSG OVAL** 定義ファイルに対してシステムをスキャンするには、以下のコマンドを実行します。

```
~]$ oscap oval eval --results scan-oval-results.xml /usr/share/xml/scap/ssg/content/ssg-rhel6-ds.xml
```

スキャンの結果は、現在のディレクトリーに **scan-oval-results.xml** ファイルとして保存されます。

**例8.7 SSG OVAL 定義を使用したシステムのスキャン**

**SSG** データストリームファイルで示されたセキュリティーポリシーから特定の **OVAL** 定義を評価するには、以下のコマンドを実行します。

```
~]$ oscap oval eval --id oval:ssg:def:100 --results scan-oval-results.xml /usr/share/xml/scap/ssg/content/ssg-rhel6-ds.xml
```

スキャンの結果は、現在のディレクトリーに **scan-oval-results.xml** ファイルとして保存されます。

**例8.8 SSG XCCDF ベンチマークを使用したシステムのスキャン**

お使いのシステムで **xccdf_org.ssgproject.content_profile_rht-ccp** プロファイル に対して **SSG XCCDF** ベンチマークを実行するには、以下のコマンドを実行します。

```
~]$ oscap xccdf eval --profile xccdf_org.ssgproject.content_profile_rht-ccp --results scan-xccdf-results.xml /usr/share/xml/scap/ssg/content/ssg-rhel6-ds.xml
```

スキャンの結果は、現在のディレクトリーに **scan-xccdf-results.xml** ファイルとして保存されます。

**注記**

**--profile** コマンドライン引数は、指定の **XCCDF** ファイルまたはデータストリームファイルからセキュリティープロファイルを選択します。**oscap info** コマンドを実行して、利用可能なプロファイルの一覧を取得できます。**--profile** コマンドライン引数を省略する場合は、**SCAP** 標準で必要なようにデフォルトの **XCCDF** プロファイルが使用されます。デフォルトの **XCCDF** プロファイルは、適切なセキュリティーポリシーである可能性があることに注意してください。

### 8.4.4. レポートおよびガイドの生成

**oscap** のもう 1 つの便利な機能は、**SCAP** コンテンツを人間が判読可能な形式で生成できることです。**oscap** ユーティリティーを使用すると、**XML** ファイルを **HTML** またはプレーンテキスト形式に変換できます。この機能は、情報源となるセキュリティーガイドおよびチェックリストの生成に使用され、システム設定のセキュアなガイダンスを提供します。システムスキャンの結果は、適切に読み取り可能な結果レポートに変換することもできます。一般的なコマンド構文は以下のとおりです。

**oscap** ***module*** **generate** ***sub-module*** **[specific_module/sub-module_options_and_arguments]** ***file***

ここで、***module*** は **xccdf** または **oval**、*サブモジュール* は生成されたドキュメントのタイプで、***file*** は **XCCDF** または **OVAL** ファイルを表します。

以下は、コマンドの使用方法の最も一般的な例です。

**例8.9 チェックリストを使用したガイドの生成**

**xccdf_org.ssgproject.content_profile_rht-ccp** プロファイル のチェックリストを含む **SSG** ガイドを生成するには、以下のコマンドを実行します。

```
~]$ oscap xccdf generate guide --profile xccdf_org.ssgproject.content_profile_rht-ccp
/usr/share/xml/scap/ssg/content/ssg-rhel6-ds.xml > ssg-guide-checklist.html
```

本ガイドは、現在のディレクトリーに **ssg-guide-checklist.html** ファイルとして保管されます。

**例8.10 SSG OVAL Scan Result のレポートへの変換**

**SSG OVAL スキャンの結果を HTML ファイルに変換するには、以下のコマンドを実行します。**

```
~]$ oscap oval generate report scan-oval-results.xml > ssg-scan-oval-report.html
```

**結果レポートは、現在のディレクトリーの ssg-scan-oval-report.html ファイルとして保存されます。この例では、scan-oval-results.xml ファイルが保存される場所と同じ場所からコマンドを実行します。それ以外の場合は、スキャン結果が含まれるファイルの完全修飾パスを指定する必要があります。**

**例8.11 SSG XCCDF スキャン結果のレポートへの変換**

**SSG XCCDF スキャンの結果を HTML ファイルに変換するには、以下のコマンドを実行します。**

```
~]$ oscap xccdf generate report scan-xccdf-results.xml > scan-xccdf-report.html
```

**結果レポートは、現在のディレクトリーの ssg-scan-xccdf-report.html ファイルとして保存されます。このレポートは、--report コマンドライン引数を使用してスキャンのタイミングで生成できます。**

```
~]$ oscap xccdf eval --profile xccdf_org.ssgproject.content_profile_rht-ccp --resultsscan-xccdf-
results.xml --report scan-xccdf-report.html /usr/share/xml/scap/ssg/content/ssg-rhel6-ds.xml
```

### 8.4.5. SCAP コンテンツの検証

**システムでセキュリティーポリシーを使用する前に、ポリシーで構文やセマンティクスエラーが発生しないように、ポリシーを検証する必要があります。oscap ユーティリティーを使用すると、標準の SCAP XML スキーマに対してセキュリティーコンテンツを検証できます。検証結果は標準エラーストリーム(stderr)に出力されます。このような検証コマンドの一般的な構文は以下のとおりです。**

**oscap *module* validate [module_options_and_arguments] *file***

**ここで、*file* は検証されるファイルへのフルパスになります。唯一の例外は、ではなく sds-validate 操作を使用するデータストリームモジュール(ds)です validate。以下の例にあるように、指定データストリーム内のすべての SCAP コンポーネントが自動的に検証され、コンポーネントは個別に指定されません。**

```
~]$ oscap ds sds-validate /usr/share/xml/scap/ssg/content/ssg-rhel6-ds.xml
```

OVAL 仕様など、特定の SCAP コンテンツを使用して、Schematron 検証を実行することもできます。Schematron 検証は標準検証よりも遅くなりますが、より深い分析を提供するため、より多くのエラーを検出できます。以下の例は、コマンドの典型的な使用方法を示しています。

```
~]$ oscap oval validate --schematron /usr/share/xml/scap/ssg/content/ssg-rhel6-ds.xml
```

#### 8.4.6. OpenSCAP を使用したシステムの修復

OpenSCAP を使用すると、コンプライアンス違反の状態で見つかったシステムを自動的に修正できます。システムの修復には、命令を含む XCCDF ファイルが必要です。には、scap-security-guide package 特定の修復手順が含まれます。

システム修復は、以下の手順で構成されます。

1. OpenSCAP は、通常の XCCDF 評価を実行します。
2. 結果の評価は、OVAL 定義を評価することで実行されます。失敗した各ルールは修復の候補としてマークされます。
3. OpenSCAP は適切な修正要素を検索し、解決し、環境を準備し、修正スクリプトを実行します。
4. 修正スクリプトの出力は、OpenSCAP によってキャプチャーされ、rule-result 要素に保存されます。修正スクリプトの戻り値も保存されます。
5. OpenSCAP が修正スクリプトを実行すると、直ちに OVAL 定義を再評価します（修正スクリプトが正しく適用されていることを検証するため）。この 2 回目の実行中に OVAL 評価が success を返すと、ルールの結果はになります fixed。そうでない場合はになり errorます。
6. 修正の詳細な結果は、XCCDF 出力ファイルに保存されます。これには 2 つの TestResult 要素が含まれます。最初の TestResult 要素は、修復前のスキャンを表します。2 つ目 TestResult は最初のデータから派生し、修復の結果が含まれます。

OpenSCAP には、オンライン、オフライン、およびレビューの修正に関して 3 つの動作モードがあ

ります。

#### 8.4.6.1. OpenSCAP オンライン修正

オンライン修復は、スキャン時に修正要素を実行します。評価および修正は、1 つのコマンドの一部として実行します。

オンラインの修復を有効にするには、**--remediate** コマンドラインオプションを使用します。たとえば、**scap-security-guide** パッケージを使用してオンライン修復を実行するには、以下を実行します。

```
~]$ oscap xccdf eval --remediate --profile xccdf_org.ssgproject.content_profile_rht-ccp --results scan-xccdf-results.xml /usr/share/xml/scap/ssg/content/ssg-rhel6-ds.xml
```

このコマンドの出力は、2 つのセクションで構成されます。最初のセクションは、修復前のスキャン結果を示し、2 番目のセクションに修正を適用後のスキャン結果が表示されます。2 番目の部分には **fixed**、および **error** 結果のみを含めることができます。**fixed** 結果は、修復に合格した後にスキャンが実行されていることを示しています。**error** 結果は、修復の適用後も評価は合格していないことを示します。

#### 8.4.6.2. OpenSCAP オフライン修正

オフラインの修復により、修正の実行を保留できます。最初のステップでは、システムは評価され、結果が **XCCDF** ファイルの **TestResult** 要素に保存されます。

次のステップでは、修正スクリプトを **oscap** 実行し、結果を検証します。結果を入力ファイルに保存しても、データは失われません。オフラインの修正時に、**OpenSCAP** は入力内容に基づく新しい **TestResult** 要素を作成し、すべてのデータを継承します。新規に作成された要素は、障害が発生した **rule-result** 要素にのみ **TestResult** 異なります。そのためには、修復が実行されます。

**scap-security-guide** パッケージを使用してオフラインの修復を実行するには、以下を実行します。

```
~]$ oscap xccdf eval --profile xccdf_org.ssgproject.content_profile_rht-ccp --results scan-xccdf-results.xml /usr/share/xml/scap/ssg/content/ssg-rhel6-ds.xml
```

```
~]$ oscap xccdf remediate --results scan-xccdf-results.xml scan-xccdf-results.xml
```

#### 8.4.6.3. OpenSCAP 修正の確認

review モードでは、さらに確認するためにファイルに修復手順を保存できます。修復の内容は、この操作中には実行されません。

シェルスクリプトの形式で修正手順を生成するには、以下を実行します。

```
~]$ oscap xccdf generate fix --template urn:xccdf:fix:script:sh --profile
xccdf_org.ssgproject.content_profile_rht-ccp --output my-remediation-script.sh
/usr/share/xml/scap/ssg/content/ssg-rhel6-ds.xml
```

### 8.5. RED HAT SATELLITE での OPENSCAP の使用

複数の Red Hat Enterprise Linux システムを実行している場合は、セキュリティーポリシーに準拠しているすべてのシステムを維持し、ある場所からリモートでセキュリティースキャンと評価を実行することが重要です。これには、Red Hat Satellite 5.5 以降を使用して、お使いの Satellite クライアントに spacewalk-oscap パッケージをインストールしてください。パッケージは、Red Hat Network Tools チャンネルから入手できます。

このソリューションは、セキュリティーコンプライアンススキャンを実行する 2 つの方法をサポートし、スキャン結果の表示と詳細な処理を行います。OpenSCAP Satellite Web インターフェースを使用するか、Satellite API からコマンドおよびスクリプトを実行できます。セキュリティーコンプライアンス、要件、および機能に対するこのソリューションの詳細は、Red Hat Satellite のドキュメントを参照してください。

### 8.6. キックスタートによる USGCB 対応システムのインストール

Red Hat Enterprise Linux 6.7 以降 USGCB （*米国の設定ベースライン*）ベンチマークキックスタートファイルは 、 Red Hat Enterprise Linux のサーバーバリアントに同梱されています。これにより、管理者はをインストールできます。 USGCB最小作業によるシステムに準拠している。生の Red Hat Enterprise Linux システムをインストールする代わりに、を使用してスキャンします。 USGCB SCAP コンプライアンスを実現するためにシステムのコンテンツを再設定すると、管理者は以下を使用できます。 USGCB を自動的に取得するベンチマークキックスタートファイル USGCB 起動時から準拠したシステム。

自動 インストール（キックスタートインストール）にキックスタートファイルを使用する方法と、提供されているの使用方法は『 Red Hat Enterprise Linux インストールガイド』を参照してください。 USGCB ベンチマークキックスタートファイル（ USGCB または DISA 準拠のインストールイメージの作成）The USGCB キックスタートファイルは scap-security-guide パッケージに含まれ、その永続的な場所は次のようになります。

```
/usr/share/scap-security-guide/kickstart/ssg-rhel6-usgcb-server-with-gui-ks.cfg
```

### 8.7. 実用的な例

本項では、Red Hat 製品が提供する特定のセキュリティーコンテンツの実用的な使用方法を説明します。

### 8.7.1. セキュリティー脆弱性の監査 Red Hat 製品の脆弱性

Red Hat は、製品の OVAL 定義を継続的に提供します。この定義により、インストールしたソフトウェアの脆弱性の完全な監査が可能になります。このプロジェクトについての詳細は、http://www.redhat.com/security/data/metrics/ を参照してください。これらの定義をダウンロードするには、以下のコマンドを実行します。

```
~]$ wget http://www.redhat.com/security/data/oval/com.redhat.rhsa-all.xml
```

Red Hat Satellite 5 のユーザーは、パッチ定義の XCCDF の部分で有用です。これらの定義をダウンロードするには、以下のコマンドを実行します。

```
~]$ wget http://www.redhat.com/security/data/metrics/com.redhat.rhsa-all.xccdf.xml
```

システムにインストールされているソフトウェアのセキュリティー脆弱性を監査するには、以下のコマンドを実行します。

```
~]$ oscap oval eval --results rhsa-results-oval.xml --report oval-report.html com.redhat.rhsa-all.xml
```

oscap ユーティリティーは、Red Hat セキュリティーアドバイザリーを、National Vulnerability Database にリンクされた CVE 識別子にマッピングし、適用されないセキュリティーアドバイザリーを報告します。

注記

この OVAL 定義は、Red Hat がリリースするソフトウェアおよび更新のみに対応するように設計されています。サードパーティーソフトウェアのパッチステータスを検出するには、追加の定義を指定する必要があります。

### 8.7.2. SCAP セキュリティーガイドを使用したシステム設定の監査

SCAP Security Guide(SSG)プロジェクトのパッケージは scap-security-guide、Linux システム向けの最新のセキュリティーポリシーセットが含まれています。の一部 scap-security-guide は、Red Hat Enterprise Linux 6 設定のガイダンスでもあります。で利用可能なセキュリティーコンテンツを検査するに scap-security-guideは、oscap info モジュールを使用します。

```
~]$ oscap info /usr/share/xml/scap/ssg/content/ssg-rhel6-ds.xml
```

このコマンドの出力には、SSG ドキュメントの概要と、利用可能な設定プロファイルが含まれています。システム設定を監査するには、適切なプロファイルを選択して、適切な評価コマンドを実行します。たとえば、以下のコマンドを使用して、Red Hat 認定クラウドプロバイダーのドラフト SCAP プロファイルに対して、指定のシステムを評価します。

```
~]$ oscap xccdf eval --profile xccdf_org.ssgproject.content_profile_rht-ccp --results ssg-rhel6-xccdf-result.xml --report ssg-rhel6-report.html /usr/share/xml/scap/ssg/content/ssg-rhel6-ds.xml
```

### 8.8. その他のリソース

関連するさまざまなセキュリティーコンプライアンスフィールドの詳細は、以下の資料を参照してください。

インストールされているドキュメント

- **oscap(8)- oscap** コマンドラインユーティリティーの **man** ページでは、利用可能なオプションの完全なリストとその使用方法が説明されます。
- **Guide to the Secure Configuration of Red Hat Enterprise Linux 6 - XCCDF** チェックリスト形式でシステムのセキュリティー設定の詳細ガイドを提供する **/usr/share/doc/scap-security-guide-0.1.18/** ディレクトリーにある **HTML** ドキュメントです。

オンラインドキュメント

- **OpenSCAP プロジェクトページ - OpenSCAP** プロジェクトのホームページでは、**oscap** ユーティリティーと、**SCAP** に関連するその他のコンポーネントおよびプロジェクトの詳細情報が提供されています。
- **SCAP Workbench プロジェクトページ - SCAP Workbench** プロジェクトのホームページでは、**scap-workbench** アプリケーションの詳細情報が提供されています。
- **SCAP Security Guide(SSG)プロジェクトページ - SSG プロジェクト** のホームページでは、**Red Hat Enterprise Linux** 向けの最新セキュリティーコンテンツが提供されています。
- **National Institute of Standards and Technology(NIST)SCAP のページ - このページでは、SCAP** の発行、仕様、**SCAP** 検出プログラムなどの **SCAP** 関連の資料が多数提供されます。

- National Vulnerability Database(NVD) - このページは、SCAP コンテンツおよびその他の SCAP 規格ベースの脆弱性管理データに関する最大のリポジトリーです。

- Red Hat OVAL content repository - Red Hat Enterprise Linux システムの OVAL 定義を含むリポジトリーです。

- MITRE CVE - これは、MITRE corporation が提供する既知のセキュリティー脆弱性のデータベースです。

- MITRE OVAL - このページでは、MITRE corporation が提供する OVAL 関連のプロジェクトが紹介されています。OVAL の関連情報、たとえば OVAL 言語の最新バージョン、OVAL コンテンツの Huge リポジトリー、22数千以上の OVAL 定義がカウントされています。

- Red Hat Satellite ドキュメント - このガイドセットでは、OpenSCAP を使用して複数のシステムでシステムセキュリティーを維持する方法について説明しています。

# 第9章 AIDEでの整合性の確認

## 9.1. はじめに

AIDE (Advanced Intrusion Detection Environment) は、システムのファイルのデータベースを作成し、そのデータベースを使用してファイルの整合性を確保し、システムの侵入を検出します。

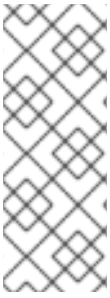
## 9.2. AIDEのインストール

aide パッケージをインストールするには、root で次のコマンドを実行します。

```
~]# yum install aide
```

初期データベースを生成するには、root で以下のコマンドを入力します。

```
~]# aide --init

AIDE, version 0.14

### AIDE database at /var/lib/aide/aide.db.new.gz initialized.
```



**注記**

デフォルト設定では、aide --init コマンドは、ファイルで定義されているディレクトリーおよびファイルのセットのみを確認し /etc/aide.conf ます。ディレクトリーまたはファイルを AIDE データベースに追加し、監視パラメーターを変更するには、適切に編集し /etc/aide.conf ます。

データベースの使用を開始するには、初期データベースのファイル名から .new サブ文字列を削除します。

```
~]# mv /var/lib/aide/aide.db.new.gz /var/lib/aide/aide.db.gz
```

AIDE データベースの場所を変更するには、/etc/aide.conf ファイルを編集して DBDIR 値を変更します。追加のセキュリティーを確保するには、データベース、設定、/usr/sbin/aide バイナリーファイルを読み取り専用メディアなどの安全な場所に保存します。

重要

AIDE データベースの場所の変更後に SELinux の拒否を回避するには、それに応じて SELinux ポリシーを更新します。詳細 は、『SELinux ユーザーおよび管理者のガイド』 を参照してください。

## 9.3. INTEGRITY チェックの実行

手動でチェックを開始するには、root で次のコマンドを実行します。

```
~]# aide --check
AIDE found differences between database and filesystem!!
Start timestamp: 2017-04-07 17:11:33

Summary:
  Total number of files: 104892
  Added files:   7
  Removed files:  0
  Changed files:  52
...
```

AIDE は、最低でも、スキャンを毎週実行するように設定する必要があります。AIDE は毎日実行する必要があります。たとえば、を *使用する AIDE を毎日 4:05* で実行する cron （『システム管理者ガイド』の「 システムタスクの自動 実行」の章を参照）、に以下の行を追加し /etc/crontabます。

```
05 4 * * * root /usr/sbin/aide --check
```

## 9.4. AIDE データベースの更新

パッケージの更新や設定ファイルの調整など、システムの変更を検証した後に、基本となる AIDE データベースを更新します。

```
~]# aide --update
```

aide --update コマンドは /var/lib/aide/aide.db.new.gz データベースファイルを作成します。整合性チェックに使用を開始するには、ファイル名から .new サブ文字列を削除します。

## 9.5. その他のリソース

AIDE の詳細は、次のドキュメントを参照してください。

- aide(1) の man ページ
- aide.conf(5) の man ページ
- Guide to the Secure Configuration of Red Hat Enterprise Linux 7(OpenSCAP Security Guide): Verify Integrity with AIDE
- AIDE マニュアル

# 第10章 電子規格および規則

## 10.1. はじめに

セキュリティーレベルを維持するために、お客様の組織はセキュリティーレベルおよび業界のセキュリティー仕様、標準仕様、規制を順守するよう努めています。本章では、これらの標準および規制の一部を説明します。

## 10.2. 連邦情報処理標準(FIPS)

FIPS(Federal Information Processing Standard)Publication 140-2 は、米国によって開発されたコンピューターセキュリティー標準です。米国および業界で作業グループで、暗号化モジュールの品質を検証します。FIPS 公開情報（140-2 を含む）は、http://csrc.nist.gov/publications/PubsFIPS.html の URL にあります。FIPS 標準は、さまざまな業界、暗号化モジュールの実装、組織サイズおよび要件に対応するために 4 つのセキュリティー レベル を提供します。これらのレベルは以下のとおりです。

- レベル 1: セキュリティーレベル 1 は、セキュリティーの最低レベルを提供します。基本的なセキュリティー要件は、暗号化モジュールに指定されます（例： 1 つ以上の承認済みアルゴリズムまたは承認済みセキュリティー機能が使用される場合など）。セキュリティーレベル 1 暗号化モジュールは、実稼働グレードコンポーネントの基本的な要件を超えて、特定の物理セキュリティーメカニズムは必要ありません。セキュリティーレベル 1 暗号化モジュールの例として、個人コンピューター(PC)暗号化が挙げられます。

- レベル 2 - セキュリティーレベル 2 は、改ざんネビデンスの要件を追加することで、セキュリティーレベル 1 暗号化モジュールの物理セキュリティーメカニズムを強化します。これには、タムパーエビデンスまたはシールドの使用や、モジュールのリムーバブル保留回復ロックの使用が含まれます。改ざん規則またはシールは暗号化モジュールに配置され、モジュール内のプレーンテキスト暗号鍵と重要なセキュリティーパラメーター(CSP)への物理アクセスのために破損する必要があります。改ざんされたシールドまたは選択回復性ロックは、承認されていない物理アクセスから保護するために保護されます。

- レベル 3: セキュリティーレベル 2 で必要な改ざんされた物理セキュリティーメカニズムに加えて、侵入者が暗号化モジュール内に保持される CSP へのアクセスを阻止しようとします。セキュリティーレベル 3 で必要となる物理セキュリティーメカニズムは、物理アクセス、使用、または修正を行う可能性が高いことが意図されています。物理セキュリティーメカニズムには、暗号モジュールのリムーバブル対象/交換が開かれるとすべてのプレーンテキスト CSP をゼロにする強固な エラーと改ざん検出/応答サーキットなどが含まれる場合があります。

- レベル 4 - セキュリティーレベル 4 は、この標準で定義されている最高のセキュリティーレベルを提供します。このセキュリティーレベルでは、物理セキュリティーメカニズムは、物理アクセスで承認されていないすべての試行を検出して応答する目的で、暗号化モジュールに関する完全な保護を提供します。あらゆる方向からの暗号化モジュールの特性に侵入すると、検出される可能性が非常に高くなり、すべてのプレーンテキストの CSP のゼロ化が非常に高くな

ります。セキュリティーレベル 4 暗号化モジュールは、物理的に保護されていない環境での操作に役立ちます。

これらのレベルと FIPS 標準の仕様 に関する 詳細情報は、http://csrc.nist.gov/publications/fips/fips140-2/fips1402.pdf の FIPS 140-2 標準仕様を参照してください。

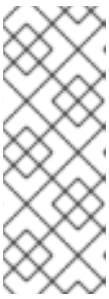
### 10.2.1. FIPS モードの有効化

Red Hat Enterprise Linux 6 を連邦情報処理標準(FIPS)公開文書 140-2 に準拠させるには、認定済みの暗号化モジュールを使用できるようにするためにいくつかの変更を加える必要があります。システム（カーネルおよびユーザー領域）を FIPS モードに変換するには、以下の手順に従います。

1. モジュール内整合性検証の適切な操作を行うには、事前リンクを無効にする必要があります。これは、設定 /etc/sysconfig/prelink ファイルにを設定すること PRELINKING=no で実行できます。既存の事前リンクがある場合は、prelink -u -a コマンドを使用してすべてのシステムファイルで元に戻す必要があります。

2. 次に、dracut-fips パッケージをインストールします。

```
~]# yum install dracut-fips
```



**注記**

FIPS 整合性検証は、システムが FIPS モードで動作するかどうかに関わらず、dracut-fips パッケージがシステムに存在するときに実行されます。ただし、システムまたは共有ライブラリーが dracut-fips 存在する場合でも、整合性検証の結果は無視（またはログに記録のみ）されます。

3. initramfs ファイルを再作成します（この操作は既存の initramfs ファイルを上書きします）。

```
~]# dracut -f
```

4. 以下のオプションを追加して、/boot/grub/grub.conf ファイルの現在のカーネルのカーネルコマンドラインを変更します。

```
fips=1
```

/boot または /boot/efi/ ディレクトリーが別のパーティションにある場合は、*boot=partition* カーネルパラメーターをカーネルコマンドラインに追加する必要があります。*partition* を、/boot または /boot/efi/ ディレクトリーを含むパーティションに置き換えます。パーティションは、df コマンドを使用して識別できます。以下に例を示します。

```
~]$ df /boot
Filesystem           1K-blocks      Used Available Use% Mounted on
/dev/sda1               495844     53780    416464  12% /boot
```

上記の例では、/boot ディレクトリーはにあり /dev/sda1ます。したがって、カーネルコマンドラインに以下の文字列を追加する必要があります。

```
boot=/dev/sda1
```

5. システムを再起動します。

デフォルトでは、暗号および Message Authentication Codes(MAC)は、FIPS モードの /etc/ssh/sshd_config ファイルに設定されていることに注意してください。に他の暗号および MAC が /etc/ssh/sshd_config 含まれている場合は、FIPS モードでサポートされるアルゴリズムのみを使用するように変更します。これを行うには、以下の設定またはその設定のサブセットを使用します。

```
Protocol 2
Ciphers aes128-ctr,aes192-ctr,aes256-ctr,aes128-cbc,3des-cbc,aes192-cbc,aes256-cbc
Macs hmac-sha1,hmac-sha2-256,hmac-sha2-512
```

厳密な FIPS コンプライアンスが必要な場合は、システムインストール時に fips=1 カーネルオプションをカーネルコマンドラインに追加し、FIPS 承認アルゴリズムと継続的な監視テストで鍵の生成を行う必要があります。また、インストールプロセス中にマウスを移動するか、マウスが利用できない場合は多くのキートロピーが設定されるようにすることで、システムに多くのエントロピーが使用されるようにする必要があります。キー操作の推奨される量は 256 以上です。256 未満のキーにより、一意の鍵が生成される可能性があります。

### 10.2.2. NSS を使用したアプリケーションの FIPS モードの有効化

で説明されている Red Hat Enterprise Linux システムで FIPS モードを有効にする手順は、ネットワークセキュリティーサービス(NSS)の FIPS 状態には影響を与え 「FIPS モードの有効化」 ないため、NSS を使用するアプリケーションには影響しません。必要に応じて、以下のコマンドを使用して NSS アプリケーションを FIPS モードに切り替えることができます。

```
~]# modutil -fips true -dbdir dir
```

*dir* を、アプリケーションが使用する NSS データベースを指定するディレクトリーに置き換えます。複数の NSS アプリケーションがこのデータベースを使用する場合、これらのすべてのアプリケーションがすべて FIPS モードに切り替わります。NSS FIPS モードを有効にするには、アプリケーションを再起動する必要があります。

nss-sysinit パッケージがインストールされ、検索する必要のある NSS データベースが開かれる場合は /etc/pki/nssdb、ユーザーの NSS データベースへのパスがになり ~/.pki/nssdbます。

Firefox Web ブラウザーおよび Thunderbird メールクライアントの FIPS モードを有効にするには、に進み Edit → Preferences → Advanced → Certificates → Security Devices → Enable FIPSます。

### 10.3. NISPOM (NATIONAL INDUSTRIAL SECURITY PROGRAM OPERATING MANUAL)

NISPOM（DoD 5220.22-M とも呼ばれます）は、NISPOM(National Industrial Security Program)のコンポーネントとして、機密情報に関して、すべての米国の契約者に対して一連の手順と要件を確立します。現在の NISPOM は、2013 年 3 月から主な変更点と共に 2 年 2 月に記載されています。NISPOM ドキュメントは、http://www.nispom.org/NISPOM-download.html の URL からダウンロードできます。

### 10.4. PCI DSS(PAYMENT CARD INDUSTRY DATA SECURITY STANDARD)

https://www.pcisecuritystandards.org/about/index.shtml から： PCI セキュリティー標準は、Data Security Standard(DSS)を含む PCI セキュリティー標準の開発、管理、および認識を担当する、オープングローバルなフォークです。

PCI DSS 標準は、https://www.pcisecuritystandards.org/security_standards/pci_dss.shtml からダウンロードできます。

### 10.5. セキュリティー技術実装ガイド

セキュリティー技術実装ガイド(STIG)は、コンピューターソフトウェアおよびハードウェアの標準化された安全なインストールとメンテナンスを行うための手法です。

STIG の詳細は、公式の IASE の Web サイト を参照してください。

# 第11章 リファレンス

以下の参考は、SELinux および Red Hat Enterprise Linux に関連し、本ガイドの範囲外にある追加情報へのポインターです。SELinux の迅速な開発により、この資料の一部は Red Hat Enterprise Linux の特定のリリースにのみ適用される可能性があることに注意してください。

## ブラジル

### SELinux の例

#### mayer、MacMillan、および Caplan

#### 前提条件

## チュートリアルとヘルプ

### Chrisk Coker のチュートリアルと対話

http://www.coker.com.au/selinux/talks/ibmtu-2004/

### 一般的な SELinux ポリシー HOWTO

http://www.lurking-grue.org/writingselinuxpolicyHOWTO.html

### Red Hat ナレッジベース

https://access.redhat.com/knowledgebase

## 全般的な情報

### NSA SELinux メインの Web サイト

http://www.nsa.gov/selinux/

### SELinux NSA のオープンソースセキュリティーの強化

http://www.oreilly.com/catalog/selinux/

## テクノロジー

セキュリティーポリシーの Flexible Support の統合（Linux における Flask 実装に関する履歴）

http://www.nsa.gov/research/_files/selinux/papers/selsymp2005.pdf

Security Enhanced Linux のセキュリティーポリシー設定

http://www.nsa.gov/research/_files/selinux/papers/policy/policy.shtml

コミュニティー

SELinux コミュニティーページ

http://selinuxproject.org/

IRC

irc.freenode.net, #selinux, #fedora-selinux, #security

履歴

Flask のクイック履歴

http://www.cs.utah.edu/flux/fluke/html/flask.html

Fluke のフル背景

http://www.cs.utah.edu/flux/fluke/html/index.html

# 付録A 暗号化の標準

## A.1. 同期暗号化

### A.1.1. 高度な暗号化標準 - AES

暗号化では、Advanced Encryption Standard(AES)は U.S が採用する暗号化標準です。ロシアです。標準は、AES-128、AES-192、および AES-256 の 3 つのブロック暗号で構成されており、当初より大きなコレクションから採用されています。各 AES 暗号は 128 ビットのブロックサイズを持ち、鍵サイズは 128 ビット、192 ビットおよび 256 ビットになります。AES 暗号は広範囲に分析され、以前のデータ暗号化標準(DES)と同様に、現在使用されるようになりました。[5]

#### A.1.1.1. AES 履歴

AES は National Institute of Standards and Technology(NIST)から U.S として発表されました。2001 年 11 月 11 日に FIPS PUB 197(FIPS 197)は、5 年間の標準化プロセス後に提供されます。競合する設計が提示され、最も適したとおりに、Jandndael が選択されている前に評価されました。これは標準的な 5 月 1 日に有効になりました。これは、多くの異なる暗号化パッケージで利用できます。AES は、最初に一般に公開され、オープンされた暗号です。NSA により、シークレット情報が上書されます。

この暗号は、Jolandael の 2 つの暗号である Joanleeemen と Vincentíjmen によって開発され、AES 選択プロセスに提出されました。westjndael は 2 つの法名のポートネームです。[6]

### A.1.2. データ暗号化標準 - DES

Data Encryption Standard(DES)は、米国の公式の連邦情報処理標準(FIPS)として、このブロック暗号（共有秘密の暗号化形式）で選択され、その後国際的に使用されています。これは、56 ビットキーを使用する対称キーアルゴリズムに基づいています。このアルゴリズムは当初、分類された設計要素、比較的短いキーの長さ、NSA バックティブ(NSA)バックスピックに関するものです。このため、DES は、ブロック暗号とその暗号に関する最新の理解を促進する強力なスクラニティブに発生しました。[7]

#### A.1.2.1. DES 履歴

DES は、多くのアプリケーションで安全ではないと見なされています。これは、56 ビットの鍵のサイズが小さすぎるためかなり小さくなり、1 月、distributed.net、およびをとります（22 時間と 15 分で DES キーが破損するように促しました）。また、暗号で理論的なリーダーを実証する分析結果もいくつかありますが、実際にはマウントすることができません。このアルゴリズムは、理論的な攻撃は存在するものの、実際には Triple DES 形式で安全であると考えられています。最近のところ、暗号は高度な暗号化標準(AES)に置き換えられています。[8]

一部のドキュメントでは、DES を標準と DES として区別し、DEA（データ暗号化アルゴリズム）と呼ばれるアルゴリズムを使用します。[9]

### A.2. 公開鍵暗号化

公開鍵暗号は、多くの暗号化アルゴリズムおよび暗号システムが使用されている暗号化アプローチです。特性は対称鍵アルゴリズムの代わりに、非対称鍵アルゴリズムを使用することです。公開鍵/秘密鍵暗号法を使用することで、通信や、以前は不明なメッセージを認証する数多くの手法が実用的になりました。対称鍵アルゴリズムを使用する場合に必要な、1 つ以上の秘密鍵をセキュアに交換する必要はありません。また、デジタル署名を作成することもできます。[10]

パブリックキー暗号は、あらゆるあらゆる基盤で広く使用されている技術であり、Transport Layer Security(TLS)、PGP、および GPG などのインターネット標準を提供するアプローチです。[11]

公開鍵暗号で使用される区別技術は、非対称鍵アルゴリズムを使用することです。ここでは、メッセージを暗号化するために使用される鍵は、復号化に使用される鍵とは異なります。各ユーザーには、公開鍵と秘密鍵という鍵のペアがあります。秘密鍵は秘密に保持され、公開鍵は広く配布される場合があります。メッセージは受信者の公開鍵で暗号化され、対応する秘密鍵でしか復号化できません。このキーはかなりの関連性がありますが、秘密鍵は公開鍵から派生する実際に使用する（実際の方法または展開されたプラクティス）とは限りません。このアルゴリズムは、暗号化の慣行を2017年中央に開始させるようなアルゴリズムで発見されました。[12]

これとは対照的に、対称キーアルゴリズムは数年にわたり使用したもので、送信側と受信側で共有されている単一の秘密鍵を使用します（これはプライベートに維持する必要があるため、暗号化と復号化の両方に共通用語の曖昧さを考慮します）。対称暗号化スキームを使用するには、送信側および受信側が事前に鍵を安全に共有する必要があります。[13]

対称キーアルゴリズムは常に計算的に集中型ではないため、鍵交換アルゴリズムを使用して鍵を交換し、その鍵と対称鍵アルゴリズムを使用してデータを送信することが一般的です。PGP および SSL/TLS ファミリーのスキームは、たとえばこれを行い、結果としてハイブリッド暗号システムと呼ばれます。[14]

#### A.2.1. Diffie-Hellman

Diffie-Hellman key exchange(D-H)は、相互の事前知識のない 2 者が、安全ではない通信チャネルで共有秘密キーを共同で確立できるようにする暗号化プロトコルです。その後、このキーを使用して対称鍵暗号を使用して後続の通信を暗号化できます。[15]

##### A.2.1.1. Diffie-Hellman 履歴

このスキームは最初に、Justfield Diffie と Boston Hellman によって最初に公開されましたが、後で GCHQ 内で別途無効になっていると判断しましたが、このスキームは GCHQ 内で別々に展開されましたが、これは分類されたままでした。Hellman は、Rackson Merkle の判断で、公開鍵暗号(Hellman,1971)にアルゴリズムを Diffie-Hellman-Merkle キー交換と呼び出すことを推奨しています。[16]

Diffie-Hellman キーアグリーメント自体は匿名の（認証されていない）キーアグルーメントプロトコルですが、認証されていないプロトコルのベースとなります。これは、Transport Layer Security の一時モードで（暗号スイートに応じて EDH または DHE と呼ばれます）。[17]

S.現在は期限切れになった 4,200,770 は、アルゴリズムとクレジットの Hellman、Diffie、および Merkle について説明しています。[18]

### A.2.2. RSA

暗号では、RSA（Rivest、Shellir、および Adleman が最初に説明した）が、公開鍵暗号のアルゴリズムです。これは、暗号化の署名、および署名が適切であると認識される最初のアルゴリズムであり、公開鍵暗号における最初の前例の 1 つです。RSA は電子商取引プロトコルで広く使用され、十分な長い鍵と最新の実装の使用が十分に確保されていると考えられます。

### A.2.3. DSA

DSA(Digital Signature Algorithm)は、デジタル署名の標準である、米国のデジタル署名に関する標準です。DSA は署名のみを目的としており、暗号化アルゴリズムではありません。[19]

### A.2.4. SSL/TLS

Transport Layer Security(TLS)およびその以前の Secure Sockets Layer(SSL)は、インターネットなどのネットワークを介した通信のセキュリティーを提供する暗号化プロトコルです。TLS および SSL は、Transport Layer end-to-end でネットワーク接続のセグメントを暗号化します。

いくつかのバージョンのプロトコルは、Web ブラウジ、電子電子メール、インターネットの伝送、インスタントメッセージング、音声オーバー IP(VoIP)などのアプリケーションで広く使用されています。[20]

### A.2.5. Cramer-Shoup Cryptosystem

Cramer-Shoup システムは非対称鍵暗号化アルゴリズムで、標準の暗号仮定を使用して適応可能な

暗号文攻撃に対して安全であることが証明されています。このセキュリティーは、決定的な Diffie-Hellman 仮定の計算の不整合性（全体的に仮定される訳ではありません）に基づいています。Ronald Cramer(Kronald Cramer)と、Emily Shoup によって開発され、ElasticGamal 暗号の拡張です。Cramer-Shoup は、非常に適切ではないElasticsearchGamal とは対照的に、リソースを持つ攻撃者に対しても、誤作動性を確保するために追加の要素を追加します。この非定型性は、競合に依存しないハッシュ関数と追加の計算を使用して実現します。その結果、ElIGamal の容量が 2 回ある暗号文になります。[21]

### A.2.6. ElasticsearchGamal Encryption

暗号では、ElGamal 暗号化システムは、Diffie-Hellman 鍵契約に基づく公開鍵暗号に対する非対称鍵暗号化アルゴリズムです。これは Taher Emailgamal によって記載されました。ErrataGamal 暗号化は、無料の GNU Privacy Guard ソフトウェア、最新バージョンの PGP、およびその他の暗号システムで使用されます。[22]

---

[5] 「高度な暗号化標準」 ブラジル。2009 年 11 月 14 日 http://en.wikipedia.org/wiki/Advanced_Encryption_Standard

[6] 「高度な暗号化標準」 ブラジル。2009 年 11 月 14 日 http://en.wikipedia.org/wiki/Advanced_Encryption_Standard

[7] 「データ暗号化規格」 ブラジル。2009 年 11 月 14 日 http://en.wikipedia.org/wiki/Data_Encryption_Standard

[8] 「データ暗号化規格」 ブラジル。2009 年 11 月 14 日 http://en.wikipedia.org/wiki/Data_Encryption_Standard

[9] 「データ暗号化規格」 ブラジル。2009 年 11 月 14 日 http://en.wikipedia.org/wiki/Data_Encryption_Standard

[10] 「公開鍵暗号化」 ブラジル。2009 年 11 月 14 日 http://en.wikipedia.org/wiki/Public-key_cryptography

[11] 「公開鍵暗号化」 ブラジル。2009 年 11 月 14 日 http://en.wikipedia.org/wiki/Public-key_cryptography

[12]
「公開鍵暗号化」 ブラジル。2009 年 11 月 14 日 http://en.wikipedia.org/wiki/Public-key_cryptography

[13]
「公開鍵暗号化」 ブラジル。2009 年 11 月 14 日 http://en.wikipedia.org/wiki/Public-key_cryptography

[14]
「公開鍵暗号化」 ブラジル。2009 年 11 月 14 日 http://en.wikipedia.org/wiki/Public-key_cryptography

[15]
"Diffie-Hellman." ブラジル。2009 年 11 月 14 日 http://en.wikipedia.org/wiki/Diffie-Hellman

[16]
"Diffie-Hellman." ブラジル。2009 年 11 月 14 日 http://en.wikipedia.org/wiki/Diffie-Hellman

[17]
"Diffie-Hellman." ブラジル。2009 年 11 月 14 日 http://en.wikipedia.org/wiki/Diffie-Hellman

[18]
"Diffie-Hellman." ブラジル。2009 年 11 月 14 日 http://en.wikipedia.org/wiki/Diffie-Hellman

[19]
"DSA." ブラジル。24 February 2010 http://en.wikipedia.org/wiki/Digital_Signature_Algorithm

[20]
"TLS/SSL." ブラジル。24 February 2010 http://en.wikipedia.org/wiki/Transport_Layer_Security

[21]
"Cramer-Shoup cryptosystem" ブラジル。2010 年 2 月 24 日 http://en.wikipedia.org/wiki/Cramer-Shoup_cryptosystem

[22]
エラーメッセージ「ElGamal encryption 」。2010 年 2 月 24 日 http://en.wikipedia.org/wiki/ElGamal_encryption

# 付録B AUDIT システムのリファレンス

## B.1. 監査イベントフィールド

表B.1「イベントフィールド」 現在サポートしている Audit イベントフィールドをすべて表示します。イベントフィールドは、Audit ログファイルの等号の前にある値です。

表B.1 イベントフィールド

| イベントフィールド | 説明 |
| --- | --- |
| **a0**, **a1**, **a2**, **a3** | 16 進数表記でエンコードされた、システムコールの最初の 4 つの引数を記録します。 |
| **acct** | ユーザーのアカウント名を記録します。 |
| **addr** | IPv4 アドレスまたは IPv6 アドレスを記録します。このフィールドは、通常、に従います。 **hostname** フィールドで、ホスト名が解決するアドレスが含まれます。 |
| **arch** | システムの CPU アーキテクチャーに関する情報を記録します。16 進数表記でエンコードされます。 |
| **auid** | Audit ユーザー ID を記録します。この ID は、ログイン時にユーザーに割り当てられ、ユーザーのアイデンティティーが変更される場合でも（たとえば、でユーザーアカウントを切り替えて **su - john**）すべてのプロセスによって継承されます。 |
| **capability** | 特定の Linux 機能の設定に使用されたビット数を記録します。Linux 機能の詳細は、を参照してください。 機能(7) の man ページ。 |
| **cap_fi** | 継承されたファイルシステムベースの機能の設定に関連するデータを記録します。 |
| **cap_fp** | 許可されたファイルシステムベースの機能の設定に関連するデータを記録します。 |
| **cap_pe** | 効果的なプロセスベースの機能の設定に関連するデータを記録します。 |
| **cap_pi** | 継承されたプロセスベースの機能の設定に関連するデータを記録します。 |
| **cap_pp** | 許可されたプロセスベースの機能の設定に関連するデータを記録します。 |
| **cgroup** | Audit イベントの生成時にプロセス **cgroup** が含まれるへのパスを記録します。 |

| イベントフィールド | 説明 |
| --- | --- |
| **cmd** | 実行されたコマンドライン全体を記録します。これは、たとえば、シェルインタープリター **/bin/bash** として、**exe** フィールドが記録するシェルインタープリターの場合や、**cmd** フィールドは実行される残りのコマンドラインを記録し **helloworld.sh --help**ます。 |
| **comm** | 実行したコマンドを記録します。これは、たとえば、シェルインタープリター **/bin/bash** として、**exe** フィールドが記録するシェルインタープリターの場合や、**comm** フィールドは実行されるスクリプトの名前を記録し **helloworld.sh**ます。 |
| **cwd** | システムコールが開始したディレクトリーへのパスを記録します。 |
| **data** | TTY レコードに関連付けられたデータを記録します。 |
| **dev** | イベントに記録されたファイルまたはディレクトリーが含まれるデバイスのマイナー ID とメジャー ID を記録します。 |
| **devmajor** | メジャーデバイス ID を記録します。 |
| **devminor** | マイナーデバイス ID を記録します。 |
| **egid** | 解析しているプロセスを開始したユーザーの実効グループ ID を記録します。 |
| **euid** | 解析しているプロセスを開始したユーザーの実効ユーザー ID を記録します。 |
| **exe** | 解析しているプロセスを開始するために使用した実行可能ファイルへのパスを記録します。 |
| **exit** | システムコールが返した終了コードを記録します。この値はシステムコールによって異なります。以下のコマンドを使用すると、この値を人間が判読可能なものに変換できます。 **ausearch --interpret --exit *exit_code*** |
| **family** | IPv4 または IPv6 のいずれかで使用されたアドレスプロトコルのタイプを記録します。 |
| **filetype** | ファイルのタイプを記録します。 |
| **flags** | ファイルシステムの名前フラグを記録します。 |
| **fsgid** | 解析しているプロセスを開始したユーザーのファイルシステムグループ ID を記録します。 |
| **fsuid** | 解析しているプロセスを開始したユーザーのファイルシステムユーザー ID を記録します。 |

| イベントフィールド | 説明 |
| --- | --- |
| **gid** | グループ ID を記録します。 |
| **hostname** | ホスト名を記録します。 |
| **icmptype** | 受信したインターネット制御メッセージプロトコル(ICMP)パッケージのタイプを記録します。このフィールドを含む監査メッセージは、**iptables** により生成されます。 |
| **id** | 変更されたアカウントのユーザー ID を記録します。 |
| **inode** | Audit イベントに記録されたファイルまたはディレクトリーに関連付けられた inode 番号を記録します。 |
| **inode_gid** | inode の所有者のグループ ID を記録します。 |
| **inode_uid** | inode の所有者のユーザー ID を記録します。 |
| **items** | このレコードに接続されているパスレコードの数を記録します。 |
| **key** | Audit ログで特定のイベントを生成したルールに関連付けられたユーザー定義の文字列を記録します。 |
| **list** | Audit ルールリスト ID を記録します。既知の ID の一覧は以下のとおりです。<br>● 0 – **user**<br>● 1 – **task**<br>● 4 – **exit**<br>● 5 – **exclude** |
| **mode** | ファイルまたはディレクトリーのパーミッションを、数字表記でエンコードします。 |
| **msg** | レコードのタイムスタンプと一意の ID、またはカーネルまたはユーザー空間アプリケーションが提供するさまざまなイベント固有の ***<name>=<value>*** ペアを記録します。 |
| **msgtype** | ユーザーベースの AVC 拒否が発生した場合に返されるメッセージタイプを記録します。メッセージタイプは D-Bus によって決定されます。 |
| **name** | システムコールに引数として渡されたファイルまたはディレクトリーの完全パスを記録します。 |
| **new-disk** | 仮想マシンに割り当てられた新規ディスクリソースの名前を記録します。 |

| イベントフィールド | 説明 |
| --- | --- |
| **new-mem** | 仮想マシンに割り当てられた新規メモリーリソースの量を記録します。 |
| **new-vcpu** | 仮想マシンに割り当てられた新規仮想 CPU リソースの数を記録します。 |
| **new-net** | 仮想マシンに割り当てられた新規ネットワークインターフェースリソースの MAC アドレスを記録します。 |
| **new_gid** | ユーザーに割り当てられているグループ ID を記録します。 |
| **oauid** | （を使用などではなく **su**）システムにログインしているユーザーのユーザー ID を記録し、ターゲットプロセスを開始している。このフィールドはタイプのレコードのみに限定されます。 **OBJ_PID**. |
| **ocomm** | ターゲットプロセスを開始するために使用したコマンドを記録します。このフィールドはタイプのレコードのみに限定されます。 **OBJ_PID**. |
| **opid** | ターゲットプロセスのプロセス ID を記録します。このフィールドはタイプのレコードのみに限定されます。 **OBJ_PID**. |
| **oses** | ターゲットプロセスのセッション ID を記録します。このフィールドはタイプのレコードのみに限定されます。 **OBJ_PID**. |
| **ouid** | ターゲットプロセスの実際のユーザー ID を記録します。 |
| **obj** | オブジェクトの SELinux コンテキストを記録します。オブジェクトは、ファイル、ディレクトリー、ソケット、またはサブジェクトのアクションを受信するものになります。 |
| **obj_gid** | オブジェクトのグループ ID を記録します。 |
| **obj_lev_high** | オブジェクトの SELinux レベルの高いものを記録します。 |
| **obj_lev_low** | オブジェクトの低い SELinux レベルを記録します。 |
| **obj_role** | オブジェクトの SELinux ロールを記録します。 |
| **obj_uid** | オブジェクトの UID を記録します。 |
| **obj_user** | オブジェクトに関連付けられたユーザーを記録します。 |
| **ogid** | オブジェクトの所有者のグループ ID を記録します。 |
| **old-disk** | 新規ディスクリソースが仮想マシンに割り当てられている場合に、古いディスクリソースの名前を記録します。 |

| イベントフィールド | 説明 |
| --- | --- |
| **old-mem** | 新しいメモリーが仮想マシンに割り当てられている場合の古いメモリーリソースの量を記録します。 |
| **old-vcpu** | 新規仮想 CPU が仮想マシンに割り当てられている場合の古い仮想 CPU リソースの数を記録します。 |
| **old-net** | 新しいネットワークインターフェースが仮想マシンに割り当てられている場合に、古いネットワークインターフェースリソースの MAC アドレスを記録します。 |
| **old_prom** | ネットワーク promiscuity フラグの以前の値を記録します。 |
| **ouid** | ターゲットプロセスを開始したユーザーの実際のユーザー ID を記録します。 |
| **path** | AVC 関連の Audit イベントの場合に、システムコールに渡されたファイルまたはディレクトリーの完全パスを記録します。 |
| **perm** | イベントの生成に使用したファイルパーミッション（読み取り、書き込み、実行、または属性の変更）を記録します。 |
| **pid** | The **pid** フィールドセマンティクスは、このフィールドの値の起点によって異なります。<br><br>ユーザー空間から生成されるフィールドでは、このフィールドはプロセス ID を保持します。<br><br>カーネルによって生成されるフィールドでは、このフィールドはスレッド ID を保持します。スレッド ID は、シングルスレッドプロセスのプロセス ID と同じです。このスレッド ID の値は、の値とは異なることに注意してください。 pthread_t ユーザー空間で使用される ID。詳細はを参照してください。 gettid(2) の man ページ。 |
| **ppid** | 親プロセス ID(PID)を記録します。 |
| **prom** | ネットワーク promiscuity フラグを記録します。 |
| **proto** | 使用されたネットワークプロトコルを記録します。このフィールドは、**iptables** によって生成される監査イベントに固有のフィールドです。 |
| **res** | Audit イベントを開始した操作の結果を記録します。 |
| **result** | Audit イベントを開始した操作の結果を記録します。 |
| **saddr** | ソケットアドレスを記録します。 |
| **sauid** | 送信者の Audit ログインユーザー ID を記録します。カーネルは元のユーザーしか送信していないため、この ID は D-Bus によって提供され **auid** ます。 |

| イベントフィールド | 説明 |
| --- | --- |
| **ses** | 解析しているプロセスが開始したセッションのセッション ID を記録します。 |
| **sgid** | 解析しているプロセスを開始したユーザーのセットグループ ID を記録します。 |
| **sig** | プログラムが異常終了させるシグナルの数を記録します。通常、これはシステムの侵入の署名です。 |
| **subj** | サブジェクトの SELinux コンテキストを記録します。サブジェクトは、プロセス、ユーザー、またはオブジェクトの動作のいずれかになります。 |
| **subj_clr** | サブジェクトの SELinux クリアーを記録します。 |
| **subj_role** | サブジェクトの SELinux ロールを記録します。 |
| **subj_sen** | サブジェクトの SELinux の機密性を記録します。 |
| **subj_user** | サブジェクトに関連するユーザーを記録します。 |
| **success** | システムコールが成功したかどうかを記録します。 |
| **suid** | 解析しているプロセスを開始したユーザーのセットユーザー ID を記録します。 |
| **syscall** | カーネルに送信されたシステムコールのタイプを記録します。 |
| **terminal** | ターミナル名を記録します（なし **/dev/**）。 |
| **tty** | 制御ターミナルの名前を記録します。この値 **(none)** は、プロセスに制御ターミナルがない場合に使用されます。 |
| **uid** | 解析しているプロセスを開始したユーザーの実際のユーザー ID を記録します。 |
| **vm** | Audit イベントの発信元となる仮想マシンの名前を記録します。 |

## B.2. 監査レコードタイプ

**表B.2「レコードタイプ」 現在サポートしているすべてのタイプの Audit レコードを一覧表示します。イベントタイプは、すべての Audit レコードの最初にある type= フィールドに指定されます。**

**表B.2 レコードタイプ**

| イベントタイプ | 説明 |
| --- | --- |
| **ADD_GROUP** | ユーザーグループが追加されるとトリガーされます。 |
| **ADD_USER** | ユーザー空間ユーザーアカウントが追加されるとトリガーされます。 |
| **ANOM_ABEND**[a] | プロセスが異常終了すると発生します（有効にされている場合はコアダンプの原因となるシグナル）。 |
| **ANOM_ACCESS_FS**[a] | ファイルまたはディレクトリーアクセスが異常終了するとトリガーされます。 |
| **ANOM_ADD_ACCT**[a] | ユーザースペースアカウントの追加が異常終了するとトリガーされます。 |
| **ANOM_AMTU_FAIL**[a] | Abstract Machine Test ユーティリティー(AMTU)の障害が検出されるとトリガーされます。 |
| **ANOM_CRYPTO_FAIL**[a] | 暗号化システムの障害が検出されるとトリガーされます。 |
| **ANOM_DEL_ACCT**[a] | ユーザースペースアカウントの削除が異常終了するとトリガーされます。 |
| **ANOM_EXEC**[a] | ファイルの実行が異常終了するとトリガーされます。 |
| **ANOM_LOGIN_ACCT**[a] | アカウントログインの試行が異常終了するとトリガーされます。 |
| **ANOM_LOGIN_FAILURES**[a] | 失敗したログイン試行の制限に達するとトリガーされます。 |
| **ANOM_LOGIN_LOCATION**[a] | 禁止されている場所からログインを試みるとトリガーされます。 |
| **ANOM_LOGIN_SESSIONS**[a] | ログインの試行が同時セッションの最大量に達するとトリガーされます。 |
| **ANOM_LOGIN_TIME**[a] | による無効化時に、ログイン試行が 1 回行われるとトリガーされます（例：**pam_time** ）。 |
| **ANOM_MAX_DAC**[a] | Discretionary Access Control(DAC)の障害の最大量に達するとトリガーされます。 |
| **ANOM_MAX_MAC**[a] | Mandatory Access Control(MAC)の最大失敗量に達するとトリガーされます。 |
| **ANOM_MK_EXEC**[a] | ファイルの実行後にトリガーされます。 |
| **ANOM_MOD_ACCT**[a] | ユーザー空間アカウントの変更が異常終了するとトリガーされます。 |

| イベントタイプ | 説明 |
| --- | --- |
| **ANOM_PROMISCUOUS**[a] | デバイスがプロミスキャスモードを有効または無効にするとトリガーされます。 |
| **ANOM_RBAC_FAIL**[a] | ロールベースアクセス制御(RBAC)の自己テスト失敗が検出されるとトリガーされます。 |
| **ANOM_RBAC_INTEGRITY_FAIL**[a] | ロールベースアクセス制御(RBAC)ファイルの整合性テストの失敗が検出されるとトリガーされます。 |
| **ANOM_ROOT_TRANS**[a] | ユーザーが root になるとトリガーされます。 |
| **AVC** | SELinux パーミッションチェックの記録をトリガーされました。 |
| **AVC_PATH** | SELinux パーミッションチェックの発生時に **dentry** と **vfsmount** のペアを記録するためにトリガーされます。 |
| **BPRM_FCAPS** | ユーザーがファイルシステム機能でプログラムを実行するとトリガーされます。 |
| **CAPSET** | プロセスベースの機能の変更を記録します。 |
| **CHGRP_ID** | ユーザー名グループ ID が変更されるとトリガーされます。 |
| **CHUSER_ID** | ユーザー空間のユーザー ID が変更された場合にトリガーされます。 |
| **CONFIG_CHANGE** | Audit システム設定が変更されたときにトリガーされます。 |
| **CRED_ACQ** | ユーザーがユーザー空間の認証情報を取得するとトリガーされます。 |
| **CRED_DISP** | ユーザーがユーザー空間の認証情報を破棄するとトリガーされます。 |
| **CRED_REFR** | ユーザーがユーザー空間の認証情報を更新するとトリガーされます。 |
| **CRYPTO_FAILURE_USER** | 暗号化を復号化、暗号化、またはランダム化できない場合にトリガーされます。 |
| **CRYPTO_KEY_USER** | 暗号化に使用される暗号鍵 ID を記録するためにトリガーされます。 |
| **CRYPTO_LOGIN** | 暗号化担当者のログイン試行が検出されるとトリガーされます。 |
| **CRYPTO_LOGOUT** | 暗号担当者のログアウトの試行が検出されるとトリガーされます。 |
| **CRYPTO_PARAM_CHANGE_USER** | 暗号化パラメーターの変更が検出されるとトリガーされます。 |

| イベントタイプ | 説明 |
| --- | --- |
| **CRYPTO_REPLAY_USER** | 再生攻撃が検出されるとトリガーされます。 |
| **CRYPTO_SESSION** | TLS セッション確立中に設定したパラメーターを記録するためにトリガーされます。 |
| **CRYPTO_TEST_USER** | FIPS-140 標準規格に必要な暗号化テスト結果を記録するためにトリガーされました。 |
| **CWD** | 現在の作業ディレクトリーを記録するためにトリガーされます。 |
| **DAC_CHECK** | DAC のチェック結果を記録するためにトリガーされました。 |
| **DAEMON_ABORT** | エラーによりデーモンが停止したときにトリガーされます。 |
| **DAEMON_ACCEPT** | **auditd** デーモンがリモート接続を受け入れるとトリガーされます。 |
| **DAEMON_CLOSE** | **auditd** デーモンがリモート接続を閉じるとトリガーされました。 |
| **DAEMON_CONFIG** | デーモン設定の変更が検出されるとトリガーされます。 |
| **DAEMON_END** | デーモンが正常に停止するとトリガーされます。 |
| **DAEMON_RESUME** | **auditd** デーモンがログを再開したときにトリガーされます。 |
| **DAEMON_ROTATE** | **auditd** デーモンが Audit ログファイルをローテーションする際にトリガーされます。 |
| **DAEMON_START** | **auditd** デーモンが起動するとトリガーされます。 |
| **DEL_GROUP** | ユーザーグループが削除されるとトリガー |
| **DEL_USER** | ユーザー空間ユーザーが削除されるとトリガー |
| **DEV_ALLOC** | デバイスの割り当て時にトリガーされます。 |
| **DEV_DEALLOC** | デバイスの割り当てが解除されるとトリガーされます。 |
| **EOE** | マルチレコードイベントの最後を記録するためにトリガーされます。 |
| **EXECVE** | **execve(2)** システムコールの引数を記録するためにトリガーされました。 |
| **FD_PAIR** | システムコールの **パイプ** およびソケットペアの使用を記録するために **トリガー** されます。 |

| イベントタイプ | 説明 |
| --- | --- |
| **FS_RELABEL** | ファイルシステムの再ラベル操作が検出されるとトリガーされます。 |
| **GRP_AUTH** | グループパスワードを使用してユーザー空間グループに対する認証を行うとトリガーされます。 |
| **INTEGRITY_DATA**[b] | カーネルにより実行されるデータ整合性検証イベントを記録するためにトリガーされます。 |
| **INTEGRITY_HASH**[b] | カーネルが実行するハッシュタイプの整合性検証イベントを記録するためにトリガーされます。 |
| **INTEGRITY_METADATA**[b] | カーネルにより実行されるメタデータ整合性の検証イベントを記録するためにトリガーされます。 |
| **INTEGRITY_PCR**[b] | PCR(Platform Configuration Register)の無効化メッセージを記録するためにトリガーされます。 |
| **INTEGRITY_RULE**[b] | ポリシールールを記録するためにトリガーされました。 |
| **INTEGRITY_STATUS**[b] | トリガーされ、整合性の検証のステータスを記録します。 |
| **IPC** | システムコールによって参照される Inter-Process Communication オブジェクトに関する情報を記録するためにトリガーされます。 |
| **IPC_SET_PERM** | によって設定された新規値に関する情報を記録するためにトリガーされます。 **IPC_SET** IPC オブジェクトの制御操作。 |
| **KERNEL** | Audit システムの初期化を記録するためにトリガーされます。 |
| **KERNEL_OTHER** | サードパーティーカーネルモジュールからの情報を記録するためにトリガーされます。 |
| **LABEL_LEVEL_CHANGE** | オブジェクトのレベルラベルが変更されるとトリガーされます。 |
| **LABEL_OVERRIDE** | 管理者がオブジェクトのレベルラベルを上書きするとトリガーされます。 |
| **LOGIN** | ユーザーがシステムにアクセスする際に、関連するログイン情報を記録するためにトリガーされます。 |
| **MAC_CIPSOV4_ADD** | Commercial Internet Protocol Security Option(CIPSO)ユーザーが新しい Domain of Interpretation(DOI)を追加するとトリガーされます。DOI の追加は、NetLabel が提供するカーネルのパケットラベル機能の一部です。 |
| **MAC_CIPSOV4_DEL** | CIPSO ユーザーが既存の DOI を削除したときにトリガーされます。DOI の追加は、NetLabel が提供するカーネルのパケットラベル機能の一部です。 |

| イベントタイプ | 説明 |
| --- | --- |
| **MAC_CONFIG_CHANGE** | SELinux のブール値が変更されたときにトリガーされます。 |
| **MAC_IPSEC_EVENT** | IPsec イベントに関する情報を記録するためにトリガーされます。検出されると、または IPsec 設定の変更時にトリガーされます。 |
| **MAC_MAP_ADD** | 新しい Linux Security Module(LSM)ドメインマッピングが追加されるとトリガーされます。LSM ドメインマッピングは、NetLabel が提供するカーネルのパケットラベル機能の一部です。 |
| **MAC_MAP_DEL** | 既存の LSM ドメインマッピングが追加されるとトリガーされます。LSM ドメインマッピングは、NetLabel が提供するカーネルのパケットラベル機能の一部です。 |
| **MAC_POLICY_LOAD** | SELinux ポリシーファイルが読み込まれるとトリガーされます。 |
| **MAC_STATUS** | SELinux モード（強制、Permissive、オフ）が変更されたときにトリガーされます。 |
| **MAC_UNLBL_ALLOW** | NetLabel が提供するカーネルのパケットラベル機能を使用する際に、ラベルが解除されたトラフィックが許可されるとトリガーされます。 |
| **MAC_UNLBL_STCADD** | NetLabel が提供するカーネルのパケットラベル機能を使用すると、静的ラベルが追加されるとトリガーされます。 |
| **MAC_UNLBL_STCDEL** | NetLabel が提供するカーネルのパケットラベル機能を使用する際に静的ラベルが削除されるとトリガーされます。 |
| **MMAP** | **mmap(2)** システムコールのファイル記述子およびフラグを記録するためにトリガーされました。 |
| **MQ_GETSETATTR** | **mq_getattr(3)および mq_setattr(3)** メッセージキュー属性を記録するためにトリガーされます。 |
| **MQ_NOTIFY** | **mq_notify(3)** システムコールの引数を記録するためにトリガーされました。 |
| **MQ_OPEN** | **mq_open(3)** システムコールの引数を記録するためにトリガーされました。 |
| **MQ_SENDRECV** | **mq_send(3)および mq_receive(3)** システムコールの引数を記録するためにトリガーされました。 |
| **NETFILTER_CFG** | Netfilter チェーンの変更が検出されるとトリガーされます。 |
| **NETFILTER_PKT** | Netfilter チェーンを通過するパケットを記録するためにトリガーされました。 |

| イベントタイプ | 説明 |
| --- | --- |
| **OBJ_PID** | シグナルを送信するプロセスに関する情報を記録するためにトリガーされます。 |
| **PATH** | ファイル名のパス情報を記録するためにトリガーされました。 |
| **RESP_ACCT_LOCK**[c] | ユーザーアカウントがロックされるとトリガーされます。 |
| **RESP_ACCT_LOCK_TIMED**[c] | 指定された期間ユーザーアカウントがロックされるとトリガーされます。 |
| **RESP_ACCT_REMOTE**[c] | リモートセッションからユーザーアカウントがロックされるとトリガーされます。 |
| **RESP_ACCT_UNLOCK_TIMED**[c] | 設定した期間後にユーザーアカウントのロックが解除されるとトリガーされます。 |
| **RESP_ALERT**[c] | アラートメールが送信されるとトリガーされます。 |
| **RESP_ANOMALY**[c] | 突然動作がされなかった場合にトリガーされます。 |
| **RESP_EXEC**[c] | 侵入検出プログラムが、プログラムの実行から生じる脅威に応答するとトリガーされます。 |
| **RESP_HALT**[c] | システムのシャットダウン時にトリガーされます。 |
| **RESP_KILL_PROC**[c] | プロセスが終了したときにトリガーされます。 |
| **RESP_SEBOOL**[c] | SELinux のブール値が設定されているとトリガーされます。 |
| **RESP_SINGLE**[c] | システムがシングルユーザーモードに置かれるとトリガーされます。 |
| **RESP_TERM_ACCESS**[c] | セッションの終了時にトリガーされます。 |
| **RESP_TERM_LOCK**[c] | 端末がロックされるとトリガーされます。 |
| **ROLE_ASSIGN** | 管理者が SELinux ロールにユーザーを割り当てる際にトリガーされます。 |
| **ROLE_MODIFY** | 管理者が SELinux ロールを変更する際にトリガーされます。 |
| **ROLE_REMOVE** | 管理者が SELinux ロールからユーザーを削除したときにトリガーされます。 |
| **SELINUX_ERR** | 内部 SELinux エラーが検出されるとトリガーされます。 |

| イベントタイプ | 説明 |
| --- | --- |
| **SERVICE_START** | サービスの起動時にトリガーされます。 |
| **SERVICE_STOP** | サービスが停止したときにトリガーされます。 |
| **SOCKADDR** | ソケットアドレスを記録するためにトリガーされました。 |
| **SOCKETCALL** | **sys_socketcall** システムコールの引数を記録するためにトリガーされました（複数のソケット関連のシステムコールに使用）。 |
| **SYSCALL** | カーネルへのシステムコールを記録するためにトリガーされます。 |
| **SYSTEM_BOOT** | システムの起動時にトリガーされます。 |
| **SYSTEM_RUNLEVEL** | システムのランレベルが変更されたときにトリガーされます。 |
| **SYSTEM_SHUTDOWN** | システムのシャットダウン時にトリガーされます。 |
| **TEST** | テストメッセージの成功値を記録するためにトリガーされます。 |
| **TRUSTED_APP** | このタイプの記録は、監査が必要なサードパーティーアプリケーションで使用できます。 |
| **TTY** | TTY 入力が管理プロセスに送信されたときにトリガーされました。 |
| **USER_ACCT** | ユーザー空間ユーザーアカウントが変更されるとトリガーされます。 |
| **USER_AUTH** | ユーザー空間認証の検出時にトリガーされます。 |
| **USER_AVC** | ユーザー空間 AVC メッセージの生成時にトリガーされます。 |
| **USER_CHAUTHTOK** | ユーザーアカウント属性が変更されるとトリガーされます。 |
| **USER_CMD** | ユーザー空間シェルコマンドの実行時にトリガーされます。 |
| **USER_END** | ユーザースペースセッションの終了時にトリガーされます。 |
| **USER_ERR** | ユーザーアカウントの状態エラーが検出されるとトリガーされます。 |
| **USER_LABELED_EXPORT** | オブジェクトが SELinux ラベルでエクスポートされるとトリガーされます。 |
| **USER_LOGIN** | ユーザーのログイン時にトリガーされます。 |
| **USER_LOGOUT** | ユーザーのログアウト時にトリガーされます。 |

| イベントタイプ | 説明 |
| --- | --- |
| **USER_MAC_POLICY_LOAD** | ユーザースペースデーモンが SELinux ポリシーを読み込む際にトリガーされます。 |
| **USER_MGMT** | ユーザー空間管理データを記録するためにトリガーされます。 |
| **USER_ROLE_CHANGE** | ユーザーの SELinux ロールが変更されたときにトリガーされます。 |
| **USER_SELINUX_ERR** | ユーザー空間の SELinux エラーが検出されるとトリガーされます。 |
| **USER_START** | ユーザー空間セッションの開始時にトリガーされます。 |
| **USER_TTY** | TTY 入力に関する説明メッセージがユーザースペースから送信されるとトリガーされます。 |
| **USER_UNLABELED_EXPORT** | SELinux ラベルなしでオブジェクトがエクスポートされるとトリガーされます。 |
| **USYS_CONFIG** | ユーザー空間のシステム設定の変更が検出されるとトリガーされます。 |
| **VIRT_CONTROL** | 仮想マシンの起動、一時停止、または停止時にトリガーされます。 |
| **VIRT_MACHINE_ID** | ラベルのバインディングを仮想マシンに記録するためにトリガーされます。 |
| **VIRT_RESOURCE** | 仮想マシンのリソース割り当てを記録するためにトリガーされます。 |

[a] で始まる Audit イベントタイプ **ANOM** はすべて、侵入検出プログラムが処理することを目的としています。

[b] このイベントタイプは、TPM(Trusted Platform Module)チップで最適に機能する Integrity Measurement Architecture(IMA)に関連します。

[c] で始まる Audit イベントタイプ **RESP** はすべて、システム上の悪意のあるアクティビティーを検出した場合に侵入検出システムの応答を対象としています。

# 付録C 改訂履歴

**改訂 1-18** Tue 14 Mar 2017 Mirek Jahoda
6.9 GA 公開用バージョン

**改訂 1-14** Wed 4 May 2016 Robert Krátký
Red Hat Enterprise Linux 6.8 のセキュリティーガイドのリリース

**改訂 1-12.4** Tue Dec 16 2014 Robert Krátký
Red Hat カスタマーポータルでソート順序に合わせて更新します。

**改訂 1-12.3** Mon Dec 01 2014 Robert Krátký
POODLE vuln を反映した更新

**改訂 1-12.0** Mon Oct 13 2014 Miroslav Svoboda
Red Hat Enterprise Linux 6.6 のセキュリティーガイドのリリース

**改訂 1-9.9** Feb 21 2013 Martin Prpič
Red Hat Enterprise Linux 6.4 のセキュリティーガイドのリリース

**改訂 1-8.25** Jun 20 2012 Martin Prpič
Red Hat Enterprise Linux 6.3 のセキュリティーガイドのリリース

**改訂 1-7** Dec 6 2011 Martin Prpič
Red Hat Enterprise Linux 6.2 のセキュリティーガイドのリリース