Fedora 19

セキュリティガイド

Fedora Linux をセキュアにするためのガイド

Fuller Johnray [FAMILY Given]

Ha John [FAMILY Given]

O'Brien David [FAMILY Given]

Radvan Scott [FAMILY Given]

Christensen Eric [FAMILY Given]

Ligas Adam [FAMILY Given]

McAllister Murray [FAMILY Given]

Radvan Scott [FAMILY Given]

Walsh Daniel [FAMILY Given]

Grift Dominick [FAMILY Given]

Paris Eric [FAMILY Given]

Morris James [FAMILY Given]

Fedora 19 セキュリティガイド
Fedora Linux をセキュアにするためのガイド
エディッション 19.1

著者	Fuller Johnray [FAMILY Given]	jrfuller@redhat.com
著者	Ha John [FAMILY Given]	jha@redhat.com
著者	O'Brien David [FAMILY Given]	daobrien@redhat.com
著者	Radvan Scott [FAMILY Given]	sradvan@redhat.com
著者	Christensen Eric [FAMILY Given]	sparks@fedoraproject.org
著者	Ligas Adam [FAMILY Given]	agent86@fedoraproject.org
著者	McAllister Murray [FAMILY Given]	mmcallis@redhat.com
著者	Radvan Scott [FAMILY Given]	sradvan@redhat.com
著者	Walsh Daniel [FAMILY Given]	dwalsh@redhat.com
著者	Grift Dominick [FAMILY Given]	domg472@gmail.com
著者	Paris Eric [FAMILY Given]	eparis@parisplace.org
著者	Morris James [FAMILY Given]	jmorris@redhat.com

Fedora セキュリティガイドは、ローカルまたはリモートからの侵入、侵害および悪意のある活動に対してワークステーションとサーバーをセキュアにするプロセスとプラクティスについて、Fedora のユーザーが学習する支援

をするために設計されています。Fedora Linux に焦点を合わせており、すべての Linux システムに対して有効な概念や技術を詳細に説明することではありません。Fedora セキュリティガイドはデータセンター、仕事場および自宅用に安全なコンピューティング環境を構築することに関連する計画とツールを詳細に説明します。適切な知識、警戒およびツールを用いて、Linux を実行しているシステムが完全に機能して、かつ多くの一般的な侵入や侵害方法から安全にすることができます。

序文

1. 表記方法

本ガイドは特定の単語や語句を強調したり、記載内容の特定部分に注意を引かせる目的で次のような表記方法を使用しています。

PDF版 および印刷版では、*Liberation Fonts*[1] セットから採用した書体を使用しています。ご使用のシステムに Liberation Fonts セットがインストールされている場合、HTML 版でもこのセットが使用されます。インストールされていない場合は代替として同等の書体が表示されます。注記: Red Hat Enterprise Linux 5 およびそれ以降のバージョンにはデフォルトで Liberation Fonts セットが収納されます。

1.1. 印刷における表記方法

特定の単語や語句に注意を引く目的で 4 種類の表記方法を使用しています。その表記方法および適用される状況は以下の通りです。

等幅の太字

シェルコマンド、ファイル名、パスなどシステムへの入力を強調するために使用しています。またキー配列やキーの組み合わせを強調するのにも使用しています。例えば、

> 現在作業中のディレクトリ内のファイル `my_next_bestselling_novel` の内容を表示させるには、シェルプロンプトで `cat my_next_bestselling_novel` コマンドを入力してから `Enter` を押してそのコマンドを実行します。

上記にはファイル名、シェルコマンド、キーが含まれています。すべて等幅の太字で表されているため文中内で見分けやすくなっています。

キーが 1 つの場合と複数のキーの組み合わせになる場合を区別するため、その組み合わせを構成するキー同士をハイフンでつないでいます。例えば、

> `Enter` を押してコマンドを実行します。
>
> 1 番目の仮想ターミナルに切り替えるは、`Ctrl+Alt+F2` を押します。X-Windows セッションに戻るには、`Ctrl+Alt+F1` を押します。

最初の段落では押すべき 1 つのキーを特定して強調しています。次の段落では同時に押すべき 3 つのキーの組み合わせが 2 種類ありそれぞれ強調されています。

ソースコードの説明では 1 段落内で提示されるクラス名、メソッド、関数、変数名、戻り値を上記のように 等幅の太字 で表示します。例えば、

> ファイル関連のクラス群はファイルシステムに対しては `filesystem`、ファイルには `file`、ディレクトリには `dir` をそれぞれ含みます。各クラスは個別に関連する権限セットを持っています。

プロポーショナルの太字

アプリケーション名、ダイアログボックスのテキスト、ラベル付きボタン、チェックボックスとラジオボタンのラベル、メニュータイトルとサブメニュータイトルなどシステム上で見られる単語や語句を表します。例えば、

[1] https://fedorahosted.org/liberation-fonts/

メインメニューバーから システム > 個人設定 > マウス の順で選択し マウスの個人設定 を起動します。ボタン タブ内で 左ききのマウス チェックボックスをクリックしてから 閉じる をクリックしマウスの主要ボタンを左から右に切り替えます (マウスを左ききの人が使用するのに適した設定にする)。

gedit ファイルに特殊な文字を挿入する場合は、メインメニューバーから アプリケーション > アクセサリ > 文字マップ の順で選択します。次に 文字マップ メニューバーから 検索 > 検索… と選択して 検索 フィールド内にその文字名を入力し 次 をクリックします。探している文字が 文字表 内で強調表示されます。この強調表示された文字をダブルクリックすると コピーするテキスト フィールド内に置かれるので次に コピー ボタンをクリックします。ここでドキュメントに戻り gedit メニューバーから 編集 > 貼り付け を選択します。

上記には、アプリケーション名、システム全体のメニュー名と項目、アプリケーション固有のメニュー名、GUI インタフェースで見られるボタンやテキストがあります。すべてプロポーショナルの太字で表示されているため文中内で見分けやすくなっています。

########## または #################

等幅の太字やプロポーショナルの太字はいずれであっても斜体の場合は置換可能なテキストか変化するテキストを示します。斜体は記載されている通りには入力しないテキスト、あるいは状況に応じて変化する出力テキストを表します。例えば、

ssh を使用してリモートマシンに接続するには、シェルプロンプトで ssh ***username*@*domain.name*** と入力します。リモートマシンが `example.com` であり、そのマシンで使用しているユーザー名が john なら `ssh john@example.com` と入力します。

`mount -o remount` ***file-system*** コマンドは指定したファイルシステムを再マウントします。例えば、`/home` ファイルシステムを再マウントするコマンドは `mount -o remount /home` になります。

現在インストールされているパッケージのバージョンを表示するには、`rpm -q` ***package*** コマンドを使用します。結果として次を返してきます、***package-version-release***。

上記の太字斜体の単語 — username、domain.name、file-system、package、version、release に注目してください。いずれもコマンドを発行するときに入力するテキスト用のプレースホルダーかシステムにより出力されるテキスト用のプレースホルダーになっています。

タイトル表示のような標準的な使用の他、斜体は新しい重要な用語が初めて出現する場合にも使用されます。例えば、

Publican は *DocBook* の発行システムです。

1.2. 引用における表記方法

端末の出力とソースコード一覧は、視覚的に周囲の文から区別されています。

端末に送信される出力は `mono-spaced roman` (等幅の Roman) にセットされるので以下のように表示されます。

```
books        Desktop   documentation  drafts  mss    photos   stuff  svn
books_tests  Desktop1  downloads      images  notes  scripts  svgs
```

ソースコードの一覧も `mono-spaced roman` (等幅の Roman) でセットされますが、以下のように強調表示されます。

```
package org.jboss.book.jca.ex1;
```

```
import javax.naming.InitialContext;

public class ExClient
{
   public static void main(String args[])
       throws Exception
   {
      InitialContext iniCtx = new InitialContext();
      Object         ref    = iniCtx.lookup("EchoBean");
      EchoHome       home   = (EchoHome) ref;
      Echo           echo   = home.create();

      System.out.println("Created Echo");

      System.out.println("Echo.echo('Hello') = " + echo.echo("Hello"));
   }
}
```

1.3. 注記および警告

情報が見過ごされないよう 3 種類の視覚的なスタイルを使用して注意を引いています。

注記

注記は説明している部分に対するヒントや近道あるいは代替となる手段などになります。注記を無視しても悪影響はありませんが知っておくと便利なコツを見逃すことになるかもしれません。

重要

重要ボックスは見逃しやすい事項を詳細に説明しています。現在のセッションにのみ適用される設定上の変更点、更新を適用する前に再起動が必要なサービスなどがあります。重要ボックスを無視してもデータを喪失するような結果にはなりませんがイライラ感やフラストレーションが生じる可能性があります。

警告

警告は無視しないでください。警告を無視するとデータを喪失する可能性が非常に高くなります。

2. フィードバック

本ガイドに誤植を見つけられた場合や本ガイドの改善案をお持ちの場合はぜひお知らせください。Bugzilla *http://bugzilla.redhat.com/bugzilla/* にて、Product には Fedora. を選びレポートの提出をお願いいたします。

バグレポートを提出される場合は、そのガイドの識別子となる *security-guide* を必ず明記して頂くようお願いします。

ドキュメントに関する改善のご意見についてはできるだけ具体的にお願いいたします。エラーを発見された場合は、セクション番号および該当部分の前後の文章も含めてご報告頂くと照合が容易になります。

セキュリティの概要

ビジネスの経営および個人情報の記録のために、パワフルかつネットワーク化されたコンピューターに依存してきているため、すべての産業はネットワークとコンピューターのセキュリティの実践を中心として組成されてきています。企業は、運用している組織の要求事項を適合させるために、適切にシステムを監査して、ソリューションを仕立てるために、セキュリティ専門家の知識とスキルを求めるようになってきています。多くの組織は実際にますます変化が激しくなるので、労働者がローカルまたはリモートで会社の IT リソースへアクセスするとともに、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明確になってきています。

不幸にも、多くの組織（および個人ユーザー）はセキュリティを、結果論や増大するパワーにより見落とすプロセス、生産性および予算的な懸念としてみなしています。適切なセキュリティの導入は、しばしば事後に賛成されます — 認可されない侵入者がすでに占拠した ##。セキュリティ専門家は、インターネットのような信頼されないネットワークにサイトを接続する前に適切な対策をとることは、侵入者に多くの試みを挫折させる効果的な方法であるということに賛成します。

1.1. セキュリティのイントロダクション

1.1.1. コンピューター・セキュリティとは？

コンピューター・セキュリティは、コンピューティングと情報処理の幅広い領域を取り扱う一般的な用語です。日々のビジネス取引を行い、極めて重要な情報にアクセスするために、コンピューターシステムとネットワークに依存する産業は、それらのデータを全体の資産の最も重要な部分であると見なしています。いくつかの用語と評価指標が、Total Cost of Ownership (TCO) や Quality of Service (QoS) のように、日常のビジネス会話に入ってきています。これらの評価指標を用いることで、計画とプロセス管理のコストの一部として、データの完全性や高可用性のような観点を産業が計算できるようになります。電子商取引のようないくつかの産業において、データの可用性と信頼性は成功と失敗の分かれ目になりえます。

1.1.1.1. コンピューター・セキュリティはどのように起こるのでしょうか？

情報セキュリティは、個人情報、金融情報、および他の制限された情報が暴露されないようにするため、パブリック・ネットワークへの増大する依存のため何年もかけて進歩してきました。すべての業種にわたる組織が取り扱う情報だけでなくその転送や暴露について再検討するよう促す、Mitnick [1] や Vladimir Levin [2] の事件のような数多くの事例があります。インターネットの普及は、データ・セキュリティにおける大きな努力を促す最も重要な開発の1つでした。

インターネットが提供するリソースへアクセスするために、いまだ増え続ける人々が PC を使用しています。研究や情報探索から電子メールや電子商取引まで、インターネットは20世紀の最も重要な開発の1つとみなされるようになってきました。

しかしながら、インターネットおよびそれ以前のプロトコルは、########システムとして開発されました。つまり、インターネットプロトコル (Internet Protocol) 自身はセキュアには設計されていません。TCP/IP 通信階層に組み込まれている公式のセキュリティ標準はありません。それは、ネットワーク越しに潜在的に悪意のあるユーザーやプロセスに開かれたままです。最近の開発はインターネット通信をよりセキュアにしてきましたが、国中の注目を集め、私たちに完全に安全なものは何もないという事実を警告する、いくつかのインシデントがいまだにあります。

1.1.1.2. 今日のセキュリティ

2000年2月、分散サービス妨害 (DDoS: Distributed Denial of Service) 攻撃が、インターネットにある最も高トラフィックのサイトのいくつかに対して行われました。攻撃者は *ping flood* とも呼ばれる大きな ICMP パ

[1] http://law.jrank.org/pages/3791/Kevin-Mitnick-Case-1999.html
[2] http://www.livinginternet.com/i/ia_hackers_levin.htm

ケットを送信することにより数時間ルータを使用不能し、yahoo.com, cnn.com, amazon.com, fbi.gov, および他のいくつかのサイトを通常のユーザーから完全にアクセス不能にしました。攻撃は、脆弱性のあるネットワーク・サーバーをスキャンする、特別に作成された広く利用可能なプログラムを使用している未知の攻撃者によりもたらされ、サーバーに######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、犠牲サイトをあふれさせ利用不可能にするあらゆる感染したサーバーで、攻撃の時間を計りました。多くの人は、パケットが送られたどんなところでも、どんな目的に対しても、すべての入力データを受け付けるために構成されるよう、ルーターとプロトコルが使われる方法で基本的な欠陥にある攻撃を非難しました。

2007年、Wired Equivalent Privacy (WEP) 無線暗号化プロトコルの広く知られる脆弱性をエクスプロイトするデータ侵害により、世界中の金融機関から4500万を越えるクレジットカード番号が盗まれました。[3]

別のインシデントにおいて、バックアップ・テープに保存された、220万人の患者の請求記録が配送者のフロントシートから盗まれました。[4]

現在、世界中で推定18億人がインターネットを使用しています、または使用していました。[5] 同時に:

- ある特定の日に、CERT Coordination Center at Carnegie Mellon University[6] へと報告されたセキュリティ違反のメジャー・インシデントは推定225あります。
- 2003年、CERT に報告されたインシデントの数は、2001年の52,658、2002年の82,094から跳ね上がりました。[7]
- ここ3年の最も危険なインターネット・ウイルスに関するワールドワイドの経済的影響は、132億アメリカドルと見積もられました。[8]

2008年のグローバルなビジネスと技術のエグゼクティブ調査 "The Global State of Information Security"[9] から、*CIO Magazine* により断言された、いくつかのポイントは以下です:

- 43%のみがユーザー・コンプライアンスの監査または監視とセキュリティ・ポリシーが一致します
- 22%のみがデータを使用する外部企業の一覧を維持しています
- 約半数のセキュリティ・インシデントはソースが "Unknown" と印がつけられました
- 回答者の44%が翌年にセキュリティ予算を増やす計画をします
- 59%は情報セキュリティ戦略を持ちます

これらの結果は、コンピューター・セキュリティが IT 予算に対する支出を定量化して正当化するようになってきたことの現実性を強調します。データの完全性と高可用性を必要とする組織は、システム、サービスおよび情報の 24x7 の信頼性を確実にするために、システム管理者、開発者、および技術者のスキルを引き出します。犠牲者が悪意のあるユーザー、プロセスおよび協調された攻撃に落とされると、組織の成功に対する直接の脅威になります。

不幸にも、システムとネットワークのセキュリティは、組織が情報をどのようにみなし、使用し、処理し、転送するかの複雑な知識を必要とする、難しい命題になるでしょう。組織（および組織を構成する人々）がビジネスを実施する方法を理解することは、適切なセキュリティ計画を導入することに優先します。

[3] http://www.theregister.co.uk/2007/05/04/txj_nonfeasance/
[4] http://www.healthcareitnews.com/story.cms?id=9408
[5] http://www.internetworldstats.com/stats.htm
[6] http://www.cert.org
[7] http://www.cert.org/stats/fullstats.html
[8] http://www.newsfactor.com/perl/story/16407.html
[9] http://www.csoonline.com/article/454939/The_Global_State_of_Information_Security_

1.1.1.3. セキュリティの標準化

すべての産業における企業は、アメリカ医師会 (AMA: American Medical Association) や IEEE (Institute of Electrical and Electronics Engineers) のような標準化推進団体により作られた規制やルールに依存します。同じ理念が情報セキュリティにも有効です。多くのセキュリティ・コンサルタント・ベンダーは CIA (機密性、完全性および可用性) として知られる標準的なセキュリティ・モデルについて意見が一致します。この3階層モデルは、機密情報のリスクアセスメントやセキュリティ方針の確立のために、一般的に受け入れられたコンポーネントです。以下は、CIA モデルをさらに詳細に説明します:

- 機密性 — 機密情報は事前に定義された個人の組に対してのみ利用可能でなければいけません。情報の認可されない送信や使用は制限されなければいけません。たとえば、情報の機密性は、顧客情報や金融情報が個人情報の盗難や信用詐欺のような悪意のある目的のために認可されない個人により得られないよう、確実にします。

- 完全性 — 情報が不完全または不正確に与えられる情報で変更されないようすべきです。認可されないユーザーが機密情報を変更または破壊する能力から制限されなければいけません。

- 可用性 — 情報は、認可されたユーザーが必要なときいつでもアクセス可能でなければいけません。可用性は情報が合意された頻度とタイムリーさで得られることの保証です。これはしばしば、パーセンテージの観点で測定され、Service Level Agreements (SLA) において公式に合意されます。

1.1.2. SELinux

Fedora は SELinux という Linux カーネルを強化するものを含みます。これは、システムにおいてファイル、プロセス、ユーザーおよびアプリケーションをより精細なレベルで制御する強制アクセス制御 (MAC: Mandatory Access Control) アーキテクチャーを実装しています。SELinux の詳細な説明が *#SELinux ###* にあります。

1.1.3. セキュリティ・コントロール

コンピューター・セキュリティはしばしば、一般的に######として参照される、3つの異なるマスター・カテゴリに分割されます:

- 物理的

- 技術的

- 管理的

これら3つの幅広いカテゴリは、適切なセキュリティ導入の主な目的を定義します。これらのコントロールの中で、コントロールとそれらを実装する方法をさらに詳細化するサブカテゴリです。

1.1.3.1. 物理的コントロール

物理的コントロールは、機密なマテリアルへの認可されないアクセスを阻止または防止するために使用される、定義された構造におけるセキュリティ対策の実装です。物理的コントロールの例は以下です:

- 有線監視カメラ

- 動作・温度警告システム

- 警備員

- 写真付き身分証明書

- ロックされた錠前をかけられたスチールドア

- バイオメトリクス（指紋、声、顔、虹彩、筆跡、および個人を認識するために使われる他の自動化された方法を含みます）

1.1.3.2. 技術的コントロール

技術的コントロールは、物理構造やネットワークにおいて機密データのアクセスと制御を制御するために基礎となる技術を使用します。技術的コントロールは広範囲で以下のような技術を含みます:

- 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: Access control lists)
- ファイル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の人間的要素を定義します。

- トレーニングおよび意識向上
- 災害準備および復旧計画
- 要員採用および退職戦略
- 要員登録およびアカウンティング

1.1.4. 結論

今、セキュリティの起源、理由および観点について学んできたので、より簡単に Fedora に関する適切な行動指針を決定するようになることがわかります。どの要素と条件が適切な戦略を計画・導入するためにセキュリティを作り上げるかを知ることは重要です。セキュリティ・プロセスの細部のより深いところを調べるとき、この情報を心に留めておくと、プロセスを正式化して、パスがより明確になります。

1.2. 攻撃者と脆弱性

素晴らしいセキュリティ戦略を計画・導入するために、決意して動機付けられた攻撃者がシステムを危険にさらすためにエクスプロイトするいくつかの問題をまず理解します。しかし、これらの問題を詳細化する前に、攻撃者を識別するときに使われる用語を定義しなければいけません。

1.2.1. ハッカーの簡単な歴史

####という語の近代的な意味は、1960年代とマサチューセッツ工科大学 (MIT) の Tech Model Railroad Club (大規模で複雑な詳細の鉄道セットを設計しました) にさかのぼる起源を持ちます。ハッカーは、賢いトリックや問題の回避方法を発見したクラブのメンバーに対して使われた名前です。

ハッカーという語は、コンピューター通から才能あるプログラマまですべてを説明するためにきました。多くのハッカーの間の共通の特徴は、ほとんど外部的な動機づけではなく、コンピューター・システムとネットワークがどのように機能するかを詳細に調査したいという意欲です。オープンソース・ソフトウェアの開発者はしばしば自分自身と同僚をハッカーであると考え、尊敬を表す語としてその語を使用します。

一般的に、ハッカーは######の形式に従います。それは、情報の探求と習熟が不可欠であることを表し、この知識を共有することはコミュニティへのハッカーの義務であることを表します。この知識の探求の間、何人かのハッカーはコンピュータ・システムにおけるセキュリティ・コントロールを回避しようとするアカデミックな挑戦を楽しみます。この理由により、プレスはしばしばハッカーという言葉を悪質な、悪意のある、犯罪の意図を持ってシステムとネットワークに不法にアクセスする人々を説明するために使用します。この種類のコンピューター・

ハッカーに対するより正確な言葉は#####です ― 2つのコミュニティを区別するために1980年代中ごろにハッカーにより作成された言葉。

1.2.1.1. Shades of Gray

システムとネットワークにある脆弱性を見つけてエクスプロイトする人々のコミュニティの中には、いくつかの別々のグループがあります。これらのグループはしばしば、セキュリティ調査を実行するときにその人たちが「身につけている」帽子の色相により説明され、これらの色相はその人たちの意図を示します。

###############は、ネットワークとシステムのパフォーマンスを検査するため、およびそれらが侵入のためにどのように脆弱であるかを決めるため、それらをテストする人々です。通常、ホワイト・ハット・ハッカーは、自身のシステム、およびシステム監査の目的のために特別に雇われたクライアントのシステム、をクラックします。アカデミックな研究者とプロフェッショナルのセキュリティ・コンサルタントはホワイト・ハット・ハッカーの2つの例です。

###############はクラッカーの同義語です。一般に、クラッカーはプログラミングとシステムの侵入へのアカデミックな側面にあまりフォーカスしません。利用可能なクラック・プログラムに依存します。また、個人的利益のために機密情報を暴露するため、またはターゲット・システムやネットワークにダメージを与えるために、システムにあるよく知られた脆弱性をエクスプロイトします。

他方、##############は、多くの状況においてホワイト・ハット・ハッカーのスキルと意図を持ちますが、場合によっては崇高な目的以外にも知識を使用します。グレイ・ハット・ハッカーは自身の予定を達成するために時々ブラック・ハットをかぶるホワイト・ハット・ハッカーのように考えられます。

グレイ・ハット・ハッカーは一般的にハッカー倫理の他の形式に同意します。それは、システムに侵入可能であると同時に、ハッカーが盗難を行わない、または機密性を破らないことを言います。しかし、ある人はシステムに侵入する行為自体が非倫理的あると主張します。

侵入者の意図に関わらず、クラッカーがエクスプロイトを試みたいかもしれないという弱さを知ることは重要です。本章の残りはこれらの問題に焦点をあてます。

1.2.2. ネットワーク・セキュリティへの脅威

ネットワークを以下の観点で設定するとき、バッド・プラクティスは攻撃のリスクを増やす可能性があります。

1.2.2.1. セキュアではないアーキテクチャー

設定を誤っているネットワークは、認可されないユーザーの最初の入り口になります。信頼に基づいた、オープンなローカルネットワークを、非常にセキュアではないインターネットに対して脆弱なままにしておくことは、犯罪が多発する地区で半ドアにしておくようなものです。 ― ある期間は何も起きないかもしれませんが、##誰かが機会を活用するでしょう。

1.2.2.1.1. ブロードキャスト・ネットワーク

システム管理者はしばしば、セキュリティ・スキームにおけるネットワーク・ハードウェアの重要性に気がつきません。ハブやルーターのような単純なハードウェアは、ブロードキャストやスイッチではない原則に基づいています。すなわち、あるノードが受信ノードへネットワークを超えてデータを転送するときはいつでも、ハブやルーターは、受信ノードが受信してデータを処理するまで、データ・パケットのブロードキャストを送り続けます。この方式は、外部の侵入者やローカル・ホストの認可されないユーザーによる、address resolution protocol (*ARP*) や media access control (*MAC*) アドレスの偽装に対して最も脆弱です。

1.2.2.1.2. 集中化したサーバー

他の潜在的なネットワークの落とし穴は、集中化されたコンピューター環境の使用です。多くのビジネスに対する一般的なコスト削減の対策は、1台の強力なマシンにすべてのサービスを集約することです。管理がより簡単

になり、複数サーバーの設定よりもコストを非常に安くできるので、これは便利でしょう。しかし、集中化し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。集中化したサーバーがセキュリティ侵害されると、データ操作や窃盗を引き起こしやすいよう、ネットワークを完全に使い物にならなくしたりより悪くしたりできます。これらの状況において、集中化したサーバーはネットワーク全体へアクセスできるオープン・ドアになります。

1.2.3. サーバー・セキュリティへの脅威

サーバーはしばしば組織の重要な情報を非常に多く取り扱っているので、サーバー・セキュリティはネットワーク・セキュリティと同じように重要です。サーバーがセキュリティ侵害されると、すべてのコンテンツがクラッカーの思いのままに窃盗または操作できるようになるかもしれません。以下のセクションは、おもな問題のいくつかを詳細に説明します。

1.2.3.1. 未使用のサービスとオープン・ポート

Fedora の完全インストールには、1000以上のアプリケーションとライブラリのパッケージが含まれます。しかしながら、多くのサーバ管理者は、ディストリビューションにおいてすべての単独のパッケージをインストールしたいとは思いません。代わりに、いくつかのサーバ・アプリケーションを含めて、パッケージの基本インストールをしたいと思います。

システム管理者の間で共通の出来事は、実際にどのアプリケーションがインストールされるかに注意を払わずにオペレーティング・システムをインストールすることです。不必要なパッケージが、インストールされ、デフォルトの設定で設定され、おそらく有効にされている可能性があるので、これは問題があります。管理者が意識することなくサーバーまたはワークステーションで実行するために、Telnet、DHCP や DNS のような期待しないサービスの原因となる可能性があります。これらは、サーバーへと期待しないトラフィックを順番に引き起こす可能性があります。もしくは、クラッカーがシステムの中へ入る潜在的な道になる可能性があります。ポートを閉じて、未使用のサービスを無効にすることに関する詳細は ############ を参照してください。

1.2.3.2. パッチ未適用のサービス

デフォルトのインストールに含まれる多くのサーバー・アプリケーションは、しっかりとしていて、全体を通してテストされたソフトウェアの集まりです。何年も本番環境において使用していると、それらのコードは全体を通して精錬され、多くのバグが発見され修正されていきます。

しかしながら、完璧なソフトウェアのようなものはありません。また、さらなる精錬の余地が常にあります。さらに、比較的新しいソフトウェアはしばしば、その最近の本番環境への出現のため、または、他のサーバー・ソフトウェアほど普及していないため、期待されているほど厳しくテストされていません。

開発者とシステム管理者はしばしば、サーバー・アプリケーションにおいてエクスプロイト可能なバグを見つけます。そして、Bugtraq メーリングリスト (*http://www.securityfocus.com*) や Computer Emergency Response Team (CERT) ウェブサイト (*http://www.cert.org*) のような、バグトラックやセキュリティ関連のウェブサイトにおいて情報を公開します。これらのメカニズムはセキュリティ脆弱性をコミュニティに警告する効果的な方法であるにも関わらず、システムに適切にパッチを当てるかはシステム管理者しだいです。クラッカーがこれらの同じ脆弱性トラッキング・サービスにアクセスして、できるときにいつでもパッチ未適用のシステムをクラックするために情報を使うので、これは特に当てはまります。素晴らしいシステム管理者は、コンピューティング環境を確実によりセキュアにするために、警戒、定期的なバグ・トラッキング、および適切なシステム・メンテナンスを必要とされます。

システムを最新に保つことに関する詳細は ################ を参照してください。

1.2.3.3. 不注意な管理

システムにパッチを当てることに失敗した管理者は、サーバー・セキュリティへの最も重大な脅威の1つです。*SysAdmin, Audit, Network, Security Institute* (*SANS*) によると、コンピューター・セキュリティ脆弱性のおもな原因は、「トレーニングされていない人にセキュリティを維持することを割り当て、その仕事をできるようにす

るためのトレーニングも時間も与えないこと」です。[10] これは自信過剰または動機付けられた管理者と同じくらい、経験の少ない管理者に当てはまります。

他の人々がシステム・カーネルのログ・メッセージやネットワーク・トラフィックを見落とす一方で、何人かの管理者はサーバーとワークステーションにパッチを当てることに失敗します。他の一般的なエラーは、サービスのデフォルトパスワードまたはキーが変更されずに残っているときです。たとえば、いくつかのデータベースは、データベース開発者がシステム管理者がインストール後すぐにこれらのパスワードを変更すると考えて、デフォルトの管理パスワードを持ちます。データベース管理者がこのパスワードを変更し忘れると、経験の少ないクラッカーでさえ、データベースの管理者権限を得るために、広く知られたデフォルトのパスワードを使用できます。不注意な管理がどのようにシステムの侵害につながる可能性があるかに関する例がいくつかあります。

1.2.3.4. 本質的にセキュアではないサービス

最も注意深い組織でさえ、選択したネットワーク・サービスが本質的にセキュアでなければ、脆弱性の犠牲になる可能性があります。たとえば、信頼されたネットワーク上で使用されるという仮定の下に開発されたサービスがたくさんあります。しかしながら、サービスがインターネット（それ自体は本質的に信頼できません）で利用可能になるとすぐに、この仮定は崩壊します。

セキュアではないネットワーク・サービスのカテゴリの1つは、認証に対して暗号化されないユーザー名とパスワードを必要とするものです。Telnet と FTP はそのようなサービスの2つです。パケット盗聴ソフトウェアがリモートユーザーとそのようなサービスの間でトラフィックを監視しているならば、ユーザー名とパスワードが簡単に横取りされる可能性があります。

本質的に、そのようなサービスもより簡単に、セキュリティ業界は###攻撃と呼びますものの犠牲になります。この種類の攻撃において、意図したサーバーの代わりに彼のマシンに向けるために、ネットワークにおいてクラックされたネームサーバーをだますことにより、クラッカーはネットワーク・トラフィックをリダイレクトします。いったん誰かがサーバーへのリモート・セッションをオープンすると、攻撃者のマシンが、リモート・サービスと情報をキャプチャされていることを用心していないユーザーの間に静かに座る、見えないパイプとして動作します。この方法で、クラッカーはサーバーやユーザに気づかれることなく、管理パスワードや生のデータを集められます。

セキュアではないサービスのもう1つのカテゴリは、LAN 利用を期待して開発されたが、不幸にも (リモート・ユーザーに対して) WAN を含める拡張をされた、NFS や NIS のようなネットワーク・ファイル・システムおよびネットワーク情報サービスです。NFS はクラッカーが NFS 共有をマウントして、そこに含まれるすべてのものにアクセスするのを防ぐために設定された、あらゆる認証やセキュリティのメカニズムをデフォルトでは持ちません。NIS も同様に、プレインテキスト ASCII または DBM (ASCII から派生した) データベースの中に、パスワードやファイル・パーミッションを含む、ネットワークにあるすべてのコンピュータに知られなければいけない重要な情報を持ちます。このデータベースへのアクセス権を得たクラッカーは、管理者のアカウントを含め、ネットワークにおけるすべてのユーザーアカウントにアクセスできます。

Fedora はデフォルトでそのようなサービスをすべてオフにしてリリースされています。しかしながら、管理者がしばしば、これらのサービスを使用するよう強制されることがあるので、注意深く設定することが重要な意味を持ちます。安全なようにサービスをセットアップする方法の詳細は ############## を参照してください。

1.2.4. ワークステーションとホーム PC のセキュリティへの脅威

ワークステーションおよびホーム PC は、ネットワークやサーバーのように攻撃される傾向にないかもしれません。しかし、しばしばクレジットカード情報のような機密データを含むので、システム・クラッカーの標的にされます。ワークステーションは、ユーザーが知ることなく選出され、共同攻撃における "スレーブ" マシンとして攻撃者により使用される可能性もあります。これらの理由により、ワークステーションの脆弱性を理解することは、オペレーティング・システムの再インストール、もっと悪ければデータ窃盗からの回復の頭痛からユーザーを守ります。

[10] http://www.sans.org/resources/errors.php

1.2.4.1. 悪いパスワード

悪いパスワードは攻撃者がシステムへのアクセス権を得るために最も簡単な方法の1つです。パスワードを作成するときに一般的な落とし穴を避ける方法の詳細は、############### を参照してください。

1.2.4.2. 脆弱なクライアント・アプリケーション

管理者が完全にセキュアでパッチを当てたサーバーにしているにも関わらず、リモート・ユーザーがアクセスするときにセキュアであるとは限りません。たとえば、サーバーがパブリックネットワーク上で Telnet や FTP サービスを提供していると、攻撃者は平文のユーザー名とパスワードがネットワーク上を流れているので、それらを取ることができます。そして、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ためにアカウント情報を使用します。

SSH のようなセキュアなプロトコルを使用しているときでさえ、リモート・ユーザーは、クライアント・アプリケーションを更新していないと、特定の攻撃に対して脆弱であるかもしれません。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。一度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とることができます。この問題は v.2 SSH プロトコルで修正されました。しかしユーザーは、どのアプリケーションがそのような脆弱性を持ち、更新する必要があるのかを把握し続けないといけません。

################### は、管理者とホームユーザーがコンピューター・ワークステーションの脆弱性を制限するためにどんなステップをとるべきかをより詳細に説明しています。

1.3. 脆弱性のアセスメント

時間、リソースおよびモチベーションを与えられると、クラッカーはほとんどすべてのシステムに侵入できます。結局、現在利用可能なすべてのセキュリティの手順と技術は、あらゆるシステムが侵入から完全に安全であることを保証することはできません。ルーターはインターネットへの安全なゲートウェイの助けになります。ファイアウォールはネットワークの境界の助けになります。VPN は暗号化されたストリームにおいて安全にデータを通過させます。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、これらの技術のそれぞれの成功は、以下を含む多くの変動要因に依存します。

- 技術の設定、監視および維持に責任のあるスタッフの習熟。
- サービスとカーネルに迅速かつ効果的にパッチおよび更新する能力。
- ネットワーク上の一定した警戒を維持する責任のある人の能力

システムと技術がデータのダイナミックな状態にすると、企業のリソースをセキュアにすることは極めて難しくなります。しばしばシステムのすべてに対する専門家のリソースを見つけることは難しいです。情報セキュリティの多くの領域における高いレベルの知識を持つ要員を持つことができる間、少し以上の主題領域に精通しているスタッフを維持することは難しいです。これはおもに、情報セキュリティの各主題領域は一定の注意と集中を必要とするからです。情報セキュリティは有効なままではありません。

1.3.1. 敵のような考え

あなたが企業ネットワークの管理者であると仮定します。そ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、ネットワーク・モニタ、侵入検知システム等から構成されます。今日のソフトウェアとネットワーク環境の複雑さを与えられると、エクスプロイットとバグは必然性があります。ネットワーク全体をパッチとアップデートで最新に保つことは、異質なシステムを持つ大きなネットワークにおいて気が重い作業であることがわかります。

習熟の要件と現状維持の作業を組み合わせます。そして、不利益なインシデントが発生し、システムが侵害され、データが破壊され、サービスが中断されることは不可避です。

セキュリティ技術を強化して、システム、ネットワーク、およびデータを保護する支援とするため、あなたはクラッカーのように考え、弱さに対するチェックをすることによりシステムのセキュリティを測定しなければいけません。

自身のシステムとネットワーク・リソースに対する予防的な脆弱性アセスメントは、クラッカーがエクスプロイトする前に対処できる潜在的な問題を明らかにします。

脆弱性アセスメントはあなたのネットワークおよびシステムのセキュリティの内部監査です。(############ に説明されているように)ネットワークの機密性、完全性および可用性を支持する結果です。一般的に、脆弱性アセスメントは、対象システムに関する重要なデータを集めることを通じて、調査フェーズから始めます。このフェーズはシステム準備フェーズにつながります。それによって、対象が基本的にすべての既知の脆弱性をチェックされます。準備フェーズは報告フェーズに達します。ここで、発見したものは高中低のカテゴリに分類され、対象のセキュリティを向上させる(または脆弱性のリスクを低減させる)方法が議論されます。

あなたの自宅の脆弱性アセスメントを実行しているならば、自宅のドアが閉められて鍵がかけられているかどうかを確認するために、それぞれのドアをチェックするでしょう。確実にすべての窓が完全に閉まっており、正しく鍵がかけられていることもチェックします。この同じような概念をシステム、ネットワークおよび電子データに適用します。悪意のあるユーザーはあなたのデータの泥棒および心ない破壊者です。ツール、精神性および動機に注目します。そうすると、彼ら彼女らの行動に素早く反応できます。

1.3.2. アセスメントとテストの定義

脆弱性アセスメントは2種類に分解できます: ####### および #######。

外から中を見る脆弱性アセスメントを実行するとき、外側からシステムを危険にさらすことを試みます。会社の外側であることは、あなたにクラッカーの観点を与えます。クラッカーが見るものを見ます — 公にルート可能な IP アドレス、*DMZ* にあるシステム、ファイアウォールの外部インタフェース、およびその他。DMZ は "demilitarized zone" を意味します。ここで、企業プライベート LAN のような信頼された内部ネットワーク、およびパブリックなインターネットのような信頼され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一致します。一般的に、DMZ は ウェブ (HTTP) サーバー、FTP サーバー、SMTP (e-mail) サーバーおよび DNS サーバーのような、インターネットのトラフィックにアクセス可能なデバイスを含みます。

中から外を見る脆弱性アセスメントを実行するとき、あなたは内部にいて、状態が信頼されると昇格されるため、いくらかの優位性があります。これは一度システムにログオンしたあなたや同僚の視点です。プリント・サーバー、ファイル・サーバー、データベースおよび他のリソースを見ます。

これら2種類の脆弱性アセスメントの著しい区別があります。会社の内部にいることは、どの外部者よりも上昇された権限を与えられます。多くの組織において今でも、セキュリティは侵入者を締め出すという方法で構成されています。(部門内ファイアウォール、ユーザー・レベル・アクセス制御、内部リソースに対する認証手順などのように)組織の内部をセキュアにしていることは非常にまれです。一般的に、多くのシステムが会社の内部にあるので、中から外を見るときより多くのリソースがあります。一度あなた自身を会社の外部者と設定すると、ただちに信頼されない状態を与えられます。あなたが外部的に利用可能なシステムとリソースは一般的に非常に制限されます。

脆弱性アセスメントと#####の違いを検討します。侵入テストへの第一歩として脆弱性アセスメントを考えます。アセスメントから収集された情報はテストのために使用されます。アセスメントがホールや潜在的な脆弱性に対するチェックをするために行われるのに対して、侵入テストは発見したものを実際にエクスプロイトしようとします。

ネットワーク・インフラストラクチャをアセスメントすることは、ダイナミックなプロセスです。セキュリティ(情報も物理も)はダイナミックです。概要に示されるアセスメントを実行することは、フォールス・ポジティブとフォールス・ネガティブが現れる可能性があります。

セキュリティ管理者は、使用しているツールと保有している知識を同じくらい素晴らしいです。現在、多くの形態のアセスメント・ツールが利用可能です。それらをシステムに対して実行して、そして、大抵いくつかのフォールス・ネガティブがあることを保証します。プログラムの間違いかユーザーの誤りかによらず、結果は同じです。ツールが実際に存在しない脆弱性を見つけるかもしれません(フォールス・ポジティブ)。もしくはさらに悪いことに、ツールが実際に存在する脆弱性を見つけないかもしれません(フォールス・ネガティブ)。

これで脆弱性アセスメントと侵入テストの違いが定義されたので、あなたの新しいベスト・プラクティス・アプローチの一部として侵入テストを行う前に、アセスメントの結論を出して、注意深くレビューします。

警告

本番リソースにおける脆弱性をエクスプロイトする試みは、システムとネットワークの生産性や効率に悪影響を与える可能性があります。

以下の一覧は脆弱性アセスメントを実施するためにいくつかの有益性を検討します。

- 情報セキュリティにプロアクティブなフォーカスを当てる
- クラッカーに見つけられる前に潜在的なエクスプロイトを見つける
- システムを最新でパッチが当てられた状態をもたらす
- 成長とスタッフの習熟に役立つよう促進する
- 経済的損失とネガティブな広報を減らす

1.3.2.1. 方法論の確立

脆弱性アセスメント用のツールを選択する支援のために、脆弱性アセスメントの方法論を確立することは助けになります。不幸にも、現在のところ事前に定義された、もしくは工業的に証明された方法論はありません。しかしながら、一般的な判断およびベスト・プラクティスが十分なガイドとして振る舞います。

これらの質問に対する答えは、どのツールを選択するかだけでなく、そのツールをどのような方法で使用するかを決める助けになるので、重要です。

方法論の確立に関する詳細は、以下のウェブサイトを参照してください:

- *http://www.isecom.org/osstmm/ The Open Source Security Testing Methodology Manual* (OSSTMM)
- *http://www.owasp.org/ The Open Web Application Security Project*

1.3.3. ツールの評価

アセスメントはいくつかの形式の情報収集ツールにより始められます。ネットワーク全体をアセスメントするとき、動作しているホストを見つけるために、まずレイアウトをマップします。一度位置が決められると、それぞれ各ホストを検査します。これらのホストに焦点をあてることは、他のセットのツールを必要とします。使うためのツールを知ることは、脆弱性を見つけるときの最も重要な手順かもしれません。

日常生活のあらゆる場面のように、同じ仕事を実行する数多くの異なるツールがあります。この概念は脆弱性アセスメントを実行することにも同様に当てはまります。オペレーティングシステム、アプリケーション、そしてネットワークにさえ（使用されるプロトコルに基づきます）具体的なツールがあります。いくつかのツールはフリーです。他のものはそうではありません。いくつかのツールは直感的で使いやすいです。一方、他のツールは不可解であり、十分に文書化されませんが、他のツールにはない機能を持ちます。

正しいツールを見つけることは、気が重い仕事であるかもしれません。最後には経験が重要になります。可能ならば、実験ラボをセットアップして、それぞれの強みと弱みに注目して、できる限り多くのツールを試験します。ツールに対する README ファイルまたはマニュアル・ページをレビューします。さらに、ツールに対する記事、ステップ・バイ・ステップのガイド、またはメーリングリストのような、詳細に関してインターネットに目を向けます。

以下で説明されるツールは、単に利用可能なツールの小さなサンプルです。

1.3.3.1. Nmap を用い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利用される Fedora に含まれる一般的なツールです。Nmap は長年にわたり利用可能であり、おそらく情報を集めるときに最もよく使われるツールです。そのオプションと使用法の詳細な説明を提供する、素晴らしいマニュアル・ページが含まれます。管理者は、ホストシステムとそれらのシステムにおいて開いているポートを見つけるためにネットワークにおいて Nmap を使用できます。

Nmap は脆弱性アセスメントにおける十分な第一歩です。ネットワークの中にあるホストすべてを図示します。そして、Nmap が特定のホストで実行しているオペレーティング・システムを特定する試行ができるようにす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の使用と不必要なサービスの停止の方針を確立するための素晴らしい基礎です。

1.3.3.1.1. Nmap の使用

Nmap は nmap コマンドを、スキャンするマシンのホスト名または IP アドレスを後ろにつけて、入力することによりシェル・プロンプトから実行することができます。

```
nmap foo.example.com
```

基本的なスキャン（ホストの位置や他のネットワーク条件に依存して数分かかります）の結果は以下のように見えるでしょう。

```
Starting Nmap 4.68 ( http://nmap.org )
Interesting ports on foo.example.com:
Not shown: 1710 filtered ports
PORT    STATE  SERVICE
22/tcp  open   ssh
53/tcp  open   domain
70/tcp  closed gopher
80/tcp  open   http
113/tcp closed auth
```

Nmap は、サービスがリッスンしているまたは待っている、最も一般的なネットワーク・コミュニケーション・ポートをテストします。この知識は、不必要または未使用のサービスを閉じたいと思っている管理者の助けにすることができます。

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

http://www.insecure.org/

1.3.3.2. Nessus

Nessus は完全なサービス・セキュリティ・スキャナーです。Nessus のプラグイン・アーキテクチャーはユーザーがシステムやネットワークのためにカスタマイズできるようにします。あらゆるスキャナーと同じように、Nessus は依存するシグネチャー・データベースのみと同じくらいだけ素晴らしいです。幸運にも、Nessus は頻繁にアップデートされ、完全なレポート、ホスト・スキャン、およびリアルタイムの脆弱性検索の機能を持ちます。Nessus のようにパワフルで頻繁に更新されるツールでさえ、フォールス・ポジティブやフォールス・ネガティブがある可能性があることを覚えておいてください。

注記

Nessus ソフトウェアのクライアントとサーバは Fedora リポジトリに含まれますが、使用するためのサブスクリプションが必要になります。この人気のあるアプリケーションを使用することに興味があるユーザーのための参考情報として、このドキュメントに含まれます。

Nessus に関する詳細は、以下の URL にある公式ウェブサイトを参照してください。

http://www.nessus.org/

1.3.3.3. Nikto

Nikto は優れた CGI (common gateway interface) スクリプト・スキャナーです。Nikto は CGI の脆弱性に対するチェックだけではなく、侵入検知システムを回避するために曖昧な方法で実行します。プログラムを実行するに先立って注意深くレビューされるべき完全なドキュメントがついています。ウェブサーバーが CGI スクリプトを取り扱っているならば、Nikto はこれらのサーバーのセキュリティをチェックするための優れたリソースになるでしょう。

Nikto の詳細については、以下の URL を参照してください:

http://www.cirt.net/code/nikto.shtml

1.3.3.4. VLAD the Scanner

VLAD は Bindview 社の RAZOR チームにより開発された脆弱性スキャナーです。それは、一般的なセキュリティ問題（SNMP の問題、ファイル共有の問題など）の SANS Top Ten リストに対するチェックをします。

注記

VLAD は Fedora に含まれず、サポートされません。この一般的なアプリケーションを使用することに興味があるユーザーのために参考としてこのドキュメントに含めています。

VLAD の詳細は、以下の URL にある RAZOR チームのウェブサイトで見つけられます:

http://www.bindview.com/Support/Razor/Utilities/

1.3.3.5. 将来ニーズの予測

ターゲットよりソースに依存して、利用可能な多くのツールがあります。無線ネットワーク、Novell ネットワーク、Windows システム、Linux システムなどに対するツールがあります。アセスメントを実行することの他の重要な部分は、物理セキュリティ、人事選考、または音声/PBX ネットワークのアセスメントをレビューすることを含めるかもしれません。無線ネットワークの脆弱性のために企業の物理構造の境界線をスキャンすることを含む、*war walking* のような新しい概念は、必要に応じてアセスメントに組み込み調査をできるいくつかの持ち上がってきている概念です。想像と露出は脆弱性のアセスメントを計画および実施のみに制限されます。

1.4. 一般的なエクスプロイトと攻撃

#1.1############## は、組織のネットワーク資源にアクセスするために侵入者により使用される、いくつかの最も一般的なエクスプロイトとエントリー・ポイントを詳しく説明します。これらの一般的なエクスプロイトの要点は、それらがどのように実行されるか、および、管理者がそのような攻撃に対してどのようにネットワークを適切に保護できるかの説明にあります。

表1.1 一般的なエクスプロイト

エクスプロイト	説明	注意事項
空もしくはデフォルトのパスワード	管理パスワードが空白のままになっているか、または製品ベンダにより設定されたデフォルトのパスワードを使用していることです。これはルーターやファイアウォールのようなハードウェアにおいて、かなり一般的です。Linux において実行されているいくつかのサービスは、標準の管理パスワードを含めることができます。	routers, firewalls, VPN および network attached storage (NAS) アプライアンスのようなネットワーク・ハードウェアと一般的に関連づけられます。 (UNIX や Windows のような) 多くの古いオペレーティングシステム、とくにバンドルされたサービス、において一般的です。管理者はときどき急いで特権ユーザーアカウントを作成して、パスワードを空白にしたままにします。それは、アカウントを探索している悪意のあるユーザーにとって完璧なエントリ・ポイントを作ります。
デフォルトの共有鍵	セキュアなサービスはときどき、開発者や評価テスト目的のためにデフォルトのセキュリティ鍵をパッケージしています。これらの鍵が変更されずに残っていて、インターネットの本番環境に置かれていると、同じデフォルトの鍵を持つ####ユーザーが、共有鍵の資源およびそれに含まれる機密情報すべてにアクセスできます。	無線アクセスポイントや事前設定されたセキュアなサーバー・アプリケーションにおいて最も一般的です。
IP スプーフィング	リモート・マシンは、ネットワーク資源上の制御を得るために、ローカル・ネットワークにおけるノードとして動作し、サーバにある脆弱性を見つけ、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ます。	対象システムへの接続を順序立てて並べるために、攻撃者が TCP/IP シーケンス番号を予測することを含むので、スプーフィングはかなり難しいです。しかし、いくつかのツールは攻撃者がそのような脆弱性を実行することを支援することが可能です。 #######の認証テクニックを使用することは、ターゲットシステムが実行しているサービス (rsh, telnet, FTP および他のもののような) に依存します。それは、PKI、およびssh や SSL/TLS において使われる暗号化された認証の他の形式と比較するとき、推奨されません。
盗聴	2つのノードの間のコネクションにおいて盗聴することにより、ネットワークにおける2つのアクティブなノードを通過するデータを収集します。	この種類の攻撃は大抵、Telnet, FTP, および HTTP 転送のようなプレーン・テキストの送信プロトコルとともに機能します。 リモートの攻撃者は、そのような攻撃を実行するために、LAN において危険にさらされたシステムへとアクセスできなければいけません。クラッカーは通常、LAN においてシステムを危険にさらすために能動的

エクスプロイト	説明	注意事項
		な攻撃（IP スプーフィングや中間者攻撃のような）を使用します。 防御的対策は、パスワード盗聴を防ぐために、暗号的な鍵交換、ワンタイムパスワード、または暗号化された認証を用いたサービスを含みます。転送中、強い暗号が通知されます。
サービスの脆弱性	攻撃者はインターネット上で実行されるサービスにおいて欠陥や抜け穴を見つけます。この脆弱性を通して、攻撃者はシステム全体と保持されるデータを危険にさらし、おそらくネットワークにある他のシステムも危険にさらすでしょう。	CGI のような HTTP ベースのサービスは、リモート・コマンド実行およびインタラクティブなシェル・アクセスにも脆弱です。HTTP サービスが "nobody" のような非特権ユーザーとして実行されているときでさえ、設定ファイルやネットワーク構成のような情報が読みとらる可能性があります。または、攻撃者はシステム資源を流出させたり、他のユーザーが利用不可能にしたりするサービス妨害攻撃を開始します。 サービスはときどき開発とテストの期間中に気がつかない脆弱性を持つ可能性があります。（攻撃者が、アプリケーションのメモリー・バッファを埋める任意の値を使用してサービスをクラッシュさせ、攻撃者に任意のコマンドを実行するインタラクティブなコマンド・プロンプトを与える、##########のような）これらの脆弱性により攻撃者は完全な管理コントロールを持ちます。 管理者はサービスが root ユーザーとして実行されていないことを確実にします。また、ベンダや CERT や CVE のようなセキュリティ組織から、アプリケーションに対するパッチやエラッタ・アップデートを用心深いままでいます。
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のような）に欠陥を見つけて、任意のコードを実行します、将来の侵入のためにトロイの木馬を注入します、もしくはシステムをクラッシュさせます。侵入されたワークステーションがネットワークの残りにおいて管理特権を持つならば、さらなるエクスプロイト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、作業者が侵入を防いだり検知したりする習熟や経験を持たないため、エクスプロイトをより受ける傾向にあります。認可されないソフトウェアをインストールする、または頼んでいない電子メールの添付ファイルを開くときに、とられるリスクの個々について説明することは不可欠です。 セーフガードは、電子メールソフトウェアが添付を自動的に開いたり実行したりしない、というように導入されます。加えて、Red Hat Network や他のシステム管理サービスを通してワークステーションのソフトウェアを自動更新することにより、マルチシートのセキュリティ・デプロイの負担を軽減できます。

エクスプロイト	説明	注意事項
サービス妨害 (DoS: Denial of Service) 攻撃	攻撃者や攻撃者のグループは、ターゲット・ホスト（もしくは、サーバー、ルーター、ワークステーション）へ認可されないパケットを送ることにより組織のネットワークやサーバーのリソースに対して調整されます。これはリソースを正当なユーザーに利用不可能になるよう強制します。	アメリカで最も報告された DoS 攻撃は2000年に起こりました。いくつかの高トラフィックの商用および政府のサイトが*zombies* またはリダイレクトされたブロードキャスト・ノードとして動作する高帯域接続を持ついくつかの危険にされされたシステムを用いて、調整された ping フラッド攻撃により利用不可能になりました。 ソース・パケットは通常、攻撃の本当のソース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。 `iptables` を用いたイングレス・フィルタ (IETF rfc2267) における進歩および `snort` のような Network Intrusion Detection Systems は管理者が分散された DoS 攻撃を追いかけて防ぐのを支援します。

1.5. セキュリティ・アップデート

セキュリティ脆弱性が発見されたとき、影響を受けるソフトウェアはあらゆる潜在的なリスクを制限するために更新されなければいけません。ソフトウェアが現在サポートされている Fedora ディストリビューションの中にあるパッケージの一部ならば、できる限り早く脆弱性を修正するパッケージをリリースすることをコミットします。しばしば、提供されるセキュリティ・エクスプロイトに関するアナウンスはパッチ（または問題を修正するソースコード）を伴っています。そして、このパッチは Fedora パッケージに適用され、テストされ、アップデートとしてリリースされます。しかしながら、アナウンスはパッチを含みませんので、開発者はまず問題を修正するソフトウェアのメンテナーと作業します。問題が修正されると、パッケージはテストされ、エラッタ・アップデートとしてリリースされます。

システムにおいて使用されているソフトウェアに対するエラッタ・アップデートがリリースされたならば、システムが潜在的に脆弱である時間を最小限にするため、できる限り早く影響を受けるパッケージを更新することが強く推奨されます。

1.5.1. パッケージの更新

システムにおけるソフトウェアを更新するとき、信頼されたソースからアップデートをダウンロードすることが重要です。攻撃者は、問題を修正すると思われるもののように同じバージョン番号を持ちますが、異なるセキュリティ・エクスプロイトを持つパッケージを簡単に再構築でき、インターネットにリリースできます。これが起こると、オリジナルの RPM に対するファイルの検証のようなセキュリティ対策を用いても、エクスプロイトを検知できません。このように、（Fedora のような）信頼されたソースからのみ RPM をダウンロードし、その完全性を検証するためにパッケージの署名を確認することは非常に重要です。

注記

Fedora システムに対するアップデートがあるとき、わかりやすいアラートが表示される便利なパネル・アイコンが Fedora に含まれます。

1.5.2. 署名されたパッケージの検証

Fedora のパッケージはすべて Fedora *GPG* キーを用いて署名されています。GPG は GNU Privacy Guard または GnuPG を意味する、配布ファイルの真正性を確実にするために使用されるフリー・ソフトウェアのパッケージです。たとえば、公開鍵がパッケージをロック解除して検証するまで、プライベート鍵（秘密鍵）はパッケージをロックします。Fedora により配布される公開鍵が RPM 検証中に秘密鍵と一致しなければ、パッケージは改ざんされているかもしれず、そのため信頼できません。

Fedora の中にある RPM ユーティリティは、RPM パッケージのインストール前に自動的に GPG 署名を検証しようとします。Fedora GPG キーがインストールされていないならば、Fedora インストール CD-ROM または DVD のような、安全かつ静的な場所からそれをインストールします。

ディスクが `/mnt/cdrom` にマウントされていると仮定すると、以下のコマンドを用いて *keyring* (システムにおいて信頼されたキーのデータベース) の中にインポートすることができます:

```
rpm --import /mnt/cdrom/RPM-GPG-KEY
```

RPM 検証のためにインストールされたすべてのキーを一覧表示するために、次のコマンドを実行します:

```
rpm -qa gpg-pubkey*
```

出力は以下のように見えるでしょう:

```
gpg-pubkey-db42a60e-37ea5438
```

特定のキーに関する詳細を表示するために、この例のように、前のコマンドの出力にしたがって `rpm -qi` コマンドを使用します:

```
rpm -qi gpg-pubkey-db42a60e-37ea5438
```

RPM ファイルをインストールする前に、パッケージのオリジナル・ソースから改ざんされていないことを確実にするために、それの署名を検証することは極めて重要です。ダウンロードしたパッケージを一度に検証するために、以下のコマンドを発行します:

```
rpm -K /tmp/updates/*.rpm
```

各パッケージに対して、GPG キーが正しく検証されると、コマンドは `gpg OK` を返します。そうでなければ、コンテンツのソースを検証するだけでなく、正しい Fedora 公開鍵を使用していることを確実にします。GPG 検証を通過しなかったパッケージは、第三者により改ざんされているかもしれないので、インストールすべきではありません。

GPG キーを検証して、エラッタ・レポートに関連するすべてのパッケージをダウンロードした後、シェル・プロンプトにおいて root としてパッケージをインストールします。

1.5.3. 署名されたパッケージのインストール

多くのパッケージに対するインストールは、（カーネル・パッケージを除いて、）以下のコマンドにより、安全に実行することができます:

```
rpm -Uvh /tmp/updates/*.rpm
```

カーネル・パッケージに対しては、以下のコマンドを使用します:

```
rpm -ivh /tmp/updates/<kernel-package>
```

前の例にある <*kernel-package*> をカーネル RPM の名前で置き換えます。

マシンが新しいカーネルを用いて安全に再起動されると、古いカーネルは以下のコマンドを用いて削除することができます:

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

前の例にある <*old-kernel-package*> を古いカーネル RPM で置き換えます。

注記

古いカーネルを削除することは必要ではありません。デフォルトのブートローダ GRUB は、複数のカーネルがインストールされることを許可します。そして、ブート時にメニューから選択されます。

重要

あらゆるセキュリティ・エラッタをインストールする前に、エラッタ・レポートに含まれるすべての特別な指示を確実に読み、それに応じてそれらを実行します。エラッタ・アップデートにより行われた変更を適用することに関する一般的な情報は ######## を参照してください。

1.5.4. 変更の適用

セキュリティ・エラッタとアップデートをダウンロードしてインストールした後、古いソフトウェアの使用を停止し、新しいソフトウェアの使用を開始します。これがどのように実行されるかは、更新されたソフトウェアの種類によります。以下の一覧は、ソフトウェアの一般的なカテゴリを一覧化し、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ための説明を提供します。

注記

一般に、システムを再起動することは、ソフトウェア・パッケージの最新バージョンを確実に使用するための最も確実な方法です。しかしながら、この選択肢は必ずしも必要とされません、またはシステム管理者が利用可能ではありません。

アプリケーション
: ユーザ空間アプリケーションは、システムのユーザーにより開始できるあらゆるプログラムです。一般的に、そ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されたタスク・ユーティリティがそれらを起動して、長い期間続かないときにのみ使われます。

  そのようなユーザー空間アプリケーションが更新されると、システムにあるアプリケーションのインスタンスをすべて停止して、更新したバージョンを使用するために再びプログラムを起動します。

カーネル
: カーネルは Fedora オペレーティング・システムの中心的なソフトウェア・コンポーネントです。メモリ、プロセッサおよび周辺機器へのアクセスを管理するだけでなく、すべてのタスクをスケジュールします。

その中心的な役割のため、カーネルはコンピュータを止めることなく再起動することはできません。そのため、カーネルの更新されたバージョンはシステムが再起動されるまで使うことができません。

共有ライブラリ
: 共有ライブラリは、glibc のように、多くのアプリケーションやサービスにより使用される、コードの集合です。共有ライブラリを使用しているアプリケーションは、一般的にアプリケーションが初期化されるときに共有コードをロードします。そのため、更新されたライブラリを使用しているすべてのアプリケーションは停止して再起動しなければいけません。

  実行しているアプリケーションが特定のライブラリにリンクしているかどうかを決めるために、以下の例にあるように lsof コマンドを使用します:

```
lsof /lib/libwrap.so*
```

  このコマンドは、ホストのアクセス制御用の TCP Wrappers を使用している、実行中のプログラムをすべて返します。

SysV サービス
: SysV サービスはブート中に起動される永続的なプログラムです。SysV サービスの例は、sshd, vsftpd, および xinetd を含みます。

  通常これらのプログラムはマシンがブートしている限りはメモリに永続するので、それぞれの更新された SysV サービスはパッケージが更新された後に停止して再起動しなければいけません。これは、サービス設定ツールを用いるか、rootシェル・プロンプトにログインして、以下の例にあるように /sbin/service コマンドを発行することにより実行されます。

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

  前の例において、*<service-name>* を sshd のようなサービスの名前で置き換えます。

xinetd サービス
: xinetd スーパー・サービスにより管理されているサービスは、アクティブな接続があるときのみ実行されます。xinetd により管理されるサービスの例は Telnet, IMAP, および POP3 を含みます。

  これらのサービスの新しいインスタンスは 新しいリクエストが受け取られるたびに xinetd により起動されるので、更新後に発生した接続は更新されたソフトウェアにより取り扱われます。しかしながら、xinetd に管理されたサービスが更新されたときにアクティブな接続があるならば、それらは古いバージョンのソフトウェアによりサービスされます。

  xinetd が管理している特定のサービスの古いインスタンスを止めて、サービスに対するパッケージを更新するために、現在実行中のプロセスをすべて停止します。プロセスが実行中であるかどうかを決めるために、ps コマンドを使用します。そして、現在のサービスのインスタンスを止めるために kill または killall コマンドを使用します。

  たとえば、imap パッケージのセキュリティ・エラッタがリリースされ、パッケージを更新したならば、シェル・プロンプトの中で root として以下のコマンドを入力します:

```
ps -aux | grep imap
```

  このコマンドはすべてのアクティブな IMAP セッションを返します。各セッションは以下のコマンドを発行することにより停止できます:

```
kill <PID>
```

  これがセッションを停止するのに失敗したら、代わりに以下のコマンドを使用します:

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

前の例において、*<PID>* を IMAP セッションに対するプロセス識別番号（ps コマンドの2番目の列で見つけられます）に置き換えます。

すべてのアクティブな IMAP セッションを止めるために、以下のコマンドを発行します:

```
killall imapd
```

基本強化ガイド

Every computer system should be hardened against threats found both over the network as well as those found physically at the computer. The system changes are necessary based on default settings usually being set to allow software to work over the software being secure. As with any change to a system these changes could cause unintended results. Changes should be evaluated for appropriateness on your system before implementing.

2.1. 基本原則

ネットワーク経由で転送されるすべてのデータを暗号化します。認証情報（パスワードなど）を暗号化することはとくに重要です。
脆弱性を最小化するために、インストールおよび実行されているソフトウェアの量を最少化します。
利用可能なときはセキュリティ強化されたソフトウェアおよびツール（例えば、SELinux や IPTables）を使用します。
各ネットワークサービスをできる限り別々のサーバーにおいて実行します。これにより、あるサービスのセキュリティ侵害により他のものへの侵害につながるリスクを最小化します。
ユーザーアカウントを維持します。良いパスワードポリシーを作成して、その使用を強制します。使用していないユーザーアカウントを削除します。
定常業務としてシステムログとアプリケーションログを確認します。ログを集中ログサーバーに送信します。これにより、侵入者がローカルログを改ざんすることにより、簡単に検知されないようにすることを防ぎます。
絶対に必要なとき以外は、直接 root としてログインしません。管理者は、必要なときに root としてコマンドを実行するために `sudo` を使用すべきです。sudo を使用する能力のあるアカウントは `/etc/sudoers` に指定されます。これは visudo ユーティリティを用いて編集されます。関連するログは標準で `/var/log/secure` に書き込まれます。

2.2. 物理セキュリティ

Physical security of the system is of utmost importance. Many of the suggestions given here won't protect your system if the attacker has physical access to the system. Physical access doesn't necessarily mean that the battle is lost, however. Strengthening your BIOS and boot software can help defend your system against certain types of attacks.

Configuring the BIOS to disable booting from CDs/DVDs, floppies, and external devices, can prevent bypassing the boot partition and the boot loader where other protections are in place. It is important to password-protect your BIOS settings so that an attacker cannot just change these, and other, settings. Next, set a password for the GRUB bootloader. Use the grub2-mkpasswd-pbkdf2 to create your password hash. This prevents users from entering single user mode or changing settings at boot time.

2.3. これはなぜ重要なのでしょうか？

攻撃者が外部ソースからブートすることによりシステムの完全な制御をとることができます。外部ソース（たとえば live Linux CD）からブートすることにより、多くのセキュリティ設定が回避されます。攻撃者は GRUB の設定を変更することができるならば、システムへの管理者アクセスが可能になるシングルユーザーモードでブートすることができます。

Additional explaination and hardening can be found in *#BIOS ################* of this guide.

2.4. ネットワーク

コンピューターのネットワーク接続はシステムへの入り口です。ファイルおよびプロセッサー時間は、他の保護機能が実装されていなければ、ネットワーク接続経由でシステムに正常に接続した、すべての人に利用可能です。

システムをコントロールした状態にしておく主要な方法の一つは、攻撃者が最初の場所でシステムにアクセスできないようにしておくことです。

2.4.1. iptables

iptables は今日 Linux システムにおいてもっとも広く使用されているファイアウォールソフトウェアです。このプログラムは、ネットワーク接続経由でコンピューターに受信したパケットを横取りします。そして、指定されたルールに基づいてそれらをフィルターします。

2.4.2. IPv6

IPv6 は最新のインターネットプロトコルです。アドレス不足を解決することを目指した IPv4 の後継です。また、新しいプロトコルに関連した直接的なセキュリティリスクはありません。この新しい技術を利用する前に理解することがいくつかあります。

多くのシステム管理者は IPv4 に慣れています。そして、正しく IPv4 を動作させるために加えられた暫定対処について慣れています。これらの暫定対処の一つはネットワークアドレス変換 *NAT* です。NAT は慣習的に、ローカルエリアネットワークを構築するときに、必要となるパブリック IP アドレスの数を最小限にするために使用されています。これらのネットワークにあるシステムはパブリック IP アドレスを必要としません。また、重要なアドレス空間がこれらの技術を実装することにより節約できます。NAT による副作用としていくつかのセキュリティ機能があります。もっとも大きなものは、ポートがルーターを越えて転送されない限り、外部の通信をネットワークの内部に入れないことです。IPv6 はアドレス問題を解決するので、もはや NAT を使用する必要¥nはありません。すべてのものがパブリック IP アドレスを持てます。さらに、拡張することにより、すべてのものが物理的および論理的に接続されているとき、インターネットをまたがりパブリックにルート可能ではありません。

心配するべきもう一つのことは、セキュリティソフトウェアがこの新しいプロトコルをどのように処理するかです。iptables は IPv6 を認識もしくは理解しません。そのため、これらのパケットを無視します。つまり、ネットワークが IPv6 を利用し、ip6tables を有効化していなければ、システムを世界中に向けて開け放っていることになります。

システムのソフトウェアがこの新しいネットワークプロトコルを使用できるという、変更点を把握して理解している限り、IPv6 を使用することは危険ではありません。

2.5. ソフトウェアの最新化維持

ソフトウェアは毎日パッチをあてられます。これらの更新のいくつかは、開発者により識別されたセキュリティ問題を修正します。これらのパッチが利用可能になったとき、できる限り早くシステムに適用することが重要です。システムの更新を管理するもっとも簡単な方法の一つは yum を使用することです。バグ修正と機能拡張を無視して、セキュリティ更新のみをインストールできるようにする、特別なプラグインが利用可能です。このプラグインは*#YUM ######*においてより詳しく説明しています。

2.6. サービス

Linux におけるサービスは、バックグラウンドにおいてデーモンとして実行されるプログラムです。実行する必要があるかどうかを決めるために、これらのプログラムを定期的に監査することが重要です。多くのデーモンは呼び出しをリッスンするためにネットワークのポートを開きます。不必要なポートを開いておくことにより、システム全体のセキュリティを危険にさらす可能性があります。あるソフトウェアの未知のセキュリティ侵害により、攻撃者がシステムの中に不正な理由で侵入できるようになる可能性があります。

2.7. NTP

Network Time Protocol (*NTP*) はシステムの時刻を正確に保ちます。時間はセキュリティのパズルの非常に重要なピースであり、できる限り正確に維持するべきです。時間は、ログファイル、タイムスタンプおよび暗号にお

いて使用されます。誰かがシステムにおいて時刻設定を制御できるならば、侵入の再現をより難しくすることができます。

ネットワークのセキュア化

3.1. ワークステーションのセキュリティ

Linux 環境をセキュアにすることはワークステーションから始めます。個人のマシンをロックするか企業システムをセキュアにするかどちらかに関わらず、健全なセキュリティ・ポリシーが個々のコンピュータから始まります。コンピュータ・ネットワークも最も弱いノードと同じくらいだけの安全性しかありません。

3.1.1. ワークステーションのセキュリティの評価

Fedora ワークステーションのセキュリティを評価するとき、以下を考慮します:

- *BIOS* ########## — 認可されないユーザーがマシンに物理的にアクセスして、パスワードなしでシングルユーザーモードまたはレスキューモードにてブートできますか?
- ########## — マシンのユーザー・アカウントのパスワードはどのくらいセキュアですか?
- ######### — 誰がシステムにアカウントを持ちますか、そしてどのくらいの管理的コントロールを持ちますか?
- ########## — どのサービスがネットワークからのリクエストを待ち受けていますか、またそれらはすべて実行すべきですか?
- ########## — もしあれば、どのタイプのファイアウォールが必要とされますか?
- ########## — どのツールがワークステーション間の通信に使用され、どれが避けられるべきでしょうか?

3.1.2. BIOS とブートローダのセキュリティ

BIOS とブートローダに対するパスワードの保護は、システムに物理的にアクセスできる認可されないユーザーが、リムーバブル・メディアを使用してブートしたり、シングルユーザーモードで root 特権を得たりすることを防げます。そのような攻撃に対する保護を得るためにとるべきセキュリティ対策は、ワークステーションにおける情報の機密性とマシンの場所に依存します。

たとえば、信頼された人々のみがアクセスできる安全な場所においてマシンが使用され、コンピュータが機密情報を含まないならば、そのような攻撃を防ぐことは致命的ではないかもしれません。しかしながら、会社のネットワークに対するプライベートな暗号化されていない SSH キーを持つ従業員のラップトップが展示会に出席されずに残っているならば、会社全体に対する分岐を持つ主要なセキュリティ侵害につながるでしょう。

3.1.2.1. BIOS パスワード

コンピュータの BIOS をパスワードで保護するおもな2つの理由は次のとおりです[1]:

1. *BIOS* ######## — 侵入者が BIOS へのアクセス権を持つならば、ディスクや CD-ROM からブートするよう設定できます。これにより、システムにおいて任意のプロセスを開始したり機密データをコピーしたりできるようにする、レスキューモードやシングルユーザーモードに入ることができるようになります。
2. ########## — いくつかの BIOS はブート・プロセスのパスワード保護を許可します。有効化されたとき、攻撃者は BIOS がブートローダを起動する前にパスワードを入力することが強制されます。

[1] システム BIOS は製造者間で異なるので、いくつかはどちらのタイプのパスワード保護もサポートしないかもしれません。一方、他のものは1つのタイプをサポートするかもしれませんが、さらに他のものはそうでないかもしれません。

BIOS パスワードを設定する方法はコンピュータ製造者間で異なるため、詳細な説明はコンピュータのマニュアルを参照してください。

もし BIOS パスワードを忘れたならば、マザーボードにあるジャンパーを用いてリセットします、または CMOS バッテリーを外します。このため、可能ならばコンピュータのケースをロックすることはグッド・プラクティスです。しかし、CMOS バッテリーを外そうとする前にコンピュータまたはマザーボードのマニュアルを参照してください。

3.1.2.1.1. 非 x86 プラットフォームのセキュア化

他のアーキテクチャは低レベルのタスク（x86 システムにおける BIOS のそれらとほぼ同等）を実行するために異なるプログラムを使用します。たとえば、Intel® Itanium™ コンピュータは *Extensible Firmware Interface* (*EFI*) シェルを使用します。

他のアーキテクチャにおける BIOS のようなプログラムをパスワード保護することの説明は、製造者の説明書を参照してください。

3.1.2.2. ブートローダのパスワード

Linux ブートローダをパスワードで保護する主要な理由は以下のとおりです:

1. ########################## — 攻撃者がシングルユーザーモードでシステムをブートできるならば、root パスワードを聞かれることなく自動的に root としてログインされます。
2. *GRUB* ################ — マシンがブートローダとして GRUB を使用していると、攻撃者は `cat` コマンドを用いて、設定を変更したり情報を集めたりするために、GRUB 編集インタフェースを使用できます。
3. ################################ — もしデュアルブートのシステムであれば、攻撃者は、アクセス制御とファイル・パーミッションを無視して、ブート時にオペレーティングシステム（たとえば、DOS）を選択できます。

Fedora ships with the GRUB boot loader on the x86 platform. For a detailed look at GRUB, refer to the Fedora Installation Guide.

3.1.2.2.1. GRUB のパスワード保護

You can configure GRUB to address the first two issues listed in ############### by adding a password directive to its configuration file.

システムが次回起動するとき、GRUB メニューが `p` に続けて GRUB パスワードをまず入力するまで、エディタまたはコマンド・インタフェースにアクセスするのを防ぎます。

3.1.3. パスワードのセキュリティ

パスワードは Fedora がユーザーのアイデンティティを検証するために使用される第一の方法です。これは、パスワード・セキュリティがユーザー、ワークステーション、およびネットワークの保護のために非常に重要である理由です。

セキュリティ目的のために、インストール・プログラムはシステムが *Message-Digest Algorithm* (*MD5*) および shadow パスワードを使用するよう設定します。これらの設定を変更しないことが強く推奨されます。

MD5 パスワードがインストール中に選択解除されていると、古い *Data Encryption Standard* (*DES*) 形式が使用されます。この形式はパスワードを英数字8文字に制限し、少量の56ビット・レベルの暗号化を提供します。

shadow パスワードがインストール中に選択解除されていると、すべてのパスワードが全ユーザーが読み込める `/etc/passwd` ファイルに一方向ハッシュとして保存されます。それは、システムをオフライン・パスワード・ク

ラック攻撃に対して脆弱にします。侵入者が通常のユーザーとしてマシンへのアクセス権を得られると、/etc/passwd ファイルを自分自身のマシンにコピーして、それに対してパスワード・クラック・プログラムをいくらでも実行できます。ファイルにセキュアではないパスワードがあると、パスワード・クラッカーがそれを発見するのは時間の問題です。

shadow パスワードは、root ユーザーのみが読み込める、/etc/shadow ファイルにパスワード・ハッシュを保存することにより、この種類の攻撃を取り除きます。

これは、マシンにおける SSH や FTP のようなネットワーク・サービスにログインすることにより、潜在的な攻撃者がパスワード・クラックをリモートで試みることを強制します。この種のブルートフォース攻撃は、より遅く、数百ものログイン失敗の試みがシステムファイルに書き込まれるので、明らかな証拠を残します。もちろん、クラッカーが弱いパスワードを持つシステムに夜の中ごろに攻撃を始めると、クラッカーは夜明け前にアクセス権を得て、形跡を覆い隠すためにログファイルを編集しているかもしれません。

形式と保存に関する考慮点に加えて、コンテンツの問題があります。ユーザーがパスワード・クラック攻撃に対して自分のアカウントを保護するためにできる、最も重要なことの1つは、強いパスワードを生成することです。

3.1.3.1. 強いパスワードの作成

安全なパスワードを作成するとき、これらのガイドラインに従うことは素晴らしいアイディアです:

- *Do Not Use Only Letters or Numbers* — Using only letters or numbers does not make for a complex password.

  いくつかの安全ではない例は以下を含みます:

  - 8675309
  - juan
  - hackme

- ############### — 固有名詞、辞書の単語、またはテレビ番組や小説からの単語さえ、そのような単語は最後に番号をつけたとしても避けるべきです。

  いくつかの安全ではない例は以下を含みます:

  - john1
  - DS-9
  - mentat123

- ############## — パスワード・クラッキング・プログラムはしばしば多くの言語の辞書を網羅する単語リストをチェックします。セキュアなパスワードのために外国語に依存することは、セキュアではありません。

  いくつかの安全ではない例は以下を含みます:

  - cheguevara
  - bienvenido1
  - 1dumbKopf

- ############## — ハッカー用語（l337 (LEET) speak とも言われます）を使用するので、あなたがエリートであると考えているならば、パスワードにおいては考え直してください。多くの単語リストは LEET speak を含みます。

  いくつかの安全ではない例は以下を含みます:

  - H4X0R
  - 1337
- ########## — パスワードにあらゆる個人情報を使用するのを避けます。攻撃者があなたのアイデンティティを知っているならば、パスワードを推測する作業はより簡単になります。以下はパスワードを作成するときに避ける情報の種類の一覧です:

  いくつかの安全ではない例は以下を含みます:

  - あなたの名前
  - ペットの名前
  - 家族の名前
  - Any birth or anniversary dates
  - 電話番号や郵便番号
- ############### — 良いパスワード・チェッカーは常に一般的な単語を逆順にします。そのため、悪いパスワードを逆順にすることはまったくセキュアにしません。

  いくつかの安全ではない例は以下を含みます:

  - R0X4H
  - nauj
  - 9-DS
- ############ — パスワードを紙に保存しない。記録することはよりずっと安全です。
- ########################## — 各マシンに対して別々のパスワードを作ることは重要です。このように、あるシステムが危険にさらされているならば、すべてのマシンが直ちにリスクにさらされることはありません。

以下のガイドラインは強いパスワードを作成する助けになります:

- *Make the Password at Least Twelve Characters Long* — The longer the password, the better. If using MD5 passwords, it should be 15 characters or longer.
- *Mix Upper and Lower Case Letters* — Fedora uses case sensitive passwords, so mix cases to enhance the strength of the password.
- ########## — パスワードに数字を追加すること、とくに（単に最初または最後ではなく）真ん中に追加するとき、パスワードの強度を向上させることができます。
- ############ — &, $, および > のような特別な文字はパスワードの強度を非常に上げます（DES パスワードを使用していると、これはできません）。
- ############## — あなたがパスワードを覚えられなければ、世界で最も良いパスワードはほとんど良くありません。パスワードを記憶する助けにするために頭文字または他の記憶装置を使用します。
- *Use a Password Generator* — Using a random password generator, along with secure password storage software, can make it very difficult for an attacker to discover your password.

これらのルールすべてを用いて、悪いものの特徴を避ける一方で、素晴らしいパスワードの基準のすべてに適合するパスワードを作成することは難しいかもしれません。幸運にも、覚えることが簡単かつセキュアなパスワードを生成するためにとることができるいくつかの手順があります。

3.1.3.1.1. セキュアなパスワードの作成方法

人々がセキュアなパスワードを作成するために使う方法がいくつもあります。最も一般的な方法の1つは頭文字を含めることです。たとえば:

- 以下のような簡単で覚えやすいフレーズを考えます:

  "over the river and through the woods, to grandmother's house we go."

- 次に、(句読点を含めて)頭文字にします。

  `otrattw,tghwg.`

- 頭文字にある文字を数字と記号に置き換えることにより複雑さを追加します。たとえば、t を 7 に、a をアットマーク記号 (@) に置き換えます:

  `o7r@77w,7ghwg.`

- H のように、少なくとも1文字を大文字にすることでさらに複雑性を追加します。

  `o7r@77w,7gHwg.`

- ##。

セキュアなパスワードを作成することが不可欠である一方、それらを適切に管理することも重要です。とくに、大きな組織の中のシステム管理者にとってはそうです。以下のセクションは、組織の中においてユーザー・パスワードを作成および管理することのグッド・プラクティスを詳細に説明します。

3.1.3.2. 組織内でのユーザー・パスワードの作成

組織が多くのユーザーを持っているならば、システム管理者は良いパスワードの使用を強制するために利用可能な基本的なオプションが2つあります。ユーザーのためにパスワードを作成できます。もしくは、パスワードが受け入れられる質であることを検証している間、ユーザー自身がパスワードを作成できるようにします。

ユーザーのためにパスワードを作成することは、パスワードが良いものであることを確実にしますが、組織が大きくなるにつれて気の重い作業になります。ユーザーが自分のパスワードを書きとめるリスクも上昇します。

これらの理由により、多くのシステム管理者は、ユーザー自身がパスワードを作成することを好みますが、パスワードが良いものであることを実際に確認します。いくつかの場合では、パスワード・エージングを通してユーザーが定期的にパスワードを変更することを強制します。

3.1.3.2.1. 強いパスワードの強制

侵入からネットワークを保護するために、システム管理者が組織の中で使われるパスワードが強いものであることを検証することは素晴らしいアイディアです。ユーザーがパスワードを生成または変更したいとき、コマンドライン・アプリケーション `passwd` を使用できます。これは、*Pluggable Authentication Manager* (*PAM*) に対応していて、そのためパスワードが短すぎたり、さもなければクラックしやすいかったりするかを確認するためにチェックします。このチェックは `pam_cracklib.so` PAM モジュールを使用することにより実行されます。PAM はカスタマイズ可能なので、`pam_passwdqc` (*http://www.openwall.com/passwdqc/* から利用可能) のようなパスワード完全性チェッカーを追加することが可能です、または新しいモジュールを書くことが可能です。利用可能な PAM モジュールのリストのために、*http://www.kernel.org/pub/linux/libs/pam/modules.html* を参照してください。PAM の詳細は *#Pluggable Authentication Modules (PAM)#* を参照してください。

パスワードチェックは作成されるときに実行され、パスワードに対してパスワード・クラック・プログラムを実行するように効果的に悪いパスワードを発見できません。

多くのパスワード・クラック・プログラムは、オペレーティングシステムに同梱されていないにも関わらず、Fedoraで実行するものが利用可能です。以下は最も一般的なパスワード・クラック・プログラムのいくつかの簡単なリストです:

- ***John The Ripper*** — 速くて柔軟なパスワード・クラック・プログラム。複数の単語リストを使用でき、ブルートフォース・パスワード・クラックをできます。*http://www.openwall.com/john/* においてオンラインで利用可能です。
- ***Crack*** — おそらく最もよく知られたパスワード・クラック・ソフトウェア。Crack は非常に速いですが、John The Ripper ほど使うのが簡単ではありません。*http://www.crypticide.com/alecm/security/crack/c50-faq.html* においてオンラインで利用可能です。
- ***Slurpie*** — Slurpie は John The Ripper および Crack と似ていますが、分散パスワード・クラック攻撃を生成して、並行して複数のコンピューターで実行するよう設計されています。*http://www.ussrback.com/distributed.htm* においてオンラインで、数多くの他の分散攻撃セキュリティ評価ツールとともに見つけられます。

警告

組織内でパスワードをクラックする試行を始める前に常に書面で認可を得てください。

3.1.3.2.2. パスフレーズ

パスフレーズとパスワードは今日のシステムの多くにおいてセキュリティの基礎です。不幸にも、バイオメトリクスや2要素認証のような技術は、多くのシステムにおいて主流になってきていません。パスワードがシステムをセキュアにするために使われるようになってくると、パスフレーズの使用が検討されるべきです。パスフレーズは、数字や記号のような標準的ではない文字とともに導入されるとき、パスワードよりも長く、パスワードよりも良い保護を提供します。

3.1.3.2.3. パスワード・エージング

パスワード・エージングは、組織の中で悪いパスワードを防御するためにシステム管理者により使用されるもう1つのテクニックです。パスワード・エージングは、指定された期間（通常90日）経過後、ユーザーは新しいパスワードを作成するようプロンプトが出されることを意味します。この後ろにある理論は、ユーザーが定期的にパスワードを変更することを強制されるならば、クラックされたパスワードが限られた期間のみ侵入者にとって有用である、というものです。しかしながら、パスワード・エージングの不利な面は、ユーザーがパスワードをより書きとめるかもしれないことです。

Fedora 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が2つあります。chage コマンドまたはグラフィカルなユーザー管理 (system-config-users) アプリケーション。

chage コマンドの -M オプションは、パスワードが有効である最大日数を指定します。たとえば、ユーザーのパスワードを90日で期限切れに設定するために、以下のコマンドを使用します:

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

上のコマンドで、*<username>* をユーザーの名前で置き換えます。パスワードの期限切れを無効にするために、伝統的に -M オプションの後ろに 99999 の値（273年と少しと同じです）を使用します。

複数のパスワード・エージングとアカウントの詳細を変更するためにインタラクティブ・モードにおいて chage コマンドを使用することもできます。インタラクティブ・モードに入るために以下のコマンドを使用します:

```
chage <username>
```

以下はこのコマンドを用いたインタラクティブなセッションのサンプルです:

```
[root@myServer ~]# chage davido
Changing the aging information for davido
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]:
[root@myServer ~]#
```

利用可能なオプションの詳細は chage のマニュアル・ページを参照してください。

パスワード・エージング・ポリシーを作成するために、グラフィカルなユーザー・マネージャー・アプリケーション を使用することもできます。注記:この手順を実行するために管理者特権が必要になります。

1. ユーザー・マネージャーを表示するために、パネルにあるシステムメニューをクリックして、管理をポイントして、ユーザーとグループをクリックします。代わりに、シェル・プロンプトにおいて system-config-users コマンドを入力します。
2. ユーザータブをクリックして、ユーザーのリストにおいて必要なユーザーを選択します。
3. ユーザー・プロパティのダイアログ・ボックスを表示するためにツールバーにおいてプロパティをクリックします(または、ファイルメニューのプロパティを選択します)。
4. パスワード情報タブをクリックして、パスワードの有効期限を有効にするためにチェックボックスを選択します。
5. 変更が必要になるまでの日数フィールドに必要な値を入力して、OKをクリックします。

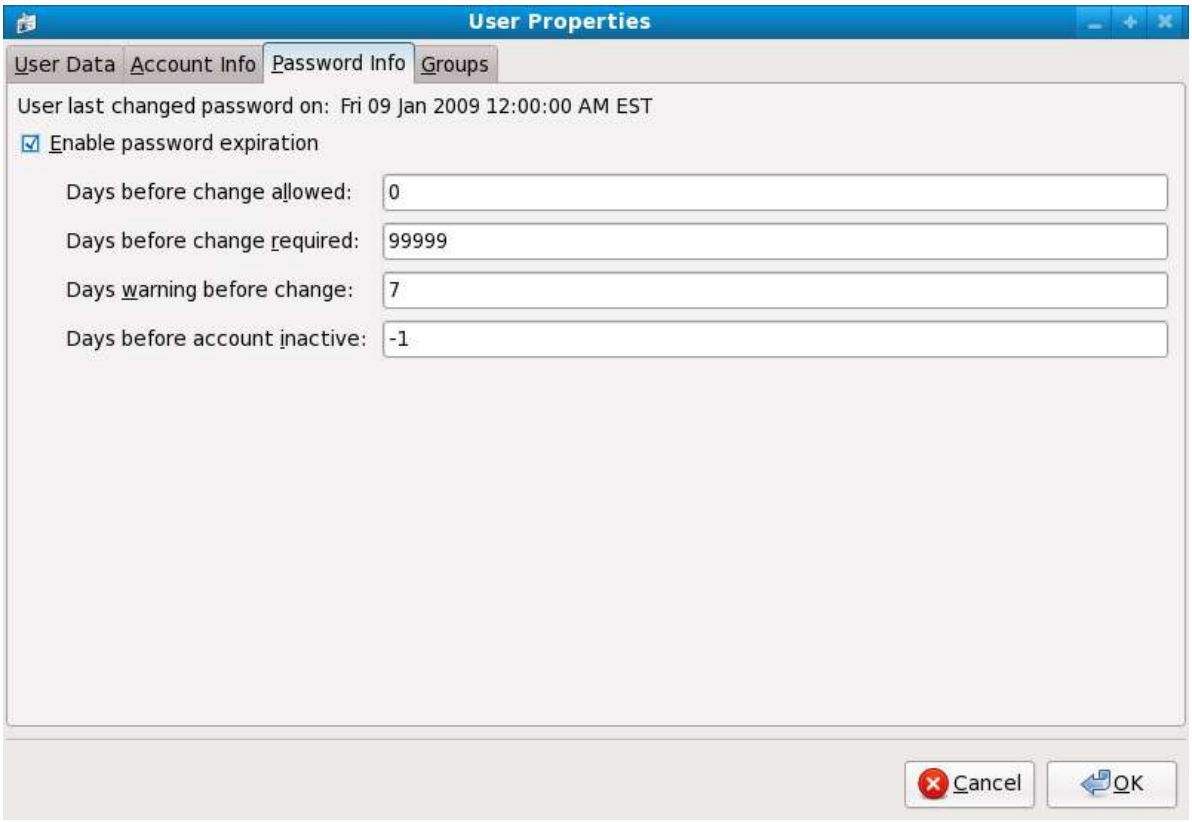

図3.1 パスワード・エージングのオプションの指定

3.1.4. 管理的コントロール

自宅のマシンを管理しているとき、root ユーザーとして、または sudo や su のような *setuid* プログラムを経由して効果的な root 特権を取得することにより、ユーザーはいくつかのタスクを実行しなければいけません。setuid プログラムは、プログラムを実行してい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*) で実行されるものです。そのようなプログラムは、以下の例にあるように、ロング形式リストの所有者セクションに s により表現されます:

```
-rwsr-xr-x 1 root root 47324 May 1 08:09 /bin/su
```

注記

s は大文字または小文字かもしれません。大文字で表示されるならば、基礎となるパーミッション・ビットがセットされていないことを意味します。

しかしながら、組織のシステム管理者に対して、組織の中のユーザーがマシンへどのくらいの管理的アクセスを持たせるかの選択をしなければいけません。pam_console.so と呼ばれる PAM モジュールを通して、リブートやリムーバブル・メディアのマウントのような、通常 root ユーザーに対してのみ指定されたいくつかのアクティビティは、物理コンソールにログインした最初のユーザーに対して許可されます。(pam_console.so モジュールの詳細は *#Pluggable Authentication Modules (PAM)#* を参照してください。)しかしながら、ネットワーク設定の変更、新しいマウスの設定やネットワーク・デバイスのマウントのような、他のシステム管理的タスクは管理特権なしでは不可能です。結果として、システム管理者はネットワークにおけるどのくらいのユーザーがアクセス権を受け取るかを決めなければいけません。

3.1.4.1. 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・リテラシがあるならば、root アクセスを許可することは問題ないかもしれません。ユーザーによる root アクセス権を許可することは、デバイスの追加またはネットワークインタフェースの設定のような軽微な活動が個々のユーザにより取り扱われることを意味します。システム管理者をネットワーク・セキュリティおよび他の重要な問題を取り扱うことから開放します。

他方、個々のユーザーに root アクセス権を与えることは、以下の問題につながる可能性があります:

- ######## — root アクセス権を持つユーザーは、マシンの設定を誤り、問題を解決するために支援を必要とする可能性があります。さらに悪いことに、意識せずにセキュリティ・ホールを開けるかもしれません。
- ################ — root アクセス権を持つユーザーは、潜在的にユーザー名とパスワードをリスクにさらす、FTP や Telnet のようなセキュアではないサーバーをマシンにおいて実行するかもしれません。これらのサービスはこの情報をネットワーク上で平文で転送します。
- *root #### email #####* — 珍しいことですが、Linux に影響する email ウイルスが存在します。しかしながら、それらが脅威である唯一のときは、それらが root ユーザーとして実行されたときです。

3.1.4.2. root アクセスの不許可

管理者はユーザーに root としてログインできるようにすることが気持ち悪ければ、root パスワードは秘密にしておくべきです。また、ランレベル1やシングルユーザーモードへのアクセスはブートローダー・パスワード保護を通して無効にされるべきです。(この話題の詳細は ############### を参照してください。)

#3.1#root ############ は、管理者が root ログインを無効にされていることをさらに確実にすることができる方法について説明しています:

表3.1 root アクセスを無効化する

方法	説明	効果	影響なし
root シェルの変更	/etc/passwd ファイルを編集して、シェルを /bin/bash から /sbin/nologin に変更します。	root シェルへのアクセスを防ぎ、そのような試行をすべて記録する。 以下のプログラムは root アカウントへのアクセスを防がれます: · login · gdm · kdm · xdm · su · ssh · scp · sftp	FTP クライアント、メール・クライアント、および多くの setuid プログラムのような、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 アカウントへのアクセスを防ぎ###: · sudo · FTP クライアント · Email クライアント
すべてのコンソール・デバイス (tty) 経由の root アクセスの無効化	空の /etc/securetty ファイルは、コンピュータに接続されたすべてのデバイスに root ログオンするのを防ぎます。	コンソールまたはネットワーク経由で root アカウントへアクセスするのを防ぎます。以下のプログラムは root アカウントにアクセスするのを防ぎます: · login · gdm · kdm · xdm	root としてログインしないが、setuid や他のメカニズムを通して管理的なタスクを実行するプログラム。 以下のプログラムは root アカウントへのアクセスを防ぎ###: · su · sudo · ssh · scp

方法	説明	効果	影響なし
		· tty を開く他のネットワーク・サービス	· sftp
root SSH ログインの無効化	/etc/ssh/sshd_config ファイルを編集して、PermitRootLogin パラメータを no にセットします。	OpenSSH スイートのツール経由による root アクセスを防ぎます。以下のプログラムは root アカウントにアクセスするのを · ssh · scp · sftp	これは OpenSSH スイートのツールのみへと root アクセスを防ぎます。
サービスへの root アクセスを制限するために PAM の使用	/etc/pam.d/ ディレクトリにある対象サービスのファイルを編集します。pam_listfile.so が認証のために必要であることを確実にします。[1]	PAM に対応するネットワーク・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 アカウントへのアクセスを防ぎます: · FTP クライアント · Email クライアント · login · gdm · kdm · xdm · ssh · scp · sftp · すべての PAM 対応アプリケーション	PAM に対応しないプログラムおよびサービス。

[1] 詳細は*#PAM #### root #####*を参照してください。

3.1.4.2.1. root シェルの無効化

ユーザーが root として直接ログインすることを防ぐために、システム管理者は /etc/passwd ファイルにおいて root アカウントのシェルを /sbin/nologin に設定できます。これにより、su や ssh コマンドのような、シェルを要求するコマンドを通して root アカウントにアクセスすることを防ぎます。

重要

email クライアントや sudo コマンドのような、シェルへのアクセスを必要としないプログラムは、まだ root アカウントにアクセスすることができます。

3.1.4.2.2. root ログインの無効化

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 ファイルを編集することにより、コンソールに root ログインすることを無効にできます。このファイルは root ユーザーがログインを許可されているすべてのデバイスをリストします。ファイルがまったく存在しなければ、root ユーザーは、コンソール経由かロー・ネットワーク・デバイスかによらず、システムにあるすべてのコミュニケーション・デバイスを通してログインできます。ネットワーク上に平文でパスワードを転送する、Telnet 経由で root としてマシンにログインできるので、これは危険です。デフォルトで、Fedora の /etc/securetty ファイルは、root ユーザーがマシンに物理的

に接続されたコンソールのみにログインできます。root がログインするのを防ぐため、以下のコマンドを入力することによりこのファイルの内容を削除します:

```
echo <username> /etc/securetty
```

警告

空の /etc/securetty ファイルは、コンソールが認証される後まで開かれないので、root ユーザーが OpenSSH スイートのツールを用いてリモートログインするのを防ぎ###。

3.1.4.2.3. root SSH ログインの無効化

SSH プロトコル経由の root ログインは Fedora においてデフォルトで無効化されています。しかし、このオプションが有効化されているならば、SSH デーモンの設定ファイル (/etc/ssh/sshd_config) を編集することにより再び無効化できます。それが読み込む行を変更します:

```
PermitRootLogin yes
```

以下のように読み込むために:

```
PermitRootLogin no
```

これらの変更が効くために、SSH デーモンが再起動されなければいけません。これは以下のコマンドを通して実行できます。

```
kill -HUP `cat /var/run/sshd.pid`
```

3.1.4.2.4. PAM を用いた root の無効化

/lib/security/pam_listfile.so モジュールを通して PAM は特定のアカウントを拒否するときに大きな柔軟性を許します。管理者はログインが許可されないユーザーのリストを参照するためにこのモジュールを使用できます。以下は、モジュールが /etc/pam.d/vsftpd PAM 設定ファイルにおいて vsftpd FTP サーバーのためにどのように使用されるかの例です。(以下の例で最初の行の最後にある ¥ 文字は、ディレクティブが1行にあるならば必要#####):

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

これは PAM に /etc/vsftpd.ftpusers ファイルを参照して、リストされたユーザーすべてを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ます。また、複数のサービスへのアクセスを拒否するために、各サービスに対して別々のリストを保つことも、1つの集中したライスとを使用することもできます。

管理者が複数のサービスへのアクセスを拒否したいならば、次の同じような行が PAM 設定ファイルに追加されます。メールクライアントに対しては /etc/pam.d/pop および /etc/pam.d/imap、SSH クライアントに対して /etc/pam.d/ssh です。

PAM の詳細は *#Pluggable Authentication Modules (PAM)#* を参照してください。

3.1.4.3. root アクセスの制限

管理者は root ユーザーへのアクセスを完全に拒否するより、su や sudo のような、setuid プログラム経由でのみアクセスを許可したいと考えるかもしれません。

3.1.4.3.1. su コマンド

ユーザーが su コマンドを実行するとき、root パスワードに対するプロンプトが出されます。認証後、root シェルプロンプトが与えられます。

一度 su コマンド経由でログインすると、ユーザーは root ユーザー###、システムへの絶対的な管理アクセス権を持ちます。[2]. さらに、一度ユーザーが root になると、パスワードをプロンプトされることなくシステムにある他のすべてのユーザーに変更するために su コマンドを使用できます。

このプログラムは非常に強力であるため、組織の中にいる管理者はコマンドにアクセス権を持つ者を制限したいと思うかもしれません。

これを実行する最も簡単な方法の1つは、*wheel* と呼ばれる特別な管理グループにユーザーを追加することです。これをするために、root として以下のコマンドを入力します:

```
usermod -G wheel <username>
```

前のコマンドにおいて、*<username>* を wheel グループに追加したいユーザー名で置き換えます。

グループメンバーを変更するために、以下のようにユーザー管理を使用することもできます。注記：この手順を実行するために管理者権限を必要とします。

1. ユーザー・マネージャーを表示するために、パネルにあるシステムメニューをクリックして、管理をポイントして、ユーザーとグループをクリックします。代わりに、シェル・プロンプトにおいて system-config-users コマンドを入力します。
2. ユーザータブをクリックして、ユーザーのリストにおいて必要なユーザーを選択します。
3. ユーザー・プロパティのダイアログ・ボックスを表示するためにツールバーにおいてプロパティをクリックします（または、ファイルメニューのプロパティを選択します）。
4. グループ タブをクリックして、wheel グループのチェックボックスを選択して、OK をクリックします。*# 3.2###### "wheel" ############* を参照してください。
5. su に対する PAM 設定ファイル (/etc/pam.d/su) をテキストエディターで開き、以下の行からコメント # を削除します:

   ```
   auth  required /lib/security/$ISA/pam_wheel.so use_uid
   ```

   この変更は、管理グループ wheel のメンバーだけがこのプログラムを使用できることを意味します。

[2] このアクセス権は SELinux が有効ならば、それにより課される制限をまだ受けます。

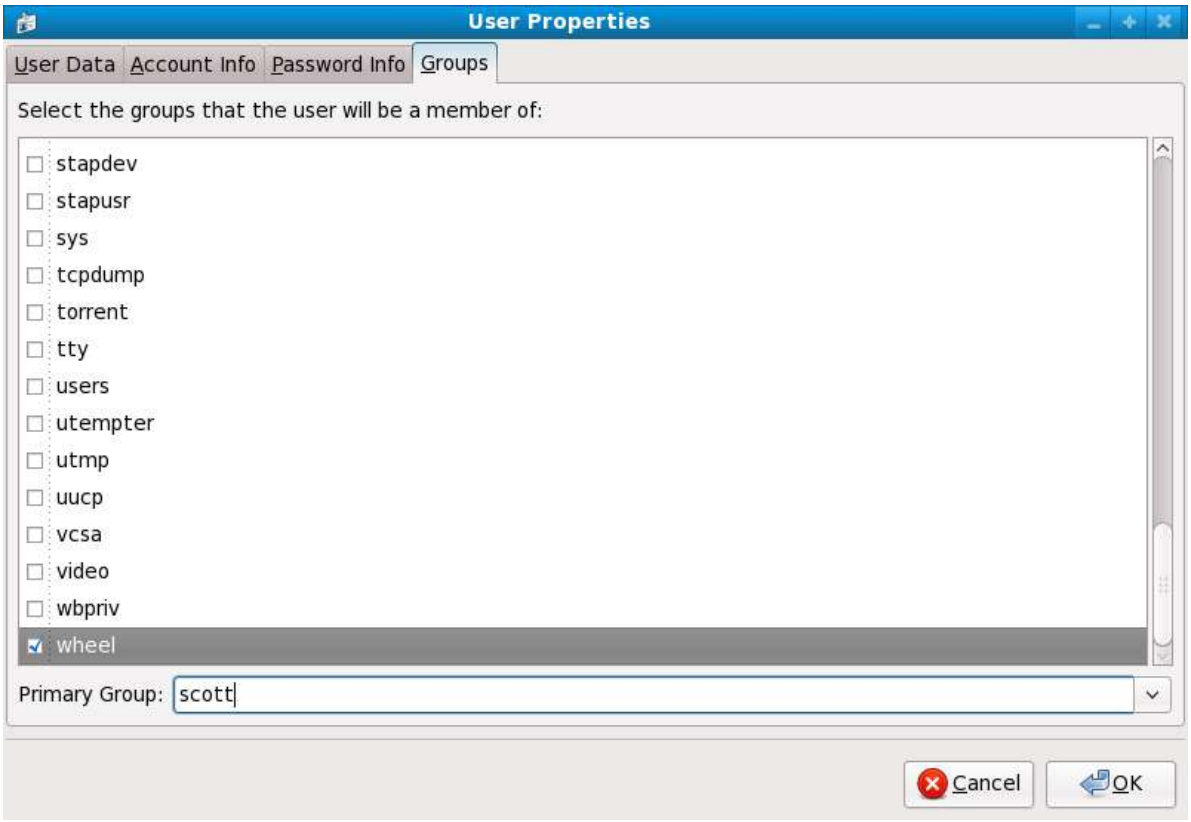

図3.2 ユーザーを "wheel" グループに追加します。

注記

root ユーザーはデフォルトで `wheel` グループの一部です。

3.1.4.3.2. sudo コマンド

sudo コマンドは、ユーザーに管理アクセス権を与えるために他のアプローチを提供します。信頼されたユーザーが管理コマンドの前に sudo をつけるとき、#####パスワードに対するプロンプトが出されます。そして、認証され、コマンドが許可されると考えられるとき、管理コマンドは root ユーザーであるかのように実行されます。

sudo コマンドの基本的な形式は以下のとおりです:

```
sudo <command>
```

上の例において、*<command>* は、`mount` のように、通常 root ユーザーのために取ってあるコマンドで置き換えられます。

重要

sudo コマンドのユーザーは、sudoers が5分以内はパスワードを聞かれることなく、再びコマンドを使用することができるので、マシンから離れる前にさらに注意深くログアウトするべきです。この設定は設定ファイル /etc/sudoers 経由で変更できます。

sudo コマンドは高いレベルの柔軟性を許します。たとえば、/etc/sudoers 設定ファイルにリストされたユーザーのみが sudo コマンドを使用できます。また、コマンドは#####シェルで実行され、root シェルではありません。このことは、*#root ########* に示されるように、root シェルが完全に無効にできることを意味します。

sudo コマンドは完全な監査証跡も提供します。それぞれの成功の認証は /var/log/messages に記録されます。また、発行されたコマンドは発行したユーザー名とともに /var/log/secure ファイルに記録されます。

sudo コマンドの他の利点は、管理者が異なるユーザーに対してそのニーズに基づいて特定のコマンドにアクセスを許可できることです。

sudo 設定ファイル (/etc/sudoers) を編集したい管理者は、visudo コマンドを使用すべきです。

誰かに完全な管理特権を与えるために、visudo を入力して、ユーザー権限指定セクションに以下のような行を追加します:

```
juan ALL=(ALL) ALL
```

この例は、ユーザー juan がすべてのホストから sudo を使用でき、すべてのコマンドを実行できます。

以下の例は、sudo を設定するときに、可能な粒度を説明します:

```
%users localhost=/sbin/shutdown -h now
```

この例は、すべてのユーザーがコンソールから発行される限り /sbin/shutdown -h now コマンドを発行できます。

sudoers のマニュアル・ページに、このファイルのオプションの詳細なリストがあります。

3.1.5. 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・アクセスが組織内でシステム管理者に対して重要な問題である間、ネットワーク・サービスが有効であることを監視することは、Linux システムを管理して運用する誰かにとって最高の重要事項です。

Fedora の下で多くのサービスはネットワーク・サービスとして動作します。ネットワーク・サービスがマシンで実行されると、サーバー・アプリケーション (*daemon* と呼ばれます) が1つかそれより多いネットワーク・ポートをリッスンしています。これらのサービスはそれぞれ攻撃の潜在的な道として取り扱われるべきです。

3.1.5.1. サービスへのリスク

ネットワーク・サービスは Linux システムに対して多くのリスクをもたらす可能性があります。以下は主要な問題のいくつかのリストです:

- *######## (DoS: Denial of Service Attacks)* — リクエストを用いてサービスを溢れさせることにより、ログとリクエストへの応答を試すので、サービス妨害攻撃はシステムを使用不能にすることができます。

- ########## *(DDoS: Distributed Denial of Service Attack)* — リクエストでサービスを溢れさせ、使用不能にする、サービスに協調した攻撃を指示するために、複数の侵入されたマシン（しばしば数千かそれより多い数です）を使用する DoS 攻撃の一種。

- ########## — サーバーが、ウェブサーバーで一般的に実行しているように、サーバーサイドアクションを実行するためにスクリプトを使用しているなら、クラッカーは不適切に書かれたスクリプトを攻撃できます。これらのスクリプト脆弱性攻撃はバッファーオーバーフローの条件に導き、攻撃者がシステムにあるファイルを改ざんできるようにする可能性があります。

- ################ — 0番から1023番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しなければいけません。アプリケーションがエクスプロイト可能なバッファー・オーバーフローを持つならば、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へのアクセス権をけることができます。エクスプロイト可能なバッファー・オーバーフローが存在するので、クラッカーは脆弱性を持つシステムを識別するために自動化されたツールを使用します。そして、一度アクセス権を得ると、システムへのアクセス権を維持するために自動化された rootkit を使用します。

注記

バッファー・オーバーフロー脆弱性の脅威は、Fedora において *ExecShield* により軽減されます。これは、x86 互換のシングル・プロセッサーおよびマルチ・プロセッサーのカーネルによりサポートされる実行可能なメモリー分割および保護の技術です。ExecShield は、仮想メモリーを実行可能および実行不可能のセグメントに分割することにより、バッファー・オーバーフローのリスクを減らします。バッファー・オーバーフローのエクスプロイトから注入された悪意のあるコードのように、実行可能セグメントの外側で実行しようとするすべてのプログラム・コードは、セグメンテーション・フォールトを引き起こし、終了します。

Execshield は、AMD64 プラットフォームにおける *No eXecute* (NX) 技術と、Itanium および Intel® 64 システムにおける *eXecute Disable* (XD) 技術に対するサポートも含みます。これらの技術は、悪意のあるコードが実行可能コードの 4KB の粒度を持つ、仮想メモリーの実行可能部分で実行されるのを防ぐために、ExecShield とともに動作して、ステルス型のバッファー・オーバーフローのエクスプロイトから攻撃のリスクを減らします。

重要

ネットワーク上の攻撃にさらされることを制限するため、使用していないすべてのサービスをオフにするべきです。

3.1.5.2. サービスの識別と設定

セキュリティを向上させるために、Fedora とともにインストールされた多くのネットワーク・サービスはデフォルトでオフにされています。しかしながら、いくつかの注意すべき例外があります:

- `cupsd` — Fedora のデフォルトのプリント・サーバ。

- `lpd` — 代替のプリント・サーバ。

- `xinetd` — `gssftp` や `telnet` のような、従属するサーバの範囲へのコネクションを制御するスーパー・サーバです。

- sendmail — Sendmail *Mail Transport Agent* (MTA) はデフォルトで有効にされていますが、localhost からのコネクションのみをリッスンします。

- sshd — Telnet のセキュアな代替である OpenSSH サーバ。

これらのサービスを実行したままにしておくかどうかを決めるとき、一般的なセンスを使用するのが最もよく、注意が過ぎると誤ります。たとえば、プリンターが利用できなければ、cupsd を実行したままにしておきません。portmap に対しても同じことが当てはまります。NFSv3 ボリュームをマウントしていない、もしくは NIS (ypbind サービス) を使用していなければ、portmap は無効にすべきです。

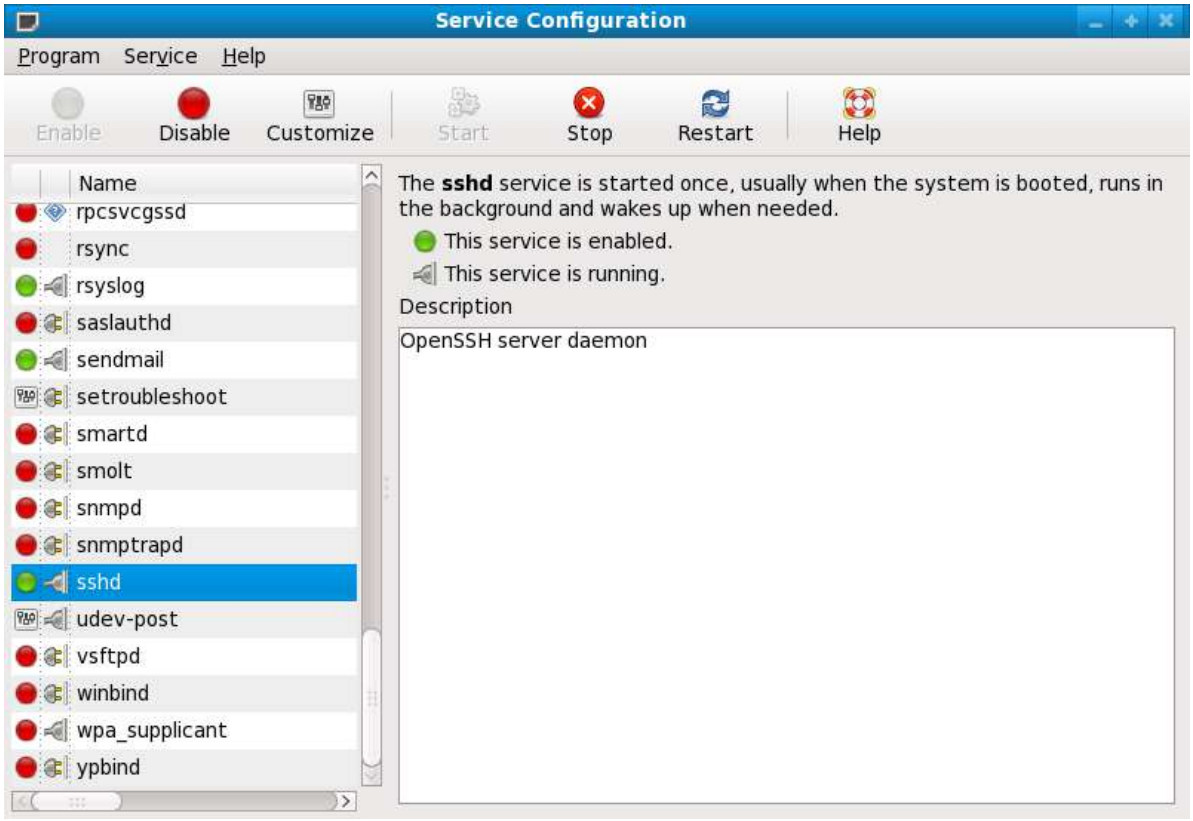

図3.3 サービス設定ツール

特定のサービスの目的が確かでなければ、サービス設定ツール が、*#3.3##########* に説明されている、追加の情報を提供する説明フィールドを持ちます。

ネットワーク・サービスがブート時に開始して利用可能かどうかを調べることは、話の一部です。どのポートが開いていて、リッスンしているかも調べるべきです。詳細は ################## を参照してください。

3.1.5.3. セキュアではないサービス

潜在的に、すべてのネットワーク・サービスはセキュアではありません。このため、使用していないサービスをオフにすることは非常に重要です。サービスに対するエクスプロイトは、定期的に公開およびパッチ提供がされ、すべてのネットワーク・サービスに関連するパッケージを定期的にアップデートすることは非常に重要になります。詳細は##################を参照してください。

いくつかのネットワーク・プロトコルは他のものよりも本質的によりセキュアではありません。これらは以下のようなあらゆるサービスを含みます:

- ################################ — Telnet や FTP のような古いプロトコルの多くは、認証セッションを暗号化せず、可能なときはいつでも避けられるべきです。

- ########################## — 多くのプロトコルは暗号化され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は Telnet, FTP, HTTP, および SMTP を含みます。NFS や SMB のような多くのネットワーク・ファイル・システムも暗号化されないネットワーク上で情報を転送します。これらのプロトコルを使用するとき、ユーザーのリポジトリはどの形式のデータが転送されるかを制限します。

  netdump のようなリモート・メモリー・ダンプ・サービスは、暗号化されないネットワーク上でメモリーの内容を転送します。メモリー・ダンプはパスワード、悪ければデータベース・エントリーや他の機密情報を含む可能性があります。

  finger や rwhod のような他のサービスは、システムのユーザーに関する情報を明らかにします。

比較的セキュアではない例として rlogin, rsh, telnet, および vsftpd があります。

すべてのリモートログインとシェルプログラムは (rlogin, rsh, および telnet) は、SSH を選んで、避けるべきです。sshd の詳細は############################を参照してください。

FTP はシステムのセキュリティに関してリモート・シェルほど本質的に危険ではありません。しかし、FTP サーバーは問題を避けるために注意深く設定され、監視されなければいけません。FTP サーバーをセキュアにすることに関する詳細は *#FTP #######* を参照してください。

注意深く導入され、ファイアウォールの後ろに置かれるべきサービスは以下です。

- finger
- authd (これは以前の Fedora リリースにおいて identd と呼ばれていました。)
- netdump
- netdump-server
- nfs
- rwhod
- sendmail
- smb (Samba)
- yppasswdd
- ypserv
- ypxfrd

ネットワーク・サービスをセキュアにすることに関する詳細は#############を参照してください。

次のセクションは簡単なファイアウォールをセットアップするために利用可能なツールについて議論します。

3.1.6. パーソナル・ファイアウォール

###ネットワーク・サービスが設定した後、ファイアウォールを導入することは重要です。

重要

インターネットやあなたが信頼できない他のあらゆるネットワークに接続する##、必要なサービスを設定し、ファイアウォールを導入すべきです。

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。リクエストがファイアウォールによりブロックされたポート宛てならば、パケットを受け取らず、効果的に無効化されます。この理由により、使っていないポートへのアクセスをブロックする一方、設定されたサービスにより使われるポートへのアクセスをブロックしないようにするためにファイアウォールを設定するとき注意すべきです。

多くのユーザーにとって、シンプルなファイアウォールを設定するための最も良いツールは、Fedora に同梱されているグラフィカルなファイアウォール設定ツールです: ファイアウォール管理ツール (`system-config-firewall`)。このツールはコントロール・パネル・インターフェースを用いて一般的な目的のファイアウォールに対する幅広い `iptables` ルールを作成します。

高度なユーザーおよびサーバー管理者に対して、`iptables` を用いてファイアウォールを手動で設定することは、おそらくより良いオプションです。詳細は*#Using Firewalls#*を参照してください。

3.1.7. セキュリティ強化したコミュニケーション・ツール

インターネットの規模と人気が拡大するにつれて、コミュニケーションの盗聴の脅威があります。何年にもわたり、それらがネットワーク上で転送されるので、ツールは暗号化されたコミュニケーションのために開発されてきました。

Fedora は、情報がネットワーク上で送られるので、それを保護するために高いレベルの公開鍵暗号ベースの暗号化アルゴリズムを使用する基本的なツールを2つ同梱しています。

- *OpenSSH* — ネットワーク通信を暗号化するための SSH プロトコルのフリー実装。
- *Gnu Privacy Guard (GPG)* — データを暗号化するための暗号アプリケーション PGP (Pretty Good Privacy) のフリー実装。

OpenSSH は、リモートマシンにアクセスするより安全な方法で、`telnet` や `rsh` のようなより古い暗号化されないサービスを置き換えます。OpenSSH は `sshd` というネットワーク・サービスおよび3つのコマンドライン・クライアント・アプリケーションを含みます:

- `ssh` — リモート・コンソールのセキュアなアクセス・クライアント
- `scp` — セキュアなリモート・コピーのコマンド
- `sftp` — インタラクティブなファイル転送セッションを可能にする、セキュアな擬似 ftp クライアント

OpenSSHに関する詳細は*#Secure Shell#*を参照してください。

重要

sshd サービスは本質的にセキュアであるにも関わらず、サービスはセキュリティの脅威を防ぐために常に最新にしておかなければ#####。詳細は ################ を参照してください。

GPG はプライベートな email コミュニケーションを確実にする1つの方法です。パブリック・ネットワーク上で秘密データを email するためや、ハードディスクにある秘密データを保護するためのどちらにも使用されます。

3.2. サーバのセキュリティ

システムがパブリック・ネットワークにおいてサーバとして使用されるとき、攻撃の対象になります。そのため、システムを堅牢化して、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最も重要なことになります。

特定の問題を掘り下げて考える前に、サーバのセキュリティを強化するための一般的な以下のヒントについて再考します:

- 最新の脅威に対して保護するために、すべてのサービスを最新に保ちます。
- できる限りセキュアなプロトコルを使用します。
- できる限りマシンあたり1種類のネットワークサービスのみを取り扱います。
- 疑わしい活動に対してすべてのサーバを注意深く監視します。

3.2.1. TCP Wrappers と xinetd を用いたサービスのセキュア化

TCP Wrappers はさまざまなサービスにアクセス制御を提供します。SSH、Telnet および FTP のような最近のネットワークサービスの多くは TCP Wrappers（入ってくるリクエストと要求されたサービスの間で見張りをします）を使用します。

TCP Wrappers により提供される利便性は、`xinetd`（追加のアクセス、ロギング、バインド、リダイレクトおよびリソース活用に関する制御を提供するスーパーサービス）と併用するときに向上します。

注記

サービスのアクセス制御の中に冗長性を持たせるために、TCP Wrappers と `xinetd` とともに iptables ファイアウォール・ルールを使用することは素晴らしいアイディアです。iptables コマンドを用いたファイアウォールの実装に関する詳細は*#Using Firewalls#*を参照してください。

以下のサブセクションは、各トピックに関する基本的な知識があることを想定し、特定のセキュリティ・オプションに焦点を合わせています。

3.2.1.1. TCP Wrappres を用いたセキュリティの強化

TCP Wrappers はサービスへのアクセスを拒否する以外にも多くの機能があります。このセクションは、接続バナーを送信し、特定のホストからの攻撃者に警告をし、ログ機能を強化するために、どのように使うことができるかを説明します。TCP Wrapper 機能と制御言語に関する詳細は、`hosts_options` man page を参照してください。

3.2.1.1.1. TCP Wrappers と接続バナー

ユーザーがサービスに接続するときに適切なバナーを表示することは、潜在的な攻撃者へとシステム管理者が気を配っていることを知らせるために有用な方法です。システムに関するどのような情報がユーザーへと表示されるかを制御することもできます。サービスに対して TCP Wrappers バナーを導入するために、banner オプションを使用します。

この例は vsftpd のバナーを導入しています。始めるにはバナーファイルを作成します。それはシステムのどこでも構いませんが、デーモンと同じ名前でなければいけません。たとえば、そのファイルは /etc/banners/vsftpd と呼ばれ、以下の行を含みます:

```
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、より接続をおじけずかせるように、ユーザー名とホスト名、または、ユーザー名と IP アドレスのような、クライアントのさまざまな情報を提供します。

受信コネクションに表示するためのこのバナーに対して、/etc/hosts.allow ファイルに以下の行を追加します:

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

3.2.1.1.2. TCP Wrappers と攻撃の警告

特定のホストやネットワークがサーバを攻撃していることを検知したら、TCP Wrappers は spawn ディレクティブを用いて、そのホストまたはネットワークからの後続の攻撃について管理者に警告するために使用されます。

この例では 206.182.68.0/24 ネットワークからのクラッカーがサーバを攻撃しようとしていることを検知したと仮定しています。そのネットワークからの接続試行をすべて拒否して、その試行を特別なファイルに記録するために、/etc/hosts.deny ファイルに以下の行を置きます:

```
ALL : 206.182.68.0 : spawn /bin/ '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たサービスの名前を提供します。

接続を許可して、それを記録するには、/etc/hosts.allow ファイルに spawn ディレクティブを置きます。

注記

spawn ディレクティブはあらゆるシェルコマンドを実行するので、特定のクライアントがサーバへ接続しようとしたときに、管理者に通知したり、一連のコマンドを実行したりする特別なスクリプトを作成することは素晴らしいアイディアです。

3.2.1.1.3. TCP Wrappers と高度な

もし特定の種類の接続が他のものよりも注意する必要があれば、severity オプションを用いて、ログレベルをそのサービスに対して上昇させることができます。

この例では、FTP サーバのポート23番（Telnet ポート）に接続しようとする者はすべて攻撃者であると仮定しています。このことを示すために、ログファイルにおいてデフォルトのフラグ info の代わりに emerg フラグを立てます。そして、接続を拒否します。

これを実行するには、/etc/hosts.deny に以下の行を置きます:

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

これはデフォルトの authpriv ログ・ファシリティを使用しますが、プライオリティをデフォルト値の¥n info から emerg (ログメッセージを直接コンソールに送ります) へと上昇させます。

3.2.1.2. xinetd を用いた高度なセキュリティ

このセクションは、トラップ・サービスを設定するために xinetd を使用すること、および与えられたすべての xinetd サービスが利用可能になるリソース・レベルを制御するために使用することに焦点を当てます。サービスに対するリソース制限を設定することで、*Denial of Service* (DoS) 攻撃を阻止する助けにできます。利用可能なオプションの一覧は、xinetd と xinetd.conf のマニュアルページを参照してください。

3.2.1.2.1. トラップの設定

xinetd の重要な機能の1つは、全体に影響する no_access リストにホストを追加する機能です。このリストにあるホストは、指定された期間または xinetd が再起動されるまで xinetd により管理されたサービスへの後続の接続が拒否されます。

SENSOR をセットアップする最初のステップは、使用しない予定のサービスを選択することです。この例では、Telnet が使われます。

/etc/xinetd.d/telnet ファイルを編集して、読み込むために flags 行を変更します:

```
flags           = SENSOR
```

以下の行を追加します:

```
deny_time       = 30
```

これにより、そのホストによるそのポートへのさらなる接続試行は30分間拒否されます。deny_time 属性に対する他の利用可能な値は FOREVER (xinetd が再起動されるまで禁止効果が続きます) および NEVER (接続を許可して記録します)です。

最後に、最終行に次を読み込むべきです:

```
disable         = no
```

これはトラップ自身を有効にします。

SENSOR を使用することは望ましくないホストからの接続を検知して停止するための素晴らしい方法ですが、欠点が2つあります:

- ステルス・スキャンに対してうまく機能しません。
- SENSOR を実行していることを知っている攻撃者は、IP アドレスを偽造して、禁止されたポートに接続することにより、特定のホストに対するサービス妨害攻撃をしかけることができます。

3.2.1.2.2. サーバ・リソースの制御

xinetd の他の重要な機能は、制御下にあるサービスに対してリソース制限を設定する能力です。

以下のディレクティブを用いて実施します:

- cps = <number_of_connections> <wait_period> — 受信コネクションの割合を制限します。このディレクティブは2つの引数をとります。

  - <number_of_connections> — 1秒あたりに処理するコネクションの数。受信コネクションの割合がこれよりも多くなると、サービスが一時的に無効にされます。デフォルト値は50です。
  - <wait_period> — サービスが無効化された後、再び有効化されるまでの待ち時間（秒単位）。デフォルトの間隔は10秒です。
- instances = <number_of_connections> — サービスへの許可されるコネクションの合計数を指定します。このディレクティブは、整数値もしくは UNLIMITED をとります。
- per_source = <number_of_connections> — 各ホストあたりのサービスへの許可される接続数を指定します。このディレクティブは、整数値もしくは UNLIMITED をとります。
- rlimit_as = <number[K|M]> — サービスが占有できるメモリアドレス空間の量をキロバイトまたはメガバイト単位で指定します。このディレクティブは、整数値もしくは UNLIMITED をとります。
- rlimit_cpu = <number_of_seconds> — サービスが CPU を占有できる合計時間を秒単位で指定します。このディレクティブは、整数値もしくは UNLIMITED をとります。

これらのディレクティブを使用すると、ある1つの xinetd サービスがシステムを制圧して、サービス妨害を達成することを防ぐ助けにできます。

3.2.2. Portmap のセキュア化

portmap サービスは、NIS や NFS のような RPC サービスに対して、動的にポートを割り当てるデーモンです。認証メカニズムは弱いです。また、制御しているサービスに対して広い範囲のポートを割り当てる機能があります。これらの理由により、セキュアにすることが難しいです。

注記

NFSv4 はもはやそれを必要としないので、portmap をセキュアにすることは NFSv2 と NFSv3 の導入に対してのみ効果があります。もし NFSv2 または NFSv3 サーバを導入しようとしているならば、portmap が必要となり、以下のセクションが適用されます。

もし RPC サービスを実行しているならば、以下の基本的なルールに従います。

3.2.2.1. TCP Wrappers を用いた portmap の保護

組み込み形式の認証を持たないので、portmap サービスへアクセスするネットワークまたはホストを制限するために、TCP Wrappers を使用することは重要です。

さらに、サービスへのアクセスを制限するとき、ホスト名を使うのを避け、IP アドレス##を使います。その理由は、ホスト名は DNS ポイズニングや他の方法により偽造される可能性があるからです。

3.2.2.2. iptables を用いた portmap の保護

portmap サービスへのアクセスをさらに制限するために、サーバに iptables ルールを追加して、特定のネットワークへのアクセスを制限することは、素晴らしいアイディアです。

以下は iptables コマンドの2つの例です。1つ目は、192.168.0.0/24 のネットワークから、ポート 111（portmap サービスにより使用されます）への TCP 接続を許可します。2つ目は、ローカルホストから同じポートへのアクセスを許可します。これは、Nautilus により使用される sgi_fam サービスのために必要となります。他のパケットはすべて破棄されます。

```
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
```

UDP トラフィックを同じように制限するために、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

ファイアウォールと iptables コマンドの実装に関する詳細は*#Using Firewalls#*を参照してください。

3.2.3. NIS のセキュア化

Network Information Service (NIS) は ypserv と呼ばれる RPC サービスです。これは、ドメイン内にあることを主張しているすべてのコンピュータへと、ユーザ名、パスワードおよび他の機密情報の対応付けを配布するために、portmap や他の関連するサービスとともに使用されます。

NIS サーバはいくつかのアプリケーションを包含しています。それらは以下を含みます:

- /usr/sbin/rpc.yppasswdd — yppasswdd サービスとも呼ばれ、このデーモンはユーザーが NIS パスワードを変更できるようにします。
- /usr/sbin/rpc.ypxfrd — ypxfrd サービスとも呼ばれ、このデーモンはネットワークにおいて NIS マップを送信する責任があります。
- /usr/sbin/yppush — このアプリケーションは変更された NIS データベースを複数の NIS サーバへ伝搬します。
- /usr/sbin/ypserv — これは NIS サーバのデーモンです。

NIS は今日の標準によるといくらかセキュアではありません。ホスト認証メカニズムを持たず、暗号化されていないネットワーク上ですべての情報を転送します。結果として、NIS を使用するネットワークをセットアップするとき、極めて注意しなければいけません。NIS のデフォルト設定は本質的にセキュアではないという事実により、さらに複雑になります。

NIS サーバを導入しようとしている人は、まず *#Portmap ####### に示されているように portmap サービスをセキュアにして、その後ネットワーク計画のような以下の問題に取り組むことが推奨されます。

3.2.3.1. ネットワークの注意深い計画

NIS はネットワーク上で暗号化せずに秘密情報を転送するので、ファイアウォールの内側で、セグメント化されたセキュアなネットワークにおいてサービスを実行することが重要です。NIS 情報はセキュアではないネットワーク上で転送されるときは必ず、傍受されるリスクがあります。慎重なネットワーク設計が深刻なセキュリティ侵害を防ぐ助けにできます。

3.2.3.2. パスワードのような NIS ドメイン名とホスト名の使用

NIS ドメインの中にあるすべてのマシンは、ユーザーが NIS サーバの DNS ホスト名を NIS ドメイン名を知っている限り、認証なしでサーバーから情報を抽出するためのコマンドを使用できます。

たとえば、誰かがネットワークの中にあるノートPCに接続する、もしくは、外部からネットワーク内に侵入する(かつ内部 IP アドレスを詐称するよう管理する)と、以下のコマンドで /etc/passwd マップを暴露します:

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

攻撃者が root ユーザーならば、以下のコマンドを入力することにより /etc/shadow ファイルを手に入れることができます:

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

注記

Kerberos を使用していると、/etc/shadow ファイルは NIS マップの中には保存されません。

攻撃者に対して NIS マップへのアクセスを堅牢化するために、o7hfawtgmhwg.domain.com のようなランダムな文字列を DNS ホスト名のために作成します。同様に、###ランダムな NIS ドメイン名を作成します。これにより、攻撃者が NIS サーバへアクセスすることがより困難になります。

3.2.3.3. /var/yp/securenets ファイルの編集

/var/yp/securenets ファイルが空白または存在しなければ(デフォルト・インストール直後の場合)、NIS はすべてのネットワークを受け付けます。最初にすることは、ypserv が適切なネットワークからのリクエストのみに応答するよう、ネットマスク/ネットワーク ペアを置くことです。

以下は /var/yp/securenets ファイルからのサンプル・エントリです:

```
255.255.255.0     192.168.0.0
```

警告

初めてのとき /var/yp/securenets ファイルを作成せずに NIS サーバを決して起動しないでください。

このテクニックは IP 詐称攻撃からの保護を提供しませんが、少なくとも NIS サーバのサービスがどのネットワークにあるかを制限します。

3.2.3.4. 静的ポートの割り当てと iptables ルールの使用

NIS に関連するすべてのサーバは、rpc.yppasswdd — 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、を除いて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 サーバデーモンrpc.ypxfrd と ypserv にポートを割り当てることにより、侵入者から NIS サーバデーモンをさらに保護するためにファイアウォール・ルールを作成できます。

これをするために、/etc/sysconfig/network に以下の行を追加します:

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

そして、以下の iptables ルールは、これらのポートに対してサーバがどのネットワークを待ち受けているかを強制するために使われます。

```
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
```

このことは、リクエストが 192.168.0.0/24 のネットワークからならば、プロトコルに関係なく、ポート834と835への接続だけが許可されることを意味します。

注記

ファイアウォールと iptables コマンドの実装に関する詳細は*#Using Firewalls#*を参照してください。

3.2.3.5. Kerberos 認証の使用

NIS を認証用に使用するときに検討する問題の1つは、ユーザーがマシンにログインするときは必ず、/etc/shadow マップからのパスワード・ハッシュがネットワーク上で送られることです。侵入者が NIS ドメインへのアクセスを獲得して、ネットワークのトラフィックを盗聴すると、ユーザー名とパスワード・ハッシュを収集することができます。十分な時間があれば、パスワード解析プログラムは弱いパスワードを推測でき、攻撃者はネットワークにおいて有効なアカウントへのアクセス権を得ることができます。

Kerberos は秘密鍵暗号を使用するので、パスワード・ハッシュがネットワーク上に送られず、システムをよりもっとセキュアにします。Kerberos の詳細は *#Kerberos#* を参照ください。

3.2.4. NFS のセキュア化

重要

Fedora に含まれるバージョンのNFS (NFSv4) は、*#Portmap #######* に概要が示されているように portmap サービスをもはや必要としません。NFS トラフィックはすべてのバージョンにおいて UDP より TCP を使用します。NFSv4 を使用するときそれを必要とします。NFSv4 は、RPCSEC_GSS カーネルモジュールの一部として、Kerberos ユーザーとグループの認証を含みます。Fedora が NFSv2 と NFSv3 をサポートするので (どちらも portmap を利用します)、portmap の情報はまだ含まれています。

3.2.4.1. ネットワークの注意深い計画

いまや NFSv4 はネットワーク上で Kerberos を用いて暗号化されたすべての情報を受け渡す機能があるので、ファイアウォールの後ろ側もしくはセグメント化されたネットワーク上にあるならば、サービスが正しく設定されることが重要です。NFSv2 と NFSv3 はまだ安全ではなくデータを受け渡します。このことは考慮に入れられるべきです。これらの観点すべてにおいてネットワークを慎重に設計することは、セキュリティ侵害を防ぐ助けにできます。

3.2.4.2. 構文エラーへの注意

NFS サーバは、/etc/exports ファイルを参照することにより、どのファイルシステムをエクスポートするか、どのホストへとこれらのディレクトリをエクスポートするかを決めます。このファイルを編集するときに、無関係な空白を追加しないよう注意してください。

たとえば、/etc/exports ファイルにある以下の行は、ディレクトリ /tmp/nfs/ を bob.example.com へと読み書き権付きで共有します。

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

一方で /etc/exports ファイルにある以下の行は、同じディレクトリを bob.example.com へと読み込み権のみ付きで共有します。また、ホスト名の後ろにある1つの空白により、それを読み書き権付きで##に共有します。

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

何が共有されているかを確認するために、showmount コマンドを用いることにより、設定された NFS 共有すべてを確認することはグッド・プラクティスです。

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

3.2.4.3. no_root_squash オプションの未使用

デフォルトで NFS 共有は root ユーザーを nfsnobody ユーザー（非特権ユーザーアカウント）に変更します。これにより root が作成したファイルの所有者はすべて nfsnobody に変更されます。ここで、setuid ビットが設定されたプログラムのアップロードは防がれます。

no_root_squash が使われていると、リモートの root ユーザーが、共有ファイルシステムにあるすべてのファイルを変更でき、他のユーザが不注意で実行するようトロイの木馬により感染されたアプリケーションを置いていけます。

3.2.4.4. NFS ファイアウォールの設定

NFS のため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。それは、ファイアウォール・ルールを作成するときに問題を引き起こす可能性があります。このプロセスを単純化するため、どのポートが使われるかを指定するために */etc/sysconfig/nfs* ファイルを使用します。

- MOUNTD_PORT — mountd (rpc.mountd) 用の TCP および UDP ポート
- STATD_PORT — status (rpc.statd) 用の TCP および UDP ポート
- LOCKD_TCPPORT — nlockmgr (rpc.lockd) 用の TCP ポート
- LOCKD_UDPPORT — nlockmgr (rpc.lockd) 用の UDP ポート

指定されたポート番号はすべての他のサービスにより使用されてはいけません。TCP および UDP ポート 2049 (NFS) と同様、指定されたポート番号を許可するようファイアウォールを設定します。

どのポートと RPC プログラムが使われているかを確認するために、NFS サーバにおいて rpcinfo -p コマンドを実行します。

3.2.5. Apache HTTP Server のセキュア化

Apache HTTP Server は、Fedora に同梱されている、最も安定していてセキュアなサービスの1つです。多くのオプションとテクニックが Apache HTTP Server をセキュアにするために利用できます — ここで深く調べるには多すぎます。以下のセクションは Apache HTTP Server を実行するときのベストプラクティスを簡単に説明します。

システムで実行するスクリプトは本番環境に置く##意図したとおりに動作することを常に確認します。また、root ユーザーのみが、スクリプトや CGI を含むすべてのディレクトリに対する書き込み権限を持つことを確認します。

1. ```
   chown root <directory_name>
   ```
2. ```
   chmod 755 <directory_name>
   ```

システム管理者が以下の設定オプションを使用するときは注意が必要です (/etc/httpd/conf/httpd.conf において設定されます):

FollowSymLinks
: このディレクティブはデフォルトで有効です。そのため、ウェブサーバのドキュメントルートにシンボリックリンクを作成するときは確実に注意します。たとえば、/ へのシンボリックリンクを提供することは悪いアイディアです。

Indexes
: このディレクティブはデフォルトで有効です。しかし、望ましくありません。訪問者がサーバーにあるファイルを探索するのを防ぐため、このディレクティブを削除します。

UserDir
: システムのユーザーアカウントの存在を確認できるので、UserDir ディレクティブはデフォルトで無効です。サーバーのユーザーディレクトリのブラウジングを有効にするために、以下のディレクティブを使います:

```
UserDir enabled
UserDir disabled root
```

これらのディレクティブは /root/ 以外のすべてのユーザーディレクトリに対するユーザー・ディレクトリのブラウジングを有効にします。無効に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 行にスペース区切りでユーザーの一覧を追加します。

重要

IncludesNoExec ディレクティブを削除しないでください。*Server-Side Includes* (SSI) モジュールはデフォルトでコマンドを実行できません。潜在的に、攻撃者がシステムにあるコマンドを実行できるようにできるので、絶対に必要にならない限り、この設定を変更しないことを推奨します。

3.2.6. FTP のセキュア化

File Transfer Protocol (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。サーバとのすべてのトランザクション(ユーザー認証を含みます)が暗号化されないので、セキュアではないプロトコルと考えられていて、慎重に設定されるべきです。

Fedora は3つの FTP サーバを提供します。

- gssftpd — ネットワーク上で認証情報を転送しない、Kerberos 対応の xinetd ベースの FTP デーモン
- Red Hat Content Accelerator (tux) — FTP 機能を持つカーネル空間のウェブサーバ
- vsftpd — スタンドアロンの、セキュリティ志向で実装された FTP サービス

以下のセキュリティ・ガイドラインは vsftpd FTP サービスをセットアップするためのものです。

3.2.6.1. FTP グリーティング・バナー

ユーザー名とパスワードを送信する前に、すべてのユーザーはグリーティング・バナーが表示されます。デフォルトで、このバナーはクラッカーがシステムにある弱点を識別するために有効なバージョン情報を含みます。

vsftpd に対するグリーティング・バナーを変更するには、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>
```

上のディレクティブにある *<insert_greeting_here>* をグリーティング・メッセージのテキストで置き換えます。

複数行のバナーには、バナー・ファイルを使用することが最も良いです。複数のバナーの管理を簡単にするために、/etc/banners/ という新しいディレクトリにすべてのバナーを置きます。この例における FTP 接続に対するバナー・ファイルは /etc/banners/ftp.msg です。以下はファイルがどのように見えるかの例です:

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

注記

*#TCP Wrappers #######*で具体化されているように、ファイルの各行を 220 で始めることは必要ありません。

vsftpd に対するこのグリーティング・バナーを参照するには、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
```

*#TCP Wrappers #######*に記載されているように、TCP Wrappers を使用して入ってくる接続へと追加のメッセージを送ることが可能です。

3.2.6.2. 匿名アクセス

/var/ftp/ ディレクトリの存在により匿名アカウントが有効化されます。

このディレクトリを作成するもっとも簡単な方法は 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、匿名ユーザーに対するディレクトリツリーを確立し、匿名ユーザーに対して読み込み専用のパーミッションをそのディレクトリに設定します。

デフォルトで匿名ユーザーはあらゆるディレクトリに書き込みできません。

警告

FTP サーバへの匿名アクセスを有効にすると、機密データが保存されている場所に注意してください。

3.2.6.2.1. 匿名アップロード

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ため、書き込み専用ディレクトリを /var/ftp/pub/ の中に作成することを推奨します。

これをするために、以下のコマンドを入力します:

```
mkdir /var/ftp/pub/upload
```

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

```
chmod 730 /var/ftp/pub/upload
```

ディレクトリの long フォーマットの一覧はこのように見えるでしょう:

```
drwx-wx---    2 root     ftp          4096 Feb 13 20:05 upload
```

警告

匿名ユーザーがディレクトリにおいて読み書きすることを許可する管理者は、しばしばそれらのサーバーが盗難されたソフトウェアの保管庫になっていることを見つけます。

加えて、vsftpd の下で、以下の行を /etc/vsftpd/vsftpd.conf ファイルに追加します:

```
anon_upload_enable=YES
```

3.2.6.3. ユーザー・アカウント

FTP は認証のためにセキュアではないネットワーク上に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ので、それらのユーザー・アカウントからサーバーへのアクセスを拒否することは素晴らしいアイディアです。

vsftpd においてすべてのユーザー・アカウントを無効にするため、以下のディレクティブを /etc/vsftpd/vsftpd.conf に追加します:

```
local_enable=NO
```

3.2.6.3.1. ユーザー・アカウントの制限

root ユーザーや sudo 特権を持つユーザーのような、特定のアカウントもしくはアカウントの特定のグループを無効にするために、最も簡単な方法は*#PAM #### root #####*に記載されている PAM リスト・ファイルを使用することです。vsftpd 用の PAM 設定ファイルは /etc/pam.d/vsftpd です。

各サービスにおいてユーザーアカウントを直接無効化することもできます。

vsftpd において特定のアカウントを無効にするには、ユーザー名を /etc/vsftpd.ftpusers に追加します

3.2.6.4. アクセス制御のための TCP Wrappers の使用

#TCP Wrappres ############### に概要が示されているように、FTP デーモンへのアクセスを制御するために TCP Wrappers を使用します。

3.2.7. Sendmail のセキュア化

Sendmail は、他の MTA と email クライアントや配送エージェントとの間で電子メッセージを配送するために Simple Mail Transfer Protocol (SMTP) を使用する Mail Transfer Agent (MTA) です。多くの MTA はもう一方との間のトラフィックを暗号化する機能がありますが、多くはそうしないので、あらゆるパブリック・ネットワーク上で email を送信することは、本質的にセキュアではないコミュニケーションの形式であると考えられています。

Sendmail サーバを導入しようしている人は以下の問題に取り組むことが推奨されます。

3.2.7.1. サービス妨害攻撃の制限

email の特性のため、本気になった攻撃者は、極めて簡単にメールを用いてサーバをあふれさせ、サービス妨害を引き起こすことができます。/etc/mail/sendmail.mc において以下のディレクティブに制限を設定することにより、そのような攻撃者を制限する効果があります。

- confCONNECTION_RATE_THROTTLE — サーバが1秒当たりに受け付ける接続数。デフォルトで、Sendmail は接続数を制限しません。制限が設定され、制限に達すると、さらなる接続は遅延させられます。
- confMAX_DAEMON_CHILDREN — サーバにより生成される子プロセスの最大数。デフォルトで、Sendmail は子プロセスの数の制限を割り当てません。制限が設定され、制限に達すると、さらなる接続は遅延させられます。
- confMIN_FREE_BLOCKS — メールを受け付けるためにサーバが利用可能でなければいけない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・ヘッダの受信可能な最大容量（バイト単位）。
- confMAX_MESSAGE_SIZE — 1つのメッセージの受信可能な最大容量（バイト単位）。

3.2.7.2. NFS と Sendmail

メールスプールのディレクトリ /var/spool/mail/ を NFS 共有ボリュームに置いてはいけません。

NFSv2 と NFSv3 はユーザーとグループの ID で制御されないので、2人以上のユーザーが同じ UID を持ち、それぞれ他のメールを受け取り、読む可能性があります。

注記

Kerberos を用いる NFSv4 を使用していると、SECRPC_GSS カーネル・モジュールは UID ベースの認証を利用しないので、これは該当しません。

3.2.7.3. メール専用ユーザー

Sendmail サーバにおいてローカルユーザーがエクスプロイトするのを防ぐ助けにするため、メールのユーザーは email プログラムを用いて Sendmail サーバのみにアクセスすることが最善です。メールサーバにおけるシェル・アカウントは許可されるべきではなく、/etc/passwd におけるユーザー・シェルはすべて /sbin/nologin に設定されるべきです（root ユーザーを除いて）。

3.2.8. リッスンしているポートの確認

ネットワーク・サービスを設定した後、システムのネットワーク・インタフェースにおいて実際にどのポートがリッスンしているかに注意することは重要です。すべての開いているポートは侵入の兆候になる可能性があります。

ネットワークにおいて待ち受けているポートを一覧化するための基本的なアプローチが2つあります。より信頼できないアプローチは、netstat -an や lsof -i のようなコマンドを用いてネットワーク・スタックを問い合わせることです。これらのプログラムはネットワークからマシンへ接続しないので、この方法はより信頼できませんが、システムにおいて実行しているものをよりチェックできます。そのため、これらのアプリケーションは頻繁に攻撃者により置き換えられる対象になります。攻撃者が認可されていないネットワーク・ポートを開くならば、netstat と lsof を自分自身の改変したバージョンに置き換えることにより、その痕跡を隠そうとします。

ネットワークにおいてどのポートがリッスンしているかを確認するためのより信頼できる方法は、nmap のようなポート・スキャナーを使用することです。

コンソールから実行される以下のコマンドは、どのポートがネットワークからの TCP 接続を待ち受けているかを決めます:

```
nmap -sT -O localhost
```

このコマンドの出力は以下のように表示されます:

```
Starting Nmap 4.68 ( http://nmap.org ) at 2009-03-06 12:08 EST
Interesting ports on localhost.localdomain (127.0.0.1):
Not shown: 1711 closed ports
PORT      STATE SERVICE
22/tcp    open  ssh
25/tcp    open  smtp
111/tcp   open  rpcbind
113/tcp   open  auth
631/tcp   open  ipp
834/tcp   open  unknown
2601/tcp  open  zebra
32774/tcp open  sometimes-rpc11
Device type: general purpose
Running: Linux 2.6.X
OS details: Linux 2.6.17 - 2.6.24
Uptime: 4.122 days (since Mon Mar  2 09:12:31 2009)
Network Distance: 0 hops
OS detection performed. Please report any incorrect results at http://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 1.420 seconds
```

この出力は、sunrpc の存在により、システムが portmap を実行していることを示しています。しかしながら、ポート834に謎のサービスがあります。そのポートが既知のサービスの公式な一覧に関連づけられるかを確認するため、次を入力します:

```
cat /etc/services | grep 834
```

このコマンドは何も出力しません。このことは、ポートが予約済み範囲（0から1023を意味します）にあり、開くために root アクセスが必要であり、既知のサービスに関連づけられていないことを意味します。

次に、netstat または lsof を用いてポートに関する情報を確認します。netstat を用いてポート834を確認するために、以下のコマンドを使用します:

```
netstat -anp | grep 834
```

コマンドは以下の出力を返します:

```
tcp   0    0 0.0.0.0:834    0.0.0.0:*   LISTEN   653/ypbind
```

攻撃者が侵入したホストにおいて密かに開けたポートが、このコマンドにより明らかにされないかもしれないので、netstat で開いているポートの存在を再確認します。また、[p] オプションは、ポートを開いているサービスのプロセス ID (PID) を明らかにします。この場合、開いているポートは ypbind (NIS) に属しています。これは、portmap サービスとともに取り扱われる RPC サービスです。

lsof コマンドは、開いているポートをサービスと対応される機能もあるので、netstat と同じような情報を明らかにします。

```
lsof -i | grep 834
```

このコマンドからの出力の関連する部分は次のようです:

```
ypbind      653        0    7u  IPv4       1319                TCP *:834 (LISTEN)
```

```
ypbind      655        0    7u  IPv4       1319                TCP *:834 (LISTEN)
ypbind      656        0    7u  IPv4       1319                TCP *:834 (LISTEN)
ypbind      657        0    7u  IPv4       1319                TCP *:834 (LISTEN)
```

これらのツールは、マシンで実行しているサービスの状態に関する詳細を非常に明らかにします。これらのツールは、柔軟であり、ネットワーク・サービスと設定に関する豊かな情報を提供します。詳細は `lsof`, `netstat`, `nmap`, および `services` のマニュアルページを参照してください。

3.3. Single Sign-on (SSO)

3.3.1. 概要

Fedora の SSO 機能は Fedora デスクトップのユーザーがパスワードを入力しなければいけない回数を減らします。いくつかの有名なアプリケーションは、ユーザーがログイン画面から Fedora にログインでき、パスワードを再入力する必要がないよう、同じ基礎となる認証と認可のメカニズムを導入します。これらのアプリケーションは以下で詳しく説明されます。

さらに、ネットワークがないとき（#########）やネットワーク接続性が信頼できないところ（たとえば、無線アクセス）でさえ、それらのメカニズムにログインすることができます。後者の場合、サービスは緩やかに機能を下げていきます。

3.3.1.1. サポートされるアプリケーション

以下のアプリケーションは Fedora における単一ログインのスキームを現在サポートしています:

- ログイン
- スクリーンセーバー
- Firefox および Thunderbird

3.3.1.2. サポートされる認証メカニズム

Fedora は以下の認証メカニズムを現在サポートしています:

- ケルベロス名/パスワードログイン
- スマートカード/PIN ログイン

3.3.1.3. サポートされるスマートカード

Fedora は Cyberflex e-gate カードとリーダを用いてテストされていますが、Java card 2.1.1 および Global Platform 2.0.1 仕様の両方を用いて組み立てられているすべてのカードは、すべてのリーダが PCSC-lite によりサポートされているので、正しく動作するでしょう。

Fedora は Common Access Cards (CAC) を用いてもテストされています。CAC 用にサポートされるリーダは SCM SCR 331 USB リーダです。

Gemalto smart cards (Cyberflex Access 64k v2, PKCSI v2.1 で設定された DER SHA1 値を持つ標準) がサポートされます。これらのスマートカードは、Chip/Smart Card Interface Devices (CCID) と互換のあるリーダを使用します。

3.3.1.4. Fedora Single Sign-on の利点

現在、多くのプロトコルやクレディンシャル保管庫を利用する、多くのセキュリティ・メカニズムが存在します。例は、SSL, SSH, IPsec, および Kerberos を含みます。Fedora SSO は上でリストされた要求事項をサポートするために、これらのスキーマを単一化することを目標としています。X.509v3 証明書を用いた Kerberos を置

き換えることを意味するわけではありません。むしろ、それらを管理しているシステムユーザーや管理者の負担を減らすために、それらを一体化させることを意味します。

この目標を達成するために、Fedora は:

- 各オペレーティングシステムにおいて単一の、共有された NSS 暗号ライブラリのインスタンスを提供します。
- 基本オペレーティングシステムに証明書システムの Enterprise Security Client (ESC) を同梱します。ESC アプリケーションは、スマートカードの挿入イベントを監視しています Certificate System サーバ製品とともに使用されるよう設計されているスマートカードをユーザーが挿入したことを検知すると、ユーザーにスマートカードを挿入する方法がユーザーインターフェースに表示されます。
- スマートカードを用いてオペレーティングシステムにログインするユーザが、Kerberos クレディンシャル（ファイルサーバにログインできるようにする、など）も取得できるよう、Kerberos と NSS を一体化します。

3.3.2. 新しいスマートカードの開始方法

システムにログインするためにスマートカードを使用でき、この技術が提供する増やされたセキュリティ・オプションの利点を得られる前に、いくつかの基本的なインストールと設定手順を実行する必要があります。これらは以下で説明されます。

注記

このセクションは、スマートカードの始め方を高いレベルでの概要を提供します。より詳細な情報は Red Hat Certificate System Enterprise Security Client Guide において入手可能です。

1. Kerberos 名とパスワードを用いてログインします。
2. `nss-tools` パッケージがロードされていることを確実にします。
3. あなたの組織固有のルート証明書をダウンロードしてインストールします。ルート CA 証明書をインストールするために以下のコマンドを使用します:

```
certutil -A -d /etc/pki/nssdb -n "root ca cert" -t "CT,C,C" -i ./ca_cert_in_base64_format.crt
```

4. システムにインストールされた次の RPM を検証します: esc, pam_pkcs11, coolkey, ifd-egate, ccid, gdm, authconfig, および authconfig-gtk。
5. スマートカード・ログインのサポートを有効にします
   a. Gnome のタイトル・バーにおいて、システム -> 管理 -> 認証を選択します。
   b. 必要に応じてマシンの root パスワードを入力します。
   c. 認証の設定ダイアログにおいて、認証タブをクリックします。
   d. スマートカードのサポートを有効にするチェックボックスを選択します。
   e. スマートカードの設定ダイアログを表示するために スマートカードを設定する... をクリックして、必要な設定を指定します:
      - ログインのためにスマートカードを要求する — このチェックボックスを外します。スマートカードを用いて正常にログインした後で、ユーザーがスマートカードなしでログインするを防ぐためにこのオプションを選択します。

- カード抜き取り時の動作 — これにより、ログインした後にスマートカードを抜いたときに何が起きるかを制御します。
  - ロック — スマートカードを抜いたときに X 画面をロックします。
  - 無視 — スマートカードを抜いても何もしません。

6. Online Certificate Status Protocol (OCSP) を有効にする必要があるなら、`/etc/pam_pkcs11/pam_pkcs11.conf` ファイルを開いて、以下の行を探します:

   ```
   enable_ocsp = false;
   ```

   次のように、この値を true に変更します:

   ```
   enable_ocsp = true;
   ```

7. スマートカードを登録します
8. CAC カードを使用しているならば、以下の手順も実行する必要があります:
   a. root アカウントに変更して、`/etc/pam_pkcs11/cn_map` というファイルを作成します。
   b. 以下のエントリを `cn_map` ファイルに追加します:

      `MY.CAC_CN.123454 -> myloginid`

      ここで、*`MY.CAC_CN.123454`* は CAC の Common Name、*`myloginid`* は UNIX ログイン ID です。
9. ログアウトします

3.3.2.1. トラブルシューティング

スマートカードを動作させるためにトラブルに遭遇したら、問題のある箇所を特定するために次のコマンドを試してください。

```
pklogin_finder debug
```

登録されたスマートカードがプラグインされている間、デバッグモードで `pklogin_finder` ツールを実行するならば、カードにある証明書からログイン ID を対応づけることがうまくいくと、証明書の検証に関する情報を出力しようとします。

3.3.3. スマートカードの登録はどのように動作しますか

スマートカードは有効な認証局 (CA: Certificate Authority) により署名された適切な証明書を受け取ったとき、##されたと言われます。これは以下で説明されるいくつかの手順に関連します。

1. ユーザーがワークステーションのスマートカードリーダにスマートカードを挿入します。このイベントは Enterprise Security Client (ESC) により認識されます。
2. 登録ページがユーザーのデスクトップに表示されます。ユーザーは必要な詳細とユーザーのシステムを完了します。そして、Token Processing System (TPS) および CA に接続します。
3. TPS は CA により署名された証明書を使用してスマートカードを登録します。

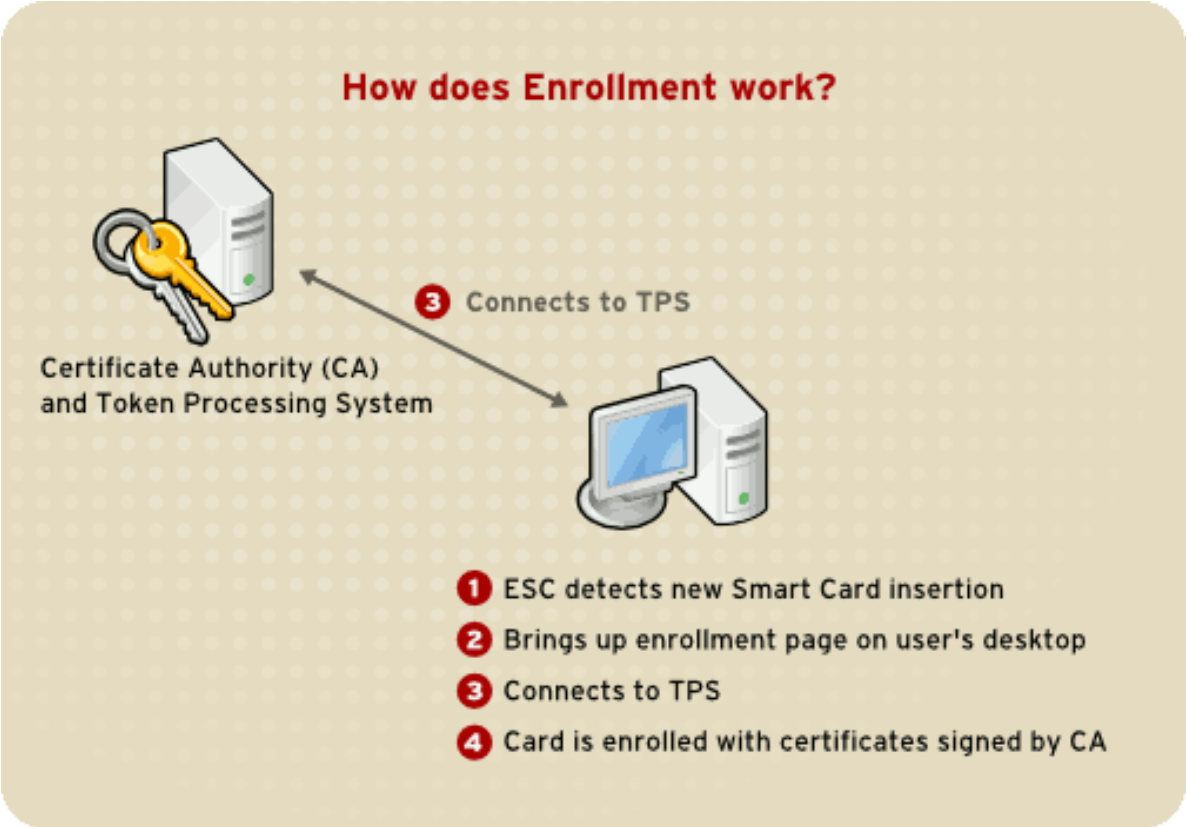

図3.4 スマートカードの登録はどのように動作しますか

3.3.4. スマートカードのログインはどのように動作しますか

このセクションは、スマートカードを用いたログインの流れについて簡単な概要を提供します。

1. ユーザーがスマートカードリーダの中にスマートカードを挿入したとき、このイベントが PAM ファシリティにより認識されます。ここで、ユーザーの PIN に対するプロンプトが出ます。
2. その後、システムはユーザーの現在の証明書を探して、それらの有効性を検証します。そして、証明書はユーザーの UID に対応づけられます。
3. これは KDC に対して検証され、ログインが許可されます。

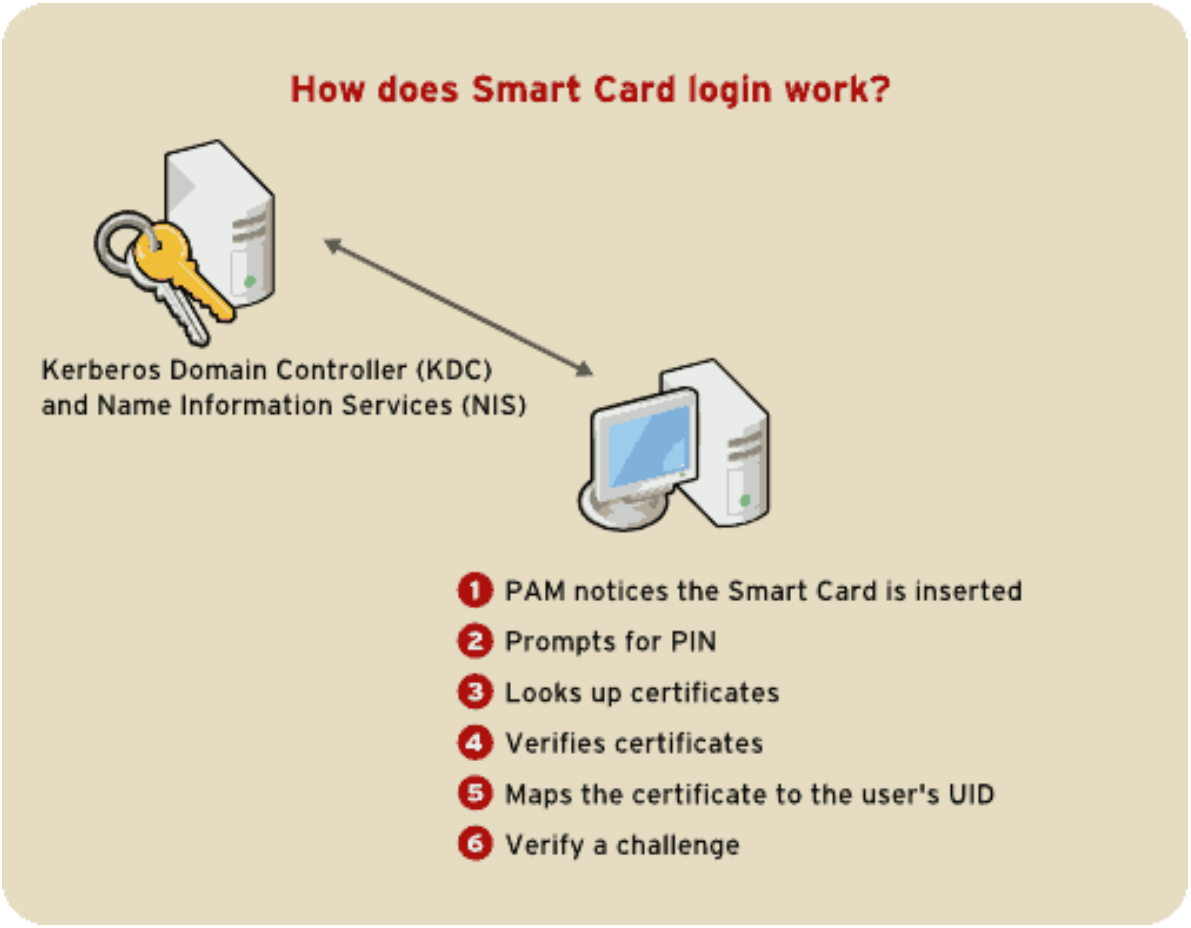

図3.5 スマートカードのログインはどのように動作しますか

注記

フォーマットされていたとしても、登録されていないスマートカードを用いてログインすることはできません。フォーマットされ、登録されたカードを用いてログインする必要があります、もしくは新しいカードを登録できるまではスマートカードを用いたログインはできません。

Kerberos と PAM に関する詳細は *#Kerberos#* および *#Pluggable Authentication Modules (PAM)#* を参照してください。

3.3.5. Firefox が SSO 用に Kerberos を使用するよう設定します

Firefox がシングルサインオンのために Kerberos を使用するよう設定できます。この機能を正しく動作させるために、Kerberos クレディンシャルを適切な KDC に送るようウェブブラウザを設定する必要があります。以下のセクションは、これを実現するために、設定の変更点と他の必要事項を説明しています。

1. Firefox のアドレスバーに、現在の設定オプションの一覧を表示するために `about:config` と入力します。
2. フィルタ フィールドに、オプションの一覧を制限するために `negotiate` と入力します。
3. ######ダイアログボックスを表示するために *network.negotiate-auth.trusted-uris* エントリをダブルクリックします。
4. 認証したいドメイン名を入力します。たとえば、*`.example.com`* です。

5. 上の手順を *network.negotiate-auth.delegation-uris* エントリに対しても、同じドメインを用いて繰り返します。

> **注記**
>
> 必要とされない Kerberos チケットをパスできるよう、この値を空白にしたままにできます。
>
> 表示されたこれら2つの設定オプションが見当たらないならば、Firefox のバージョンが Negotiate 認証をサポートしていない古すぎるバージョンである可能性があります。更新を検討すべきです。

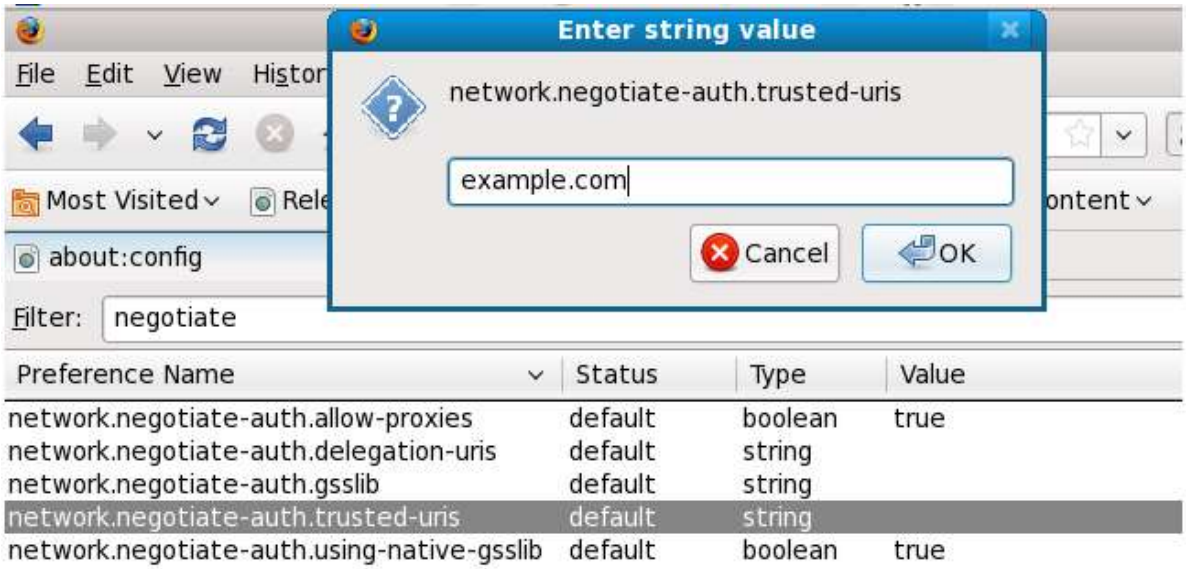

図3.6 Kerberos を用いた SSO 用に Firefox を設定

Kerberos チケットを持っていることを確実にする必要があります。コマンドシェルにおいて、Kerberos チケットを読み出すために `kinit` と入力します。利用可能なチケットの一覧を表示するために、`klist` と入力します。以下は、これらのコマンドの出力例を示しています:

```
[user@host ~] $ kinit
Password for user@EXAMPLE.COM:

[user@host ~] $ klist
Ticket cache: FILE:/tmp/krb5cc_10920
Default principal: user@EXAMPLE.COM

Valid starting     Expires            Service principal
10/26/06 23:47:54  10/27/06 09:47:54  krbtgt/USER.COM@USER.COM
        renew until 10/26/06 23:47:54

Kerberos 4 ticket cache: /tmp/tkt10920
klist: You have no tickets cached
```

3.3.5.1. トラブルシューティング

上の設定手順にしたがっても Negotiate 認証がうまく動作しないならば、認証プロセスの冗長なログを有効にします。これにより、問題の原因を見つける助けになります。冗長なログを有効にするために、以下の手順を使用します:

1. Firefox のインスタンスをすべて閉じます。

2. コマンドシェルを開いて、以下のコマンドを入力します:

```
export NSPR_LOG_MODULES=negotiateauth:5
export NSPR_LOG_FILE=/tmp/moz.log
```

3. ####### Firefox を再起動して、前に認証できなかったウェブサイトを訪問します。情報が /tmp/moz.log に記録され、問題へのヒントを与えるかもしれません。たとえば:

```
-1208550944[90039d0]: entering nsNegotiateAuth::GetNextToken()
-1208550944[90039d0]: gss_init_sec_context() failed: Miscellaneous failure
No credentials cache found
```

これは Kerberos チケットを持っていないことを意味します。kinit を実行する必要があります。

マシンから正常に kinit が実行できても、認証がうまくいかないならば、ログファイルにあるこのようなものを見かけるかもしれません:

```
-1208994096[8d683d8]: entering nsAuthGSSAPI::GetNextToken()
-1208994096[8d683d8]: gss_init_sec_context() failed: Miscellaneous failure
Server not found in Kerberos database
```

これは一般的な Kerberos の設定問題を意味します。/etc/krb5.conf ファイルの [domain_realm] セクションに正しいエントリを持つことを確実にします。たとえば:

```
.example.com = EXAMPLE.COM
example.com = EXAMPLE.COM
```

ログに何も表示されない場合、プロキシの内側にいる可能性があります。プロキシは Negotiate 認証に必要となる HTTP リクエストヘッダを取り除きます。回避策として、リクエストが変更されずに通過できるよう、代わりに HTTPS を使用しているサーバに接続を試してみることができます。そして、上で説明されたように、ログファイルを使用してデバッグを進めます。

3.4. 複数要素認証ソリューション

3.4.1. YubiKey

YubiKey は、動作のためにオープンソースソフトウェアを利用している、ハードウェア認証トークンです。このトークンは、コンピューターへキーボードとして現れる単なる USB デバイスです。トークンの1つのボタンは押すたびに、ユーザーを認証するために使われるワンタイムパスワード（OTP）を提供します。現在、ここで取り扱うこのソリューションは、いくつかの異なる実装があります。

3.4.1.1. センター・サーバーを用いた YubiKey の使用

認証サーバに問い合わせることができるようにする、コンピュータの認証を許可する PAM モジュールが、すでに Fedora リポジトリに存在します。サーバは、ドメインのレベルでセットアップすることも、Yubico のサーバを利用することもできます。この認証の方法は、ドメインにおいて複数のユーザーが複数のコンピュータにアクセスする必要がある、エンタープライズの素晴らしいソリューションです。以下の手順はこのセットアップを説明します。

1. Install *pam_yubico*

2. 二要素認証のために /etc/pam.d/gdm-password を開き、以下の位置を探します:

```
auth substack password-auth
```

この後ろの新しい行に、次を追加します:

```
auth sufficient pam_yubico.so id=16
```

3. パスワード認証なしで YubiKey トークンを単独で使用するために、上の手順から最初の行を削除して、2番目のもので置き換えます。

4. YubiKey を初めて追加するために YubiKey トークンを置きます。すべての OTP の最初の12文字を見るか、または *http://radius.yubico.com/demo/Modhex_Calculator.php* を訪問して、ページにあるテキストボックスの中に OTP を入力した後 Modhex エンコードされた文字列をコピーすることにより、これがなされます。

5. ユーザーの YubiKey を設定ファイルに追加します。`/etc/yubikey_mapping` においてグローバルに、もしくは`~/.yubico/authorized_yubikeys` において個々のユーザーにより、これがなされます。以下はその構文です:

```
username:yubikey_token:another_yubikey_token
```

6. ログアウトします。再びログインしようとするとき、システムをどのように設定したかにより、パスワードと YubiKey OTP、または両方ともを入力するようプロンプトが出ます。

注記

認証サーバーへの接続が要求されます、もしくは正しく認証されないでしょう。これは、安定したネットワーク接続性をもたないシステムにとって有害でしょう。

3.4.1.2. YubiKey を用いたウェブサイトの認証

このガイドの範囲外ではありますが、YubiKey はこの認証方法をサポートするウェブサイトへ認証するようにできます。これらのウェブサイトは一般的に Yubico の認証サーバーをサポートしますが、いくつかは上のセンターサーバーと同じようにセットアップすることができます。Yubico は、特定のウェブサイトで利用されている OpenID サービスも提供します。

3.5. Pluggable Authentication Modules (PAM)

ユーザーがシステムにアクセスするのを認可するプログラムは、お互いのアイデンティティを確認するために（つまり、ユーザーがユーザーであるとわかることを証明するために）、##を使用します。

歴史的に、各プログラムはユーザーを認証する自身の方法を持ちます。Fedora において、多くのプログラムは *Pluggable Authentication Modules* (PAM) と呼ばれる集中化した認証メカニズムを使用するよう設定されています。

PAM は抜き差し可能な、モジュール型のアーキテクチャを使用します。それは、システム管理者がシステムに対して認証ポリシーを設定することにおいて非常に大きな柔軟性を与えます。

多くの状況において、PAM 対応のアプリケーションに対してデフォルトの PAM 設定は十分です。しかしながら、ときどき、PAM 設定ファイルを編集する必要があります。PAM の設定誤りはシステムのセキュリティを危険にさらす可能性があるので、変更を始める前にこれらのファイルの構造を理解することは重要です。詳細は *#PAM ##########* を参照してください。

3.5.1. PAM の利点

PAM は以下の利点を提供します:

- 幅広い種類のアプリケーションで使うことができる一般的な認証スキーマ。
- システム管理者とアプリケーション開発者の双方に対して認証についての重要な柔軟性と制御。
- プログラマがプログラムを書くためにそれ自身の認証スキーマを作成しなくて済むようにする1つの完全にドキュメント化されたライブラリ。

3.5.2. PAM 設定ファイル

/etc/pam.d/ ディレクトリは、PAM 対応の各アプリケーションに対する PAM 設定ファイルを含みます。PAM の以前のバージョンでは、/etc/pam.conf ファイルが使われましたが、いまや不当とされ、/etc/pam.d/ ディレクトリが存在しない場合のみ使用されます。

3.5.2.1. PAM サービス・ファイル

各 PAM 対応アプリケーションまたは#### は /etc/pam.d/ ディレクトリにファイルを持ちます。このディレクトリにある各ファイルは、それがアクセスを制御するサービスと同じ名前を持ちます。

PAM 対応プログラムは、そのサービスを定義して、/etc/pam.d/ ディレクトリにそれ自身の PAM 設定ファイルをインストールする責任があります。たとえば、login プログラムはそのサービス名を login として定義し、/etc/pam.d/login PAM 設定ファイルをインストールします。

3.5.3. PAM 設定ファイルの形式

各 PAM 設定ファイルは以下のようにフォーマットされたディレクティブのグループを含みます:

```
<module interface>  <control flag>  <module name>  <module arguments>
```

これらの要素はそれぞれ以下のセクションにおいて説明されます。

3.5.3.1. モジュール・インタフェース

PAM モジュール・インタフェースは現在4種類が利用可能です。これらはそれぞれ認可プロセスの異なる観点に対応します:

- auth — このモジュール・インタフェースはユーザーを認証します。たとえば、パスワードの正当性を要求して検証します。このインタフェースを持つモジュールは、グループのメンバーシップや Kerberos チケットのような、クレディンシャルもセットします。
- account — このモジュール・インタフェースはアクセスが許可されていることを検証します。たとえば、ユーザー・アカウントが期限切れかどうか、またはユーザーが特定の期間にログインを許可されているかどうかをチェックします。
- password — このモジュール・インタフェースはユーザーのパスワードを変更するために使われます。
- session — このモジュール・インタフェースは、ユーザーのセッションを設定して管理します。このインタフェースを持つモジュールは、ユーザーのホームディレクトリをマウントしたり、ユーザーのメールボックスを作成したりするような、アクセスを許可するために必要とされる追加のタスクも実行できます。

注記

それぞれのモジュールは、何らかのもしくはすべてのモジュール・インタフェースを提供できます。たとえば、pam_unix.so は全4つのモジュール・インタフェースを提供します。

PAM 設定ファイルにおいて、モジュール・インタフェースは第1フィールドに定義されます。たとえば、設定における典型的な行はこのように見えます:

```
auth required pam_unix.so
```

これは PAM が pam_unix.so モジュールの auth インタフェースを使用するよう指示します。

3.5.3.1.1. モジュール・インタフェースのスタック

モジュール・インタフェースのディレクティブは、複数のモジュールが1つの目的のために一緒に使えるよう、### できます、もしくはお互いに重ねておくことができます。モジュールの制御フラグが "sufficient" または "requisite" 値（これらのフラグの詳細については ####### を参照してください。）を使うならば、どのモジュールがリストされるかの順番は認証プロセスにとって重要です。

スタックすることは、ユーザーが認証を許可される前に存在するために、管理者が特定の条件を要求することを簡単にします。たとえば、reboot コマンドは普通、PAM 設定ファイルに見られるように、いくつかのスタックされたモジュールを使用します。

```
[root@MyServer ~]# cat /etc/pam.d/reboot
#%PAM-1.0
auth sufficient pam_rootok.so
auth required pam_console.so
#auth include  system-auth
account required pam_permit.so
```

- 1行目はコメントであり、処理されません。
- auth sufficient pam_rootok.so — この行は、UID を確認することにより、現在のユーザーが root であるかどうかをチェックするために pam_rootok.so モジュールを使用します。このテストが成功すると、他のモジュールは参照されず、コマンドが実行されます。このテストが失敗すると、次のモジュールが参照されます。
- auth required pam_console.so — この行は、ユーザーを認証する試行のために pam_console.so モジュールを使用します。ユーザーがすでにコンソールにログインしていると、pam_console.so は /etc/security/console.apps/ ディレクトリにサービス名 (reboot) と同じ名前を持つファイルがあるかどうかをチェックします。
- #auth include system-auth — この行はコメントされ、処理されません。
- account required pam_permit.so — この行は、コンソールにログインしている root ユーザーまたは誰かがシステムを再起動できるようにするために pam_permit.so モジュールを使用します。

3.5.3.2. 制御フラグ

すべての PAM モジュールは、呼び出されたときに成功または失敗の結果を生成します。制御フラグは結果とともに何を実行するかを PAM に教えます。モジュールは特定の順番でスタックされ、制御フラグは特定のモジュールの成功または失敗が、サービスへとユーザーを認証する目標全体にとって、どのくらい重要であるかを決めます。

事前定義済みの制御フラグが4つあります:

- required — モジュールは、認証を続けるために必ず成功しなければいけません。テストがここで失敗すると、すべてのモジュールの結果がインタフェースが完了するその参照をテストするまで、ユーザーに通知されません。
- requisite — モジュールは、認証を続けるために成功しなければいけません。しかし、テストがここで失敗すると、最初に失敗した required ### requisite モジュールのテストを反映したメッセージとともにユーザーへ直ちに通知されます。
- sufficient — モジュールの結果は失敗しても無視されます。しかし、sufficient フラグのついたモジュールの結果が成功であり、##、required フラグのついたモジュールがこの前で失敗していなければ、他の結果は何も必要とされず、ユーザーはサービスへ認証されます。
- optional — モジュールの結果は無視されます。optional としてフラグのついたモジュールは、他のモジュールがインタフェースを参照されないときのみ、認証成功のために必要とされます。

重要

required モジュールが呼び出される順番は重要ではありません。sufficient および requisite 制御フラグのみが重要になる順番を与えます。

今、PAM のより精細な制御を可能にする新しい制御フラグの構文が利用可能です。

/usr/share/doc/pam-***<version-number>***/ ディレクトリ(*<version-number>* はシステムの PAM バージョン番号)にある pam.d マニュアル・ページおよび PAM ドキュメントは、この新しい構文を詳細に説明しています。

3.5.3.3. モジュール名

モジュール名は、指定されたモジュール・インタフェースを含む、挿抜可能なモジュールの名前を持つ PAM を提供します。Fedora の以前のバージョンでは、モジュールへの絶対パスが PAM 設定ファイルにおいて与えられていました。しかしながら、/lib64/security/ ディレクトリに64ビット PAM モジュールを保存する、*multilib* システムの出現により、モジュールの正しいバージョンを指定する、libpam の適切なバージョンにアプリケーションがリンクされるので、ディレクトリ名は廃止されました。

3.5.3.4. モジュール引数

PAM はいくつかのモジュールに対して認証中に抜き差し可能なモジュールに情報を受け渡すため *arguments* を使用します。

たとえば、pam_userdb.so モジュールはユーザーを認証するために Berkeley DB ファイルに保存された情報を使用します。Berkeley DB は多くのアプリケーションに組み込まれているオープンソースのデータベースシステムです。モジュールは、Berkeley DB が要求されたサービスに対して使用するためにデータベースを知ることができるよう、db 引数を取ります。

以下は、PAM 設定における典型的な pam_userdb.so 行です。*<path-to-file>* は Berkeley DB データベースファイルへのフルパスです:

```
auth required pam_userdb.so db=<path-to-file>
```

無効な引数は####無視されます。そうでなければ、PAM モジュールの成功または失敗に影響を与えます。しかし、いくつかのモジュールは、無効な引数において落ちるかもしれません。多くのモジュールは /var/log/secure ファイルにエラーを報告します。

3.5.4. サンプル PAM 設定ファイル

以下はサンプル PAM アプリケーション設定ファイルです:

```
#%PAM-1.0
auth  required  pam_securetty.so
auth  required  pam_unix.so nullok
auth  required  pam_nologin.so
account  required  pam_unix.so
password required  pam_cracklib.so retry=3
password required  pam_unix.so shadow nullok use_authtok
session required  pam_unix.so
```

- 行の最初にハッシュ記号 (#) により示された、最初の行はコメントです。
- 2〜4行目はログイン認証用の3つのモジュールを積み重ねています。

  auth required pam_securetty.so — このモジュールは、ユーザーが root としてログインしようとしている###、ユーザーがログインしている tty が /etc/securetty ファイル（存在###）にリストされていることを確実にします。

  tty がファイルにリストされていなければ、root としてログインするすべての試行は Login incorrect メッセージとともに失敗します。

  auth required pam_unix.so nullok — このモジュールは、ユーザーに対してパスワードを促し、/etc/passwd および、存在すれば /etc/shadow に保存されている情報を用いてパスワードをチェックします。

  - 引数 nullok は pam_unix.so モジュールに空のパスワードを許可するよう指示します。
- auth required pam_nologin.so — これは最後の認証手順です。/etc/nologin ファイルが存在するかどうかをチェックします。もし存在して、ユーザーが root でなければ、認証は失敗します。

  

  注記

  この例において、auth モジュールが失敗したときでも、3つの auth モジュールはすべてチェックされます。これにより、ユーザーが認証のどの段階において失敗したかを知ることを防ぎます。そのような知識が攻撃者の手にわたると、攻撃者がシステムをクラックする方法をより簡単に推定することができるようになります。

- account required pam_unix.so — このモジュールはすべての必要なアカウント検証を実行します。たとえば、shadow パスワードが有効にされていれば、pam_unix.so モジュールのアカウント・インタフェースは、アカウントが期限切れであるかどうか、または認められた猶予期間内にパスワードを変更していなかったかどうかを確認するためにチェックします。
- password required pam_cracklib.so retry=3 — パスワードが期限切れになっていれば、pam_cracklib.so モジュールのパスワード・コンポーネントは新しいパスワードのためにプロンプトを出します。パスワードが辞書ベースのパスワード・クラック・ツールにより簡単に決められるかどうかを確認するために、新しく作成されたプログラムをテストします。

  - 引数 retry=3 は、テストが初めて失敗すると、ユーザーは強いパスワードを作成するためにあと2回チャンスを持つことを指定します。

- password required pam_unix.so shadow nullok use_authtok — この行は、プログラムがユーザーのパスワードを変更するならば、pam_unix.so モジュールの password インタフェースを使用するよう指定します。

  - 引数 shadow は、ユーザーのパスワードを更新するときに shadow パスワードを作成するようモジュールに指示します。

  - 引数 nullok は、ユーザーが空のパスワード##パスワードを変更できるようモジュールに指示します、さもなければ空のパスワードはアカウント・ロックとして取り扱われます。

  - この行の最後の引数 use_authtok は、PAM モジュールをスタックするときに、順番の重要性の良い例を提供します。この引数は、ユーザーに新しいパスワードのためのプロンプトを表示しないよう、モジュールに指示します。代わりに、以前 password モジュールにより記録されたすべてのパスワードが受け付けられます。このように、すべての新しいパスワードは受け付けられる前にセキュアなパスワードのために pam_cracklib.so テストを通過しなければいけません。

- session required pam_unix.so — 最後の行は、pam_unix.so モジュールのセッション・インタフェースがセッションを管理するよう指示します。このモジュールは、せそれぞれのセッションの最初と最後に、ユーザー名とサービスタイプを /var/log/secure に記録します。このモジュールは追加の機能のために他の session モジュールを用いてそれをスタックすることにより補完されます。

3.5.5. PAM モジュールの作成

PAM 対応のアプリケーションにより使用するために、いつでも新しい PAM モジュールを作成または追加できます。

たとえば、開発者がワンタイムパスワードの生成方式を作成し、それをサポートするために PAM モジュールを書きます。PAM 対応プログラムは直ちに新しいモジュール、および再コンパイルされる、さもなければ修正されることなく、パスワード方式を使用できます。

これにより、開発者とシステム管理者が、認証方法を再コンパイルすることなく異なるプログラムに対してそれらを、混ぜて組み合わせるだけでなく、テストできるようにします。

書き込みモジュールのドキュメントは /usr/share/doc/pam-***<version-number>***/ ディレクトリに含まれます。ここで *<version-number>* はシステムにおける PAM のバージョン番号です。

3.5.6. PAM と管理クレディンシャルのキャッシュ

Fedora にある多くのグラフィカル管理ツールは、pam_timestamp.so モジュールを使用してユーザーに5分間まで権限を上昇させます。このメカニズムがどのように機能するかを理解することは重要です。なぜなら、ユーザーが pam_timestamp.so を有効なままで端末から離れると、端末に物理的にアクセスした誰かにより不正に操作されるからです。

PAM timestamp スキーマにおいて、グラフィカル管理アプリケーションが起動されたときに、ユーザに対して root パスワードのためにプロンプトを出します。ユーザーが認証されたとき、pam_timestamp.so モジュールがタイムスタンプ・ファイルを作成します。デフォルトで、これは /var/run/sudo/ ディレクトリに作成されます。もし、タイムスタンプ・ファイルがすでに存在すると、グラフィカル管理プログラムはパスワードを促しません。代わりに、pam_timestamp.so モジュールが、ユーザーに対して変更されない管理アクセスを追加の5分を割り当てる、タイムスタンプ・ファイルを新たにします。

/var/run/sudo/<user> ファイルを調査することにより、タイムスタンプ・ファイルの実際の状態を検証できます。デスクトップに対して、関連するファイルは unknown:root です。それが存在して、タイムスタンプが5分以内であれば、クレディンシャルは有効です。

タイムスタンプ・ファイルの存在は、パネルの通知エリアに表れる、認証アイコンにより示されます。

図3.7 認証アイコン

3.5.6.1. タイムスタンプ・ファイルの削除

PAM タイムスタンプが有効であるとき、コンソールを去る前に、タイムスタンプ・ファイルが廃棄されることが推奨されます。グラフィカル環境からこれを実行するために、パネルにある認証アイコンをクリックします。これにより、ダイアログボックスが表示されます。有効なタイムスタンプ・ファイルを廃棄するために Forget Authorization ボタンをクリックします。

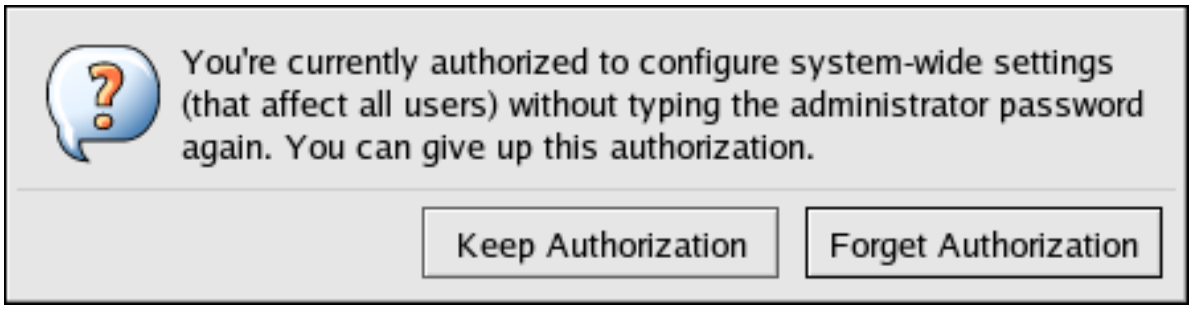

図3.8 認証ダイアログの却下

PAM タイムスタンプ・ファイルに関連して以下の事項に気をつけるべきです:

- ssh を用いてリモートでシステムにログインしているならば、タイムスタンプ・ファイルを廃棄するために /sbin/pam_timestamp_check -k root コマンドを使用します。
- あなたが特権アプリケーションを起動した同じターミナル・ウィンドウから、/sbin/pam_timestamp_check -k root コマンドを実行する必要があります。
- /sbin/pam_timestamp_check -k コマンドを使用するために、元々 pam_timestamp.so モジュールに関連したユーザーとしてログインしなければいけません。このコマンドを使用するために root としてログインしないでください。
- デスクトップにおいて(アイコンにある Forget Authorization アクションを使用せずに)クレディンシャルを削除したければ、以下のコマンドを使用します:

  ```
  /sbin/pam_timestamp_check -k root </dev/null >/dev/null 2>/dev/null
  ```

  このコマンドを使用するのに失敗すると、コマンドを実行した pty からクレディンシャル(あれば)のみを削除します。

pam_timestamp_check を使用してタイムスタンプ・ファイルを廃棄する方法に関する詳細は pam_timestamp_check マニュアル・ページを参照してください。

3.5.6.2. 一般的な pam_timestamp ディレクティブ

pam_timestamp.so モジュールはいくつかのディレクティブを受け付けます。以下は最も一般的に使われるオプションです:

- timestamp_timeout — タイムスタンプ・ファイルが有効である期間を(秒単位で)指定します。デフォルト値は300(5分)です。
- timestampdir — タイムスタンプ・ファイルが保存されるディレクトリを指定します。デフォルト値は /var/run/sudo/ です。

3.5.7. PAM とデバイスの所有

Fedora では、マシンの物理コンソールに最初にログインしたユーザーが特定のデバイスを操作でき、通常 root ユーザーのために予約されている特定のタスクを実行できます。これは、pam_console.so と呼ばれる PAM モジュールにより制御されます。

3.5.7.1. デバイスの所有

ユーザーが Fedora システムにログインするとき、pam_console.so モジュールが login またはグラフィカル・ログイン・プログラム（gdm, kdm, および xdm）により呼び出されます。このユーザーが物理コンソールにログインした最初のユーザー — *console user* として参照されます — ならば、モジュールは通常は root により所有されるさまざまなデバイスの所有権をユーザーに与えます。コンソール・ユーザーは、そのユーザーに対する最後のローカル・セッションが終了するまで、これらのデバイスを所有します。このユーザーがログアウトした後、デバイスの所有権は root ユーザーに戻されます。

影響を受けるデバイスは、サウンドカード、ディスクドライブ、および CD-ROM ドライブを含みますが、限定されるわけではありません。

この機能により、ユーザーが root アクセスを得ることなくこれらのデバイスを操作できるようになります。このようにコンソール・ユーザーの一般的なタスクを単純化します。

以下のファイルを編集することで、pam_console.so により制御されるデバイスのリストを編集できます:

- /etc/security/console.perms
- /etc/security/console.perms.d/50-default.perms

上のファイルにあるこれらのリストから他のデバイスのパーミッションを変更できます、もしくは指定されたデフォルトを上書きできます。50-default.perms ファイルを変更するよりはむしろ、新しいファイル（たとえば、***xx***-name.perms）を作成して、必要な修正を入力します。新しいデフォルト・ファイルの名前は、50より大きな数字（たとえば、51-default.perms）で始まらなければいけません。これにより、50-default.perms ファイルにあるデフォルトを上書きします。

警告

gdm, kdm, または xdm ディスプレイ・マネージャー設定ファイルは、リモート・ユーザーがログインできるよう変更されます。##、ホストがランレベル5で実行するよう設定され、/etc/security/console.perms にある <console> および <xconsole> ディレクティブを以下の値に変更することが望ましいです。

```
<console>=tty[0-9][0-9]* vc/[0-9][0-9]* :0¥.[0-9] :0 ⏎ <xconsole>=:0¥.[0-9] :0
```

これにより、リモートユーザーがマシンにおけるデバイスおよび制限されたアプリケーションへのアクセス権を得ることを防ぎます。

gdm, kdm, または xdm ディスプレイ・マネージャの設定ファイルが、リモートユーザーがログインできるよう変更されていて、##、ホストが5以外のあらゆるマルチユーザー・ランレベルで実行するよう設定されているならば、<xconsole> ディレクティブを完全に削除して、<console> ディレクティブを以下の値に変更するようアドバイスします:

```
<console>=tty[0-9][0-9]* vc/[0-9][0-9]*
```

3.5.7.2. アプリケーションのアクセス

コンソール・ユーザーは /etc/security/console.apps/ ディレクトリにおいて使用するために設定された特定のプログラムへのアクセス権も持ちます。

このディレクトリは、コンソール・ユーザーが/sbin および /usr/sbin にある特定のアプリケーションを実行できるようにする設定ファイルを含みます。

これらの設定ファイルはセットアップするアプリケーションと同じ名前を持ちます。

コンソール・ユーザーがアクセス権を持つアプリケーションの注目すべきグループの1つは、システムをシャットダウンまたは再起動する3つのプログラムです:

- /sbin/halt
- /sbin/reboot
- /sbin/poweroff

これらは PAM 対応のアプリケーションのため、使用するために必要に応じて pam_console.so モジュールを呼び出します。

3.5.8. 追加のリソース

以下のリソースは PAM を使用したり設定したりする方法を詳細に説明しています。これらのリソースに加えて、PAM 設定ファイルがどのような構造をしているかをより理解するためにシステムにあるそれらを読んでください。

3.5.8.1. インストールされている PAM ドキュメント

- PAM 関連のマニュアル・ページ — いくつかのマニュアル・ページが PAM に関連するさまざまなアプリケーションと設定ファイルに対して存在します。以下はいくつかのより重要なマニュアル・ページの一覧です。

  設定ファイル
  - pam — PAM に関する素晴らしい入門情報です、PAM 設定ファイルの構造と目的を含みます。

    このマニュアル・ページは /etc/pam.conf および /etc/pam.d/ ディレクトリにある個々の設定ファイルを説明します。デフォルトで、Fedora は /etc/pam.d/ ディレクトリにある個々の設定ファイルを使用して、/etc/pam.conf が存在しても無視します。

  - pam_console — pam_console.so モジュールの目的を記述します。PAM 設定ファイルの中にあるエントリーに対する適切な構文も記述します。

  - console.apps — /etc/security/console.apps 設定ファイルで利用可能なフォーマットとオプションを記述します。これは、どのアプリケーションが PAM により割り当てられたコンソール・ユーザーによりアクセス可能です。

  - console.perms — /etc/security/console.perms 設定ファイルで利用可能なフォーマットとオプションを記述します。これは、PAM により割り当てられるコンソール・ユーザーのパーミッションを指定します。

  - pam_timestamp — pam_timestamp.so モジュールを表します。

- /usr/share/doc/pam-***<version-number>*** — *System Administrators' Guide*、*Module Writers' Manual* および *Application Developers' Manual* だけでなく、PAM 標準 DCE-RFC 86.0 のコピーを含みます。ここで *<version-number>* は PAM のバージョンです。

- /usr/share/doc/pam-**<*version-number*>**/txts/README.pam_timestamp — pam_timestamp.so PAM モジュールに関する情報を含みます。ここで <*version-number*> は PAM のバージョン番号です。

3.5.8.2. 有用な PAM ウェブサイト

- *http://www.kernel.org/pub/linux/libs/pam/* — Linux-PAM プロジェクトの一次的なディストリビューションのウェブサイト、さまざまな PAM モジュール、FAQ、さらなる PAM ドキュメントに関する情報を含みます。

注記

上のウェブサイトにあるドキュメントは、PAM の最新リリースのアップストリームのバージョンに対するもので、Fedora に含まれるバージョンの PAM に対して、完全に正確ではないかもしれません。

3.6. TCP Wrappers と 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、サーバ管理者が直面する最も重要なセキュリティの仕事のひとつです。Fedora はこのためにいくつかのツールを提供します。たとえば、iptables ベースのファイルウォール・フィルタ¥nは、カーネルのネットワーク・スタックの中で歓迎されないネットワーク・パケットを消します。それを利用するネットワーク・サービスに対して、*TCP Wrappers* はどのホストが "######" ネットワーク・サービスへの接続を許可もしくは拒否されるかを定義することにより、さらなる保護層を追加します。そのようなラップされたネットワーク・サービスの1つは、xinetd ######## です。それはネットワーク・サービスのサブセットへの接続を制御し、アクセス制御をさらに精錬するので、このサービスはスーパーサーバと呼ばれます。

*#3.9#####################*は、これらのツールがネットワーク・サービスを保護するためにどのように動作するかに関する基本的な説明です。

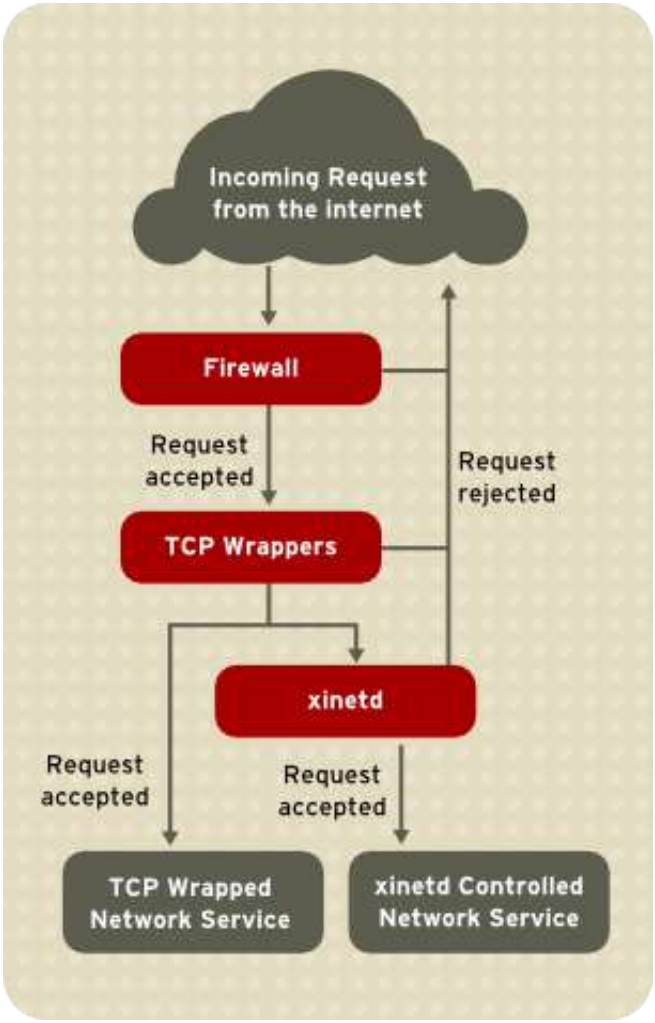

図3.9 ネットワーク・サービスへのアクセス制御

この章はネットワーク・サービスへのアクセスを制御することにおける TCP Wrappers および xinetd の役割に焦点を当てます。そして、これらのツールがログ取得と利用管理を向上するためにどのように使われるかを概説します。iptables を用いてファイアウォールを使用することに関する詳細は*#Using Firewalls#*を参照してください。

3.6.1. TCP Wrappers

TCP Wrappers パッケージ (tcp_wrappers) はデフォルトでインストールされ、ネットワーク・サービスに対するホスト・ベースのアクセス制御を提供します。パッケージの中にある最も重要なコンポーネントは /usr/lib/libwrap.a ライブラリです。一般的な用語で、TCP ラップされたサービスとは libwrap.a ライブラリに備えてコンパイルされたものです。

TCP ラップされたサービスに接続を試行するとき、クライアントが接続を許可されるかどうかを決めるために、サービスはまずホストの access ファイル (/etc/hosts.allow および /etc/hosts.deny) を参照します。多くの場合、リクエストしているクライアントとリクエストされたサービスの名前を、/var/log/secure または /var/log/messages に書き込むために、syslog デーモンを使用します。

クライアントが接続を許可されると、TCP Wrappers がコネクションの制御をリクエストされたサービスに開放し、クライアントとサーバ間のコミュニケーションにおいてそれ以上は取り入れません。

アクセス制御とロギングに加えて、リクエストされたネットワーク・サービスへのコネクションの拒否や開放をする前に、TCP Wrappers はクライアントとやりとりするためにコマンドを実行できます。

TCP Wrappers はすべてのサーバ管理者のセキュリティ・ツールの備蓄庫へと重要な追加をするので、Fedora に含まれる多くのネットワーク・サービスは libwrap.a ライブラリへリンクされます。そのようなアプリケーションのいくつかは /usr/sbin/sshd, /usr/sbin/sendmail, および /usr/sbin/xinetd を含みます。

注記

ネットワーク・サービスのバイナリが libwrap.a とリンクしているかを確認するために、root ユーザーとして以下のコマンドを入力します:

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

<binary-name> をネットワーク・サービスのバイナリの名前で置き換えます。

コマンドが何も出力せずにプロンプトが戻ってくると、ネットワーク・サービスは libwrap.a へとリンクされて####。

以下の例は /usr/sbin/sshd が libwrap.a とリンクしていることを意味します:

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

3.6.1.1. TCP Wrappers の利点

TCP Wrappers は他のネットワーク・サービス制御のテクニックに比べて以下の利点を提供します。

- ################################## — 接続しているクライアントとラップされたネットワーク・サービス双方が TCP Wrappers が使用されていることに気がつきません。正当なユーザーは記録され、要求したサービスに接続される一方、禁止されたクライアントからの接続は失敗します。
- ############ — TCP Wrappers は、多くのサーバ・アプリケーションがアクセス制御設定ファイルの一般的なセットを共有でき、よりシンプルな管理をできるようにするため、保護するネットワーク・サービスと独立して動作します。

3.6.2. TCP Wrappers の設定ファイル

クライアントがサービスへ接続を許可するかを決めるために、TCP Wrappers は、一般的に *hosts access* ファイルとして参照される、以下の2つのファイルを参照します:

- /etc/hosts.allow
- /etc/hosts.deny

TCP ラップされたサービスがクライアントのリクエストを受け取ったとき、以下の手順が実行されます:

1. ***/etc/hosts.allow*######## — TCP ラップされたサービスは順番に /etc/hosts.allow ファイルを解析し、そのサービスのために指定された最初のルールを適用します。
2. ***/etc/hosts.deny*** ######## — TCP ラップされたサービスは /etc/hosts.deny ファイルを順番に解析します。マッチするルールを見つけると、接続を拒否します。見つからなければ、サービスへのアクセスが許可されます。

ネットワーク・サービスを保護するために TCP Wrappers を使用するとき、考慮する重要なポイントは以下のとおりです:

- hosts.allow にあるアクセス・ルールが最初に適用されるので、hosts.deny で指定されたルールに優先されます。そのため、サービスへのアクセスが hosts.allow で許可されると、同じサービスに対する hosts.deny にあるアクセス拒否ルールは無視されます。
- 各ファイルにあるルールは上から下へ読み込まれ、与えられたサービスに最初にマッチするルールが1つだけ適用されます。ルールの順番は極めて重要です。
- サービスに対するルールがどちらのファイルにも見つからなければ、もしくはファイルが存在しなければ、サービスへのアクセスは許可されます。
- TCP ラップされたサービスは、hosts access ファイルをキャッシュしません。そのため、hosts.allow や hosts.deny の変更はすべて、ネットワーク・サービスを再起動しなくても、直ちに効果を持ちます。

警告

hosts アクセス・ファイルの最後の行が改行文字（Enter キーを押すことにより作成されます）でなければ、ファイルにある最後のルールは失敗して、エラーが /var/log/messages または /var/log/secure のどちらかに記録されます。バックスラッシュ文字を用いることなく複数行にわたるルールに対しても同様です。以下の例は、これらの状況どちらかによる、ルールの失敗に対するログメッセージの関連する部分を説明します:

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

3.6.2.1. アクセス・ルールのフォーマット

/etc/hosts.allow と /etc/hosts.deny のフォーマットは同じです。空行とハッシュ (#) で始まる行は無視されます。

各ルールはネットワーク・サービスへのアクセスを制御するために以下の基本的なフォーマットを使用します:

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

- *<daemon list>* — カンマ区切りのプロセス名（サービス名では#####) の一覧、もしくは ALL ワイルドカード。デーモンの一覧はより柔軟性を許すためにオペレータ（#####参照）も受け付けます。
- *<client list>* — ルールにより影響するホストのホスト名、ホスト IP アドレス、特別なパターン、またはワイルドカードのカンマ区切りのリスト。クライアント・リストはより柔軟性を持たせるために、##### にリストされた演算子も受け付けます。
- *<option>* — ルールが起動されたときに実行されるオプションのアクションまたはアクションのコロン区切りのリスト。オプション・フィールドは、拡張、シェルの起動、アクセスの許可または拒否、および他のロギング動作をサポートします。

注記

上の専門用語の詳細は、このガイドの他のところで見つけられます:

- ##########
- ######
- ####
- ##############

以下はサンプルの hosts アクセス・ルールです:

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

このルールは、TCP Wrappers が example.com ドメインにあるすべてのホストからの FTP デーモン (vsftpd) へのコネクションを待つよう指示します。このルールが hosts.allow に表れると、コネクションは受け付けられます。このルールが hosts.deny にある表れると、コネクションは拒否されます。

次のサンプル hosts access ルールは、より複雑で、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・フィールドはバックスラッシュ (¥) が先につけられ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、長さのためルールが失敗することを防ぎます。

このサンプル・ルールは次のことをしています。SSH デーモン (sshd) への接続が example.com ドメインにあるホストから試みられると、特別なログファイルに試行を追加するために echo コマンドを実行して、コネクションが拒否されます。オプションの deny ディレクティブが使われているので、この行は hosts.allow ファイルに表れたとしてもアクセスが拒否されます。利用可能なオプションの詳細は ############## を参照してください。

3.6.2.1.1. ワイルドカード

ワイルドカードは TCP Wrappers がより簡単にデーモンやホストのグループとマッチできるようにします。それらはアクセス・ルールのクライアント・リスト・フィールドにおいてより頻繁に使われます。

以下のワイルドカードが利用可能です:

- ALL — すべてにマッチします。デーモン・リストとクライアント・リストに対して使えます。
- LOCAL — localhost のようなピリオド (.) を含まないすべてのホストにマッチします。
- KNOWN — ホスト名またはホスト・アドレスが既知であるかユーザーが既知である、すべてのホストにマッチします。
- UNKNOWN — ホスト名またはホスト・アドレスが未知であるかユーザーが未知である、すべてのホストにマッチします。
- PARANOID — ホスト名とホスト・アドレスが一致しない、すべてのホストにマッチします。

重要

KNOWN, UNKNOWN, および PARANOID ワイルドカードは正しい動作が DNS サーバの機能に依存するので、注意して使用するべきです。名前解決の破壊により、正当なユーザーがサービスにアクセスを得るのを妨害するかもしれません。

3.6.2.1.2. パターン

パターンは、クライアント・ホストのグループをより正確に指定するために、クライアント・フィールドにおいて使用されます。

以下は、クライアント・フィールドにおけるエントリーの一般的なパターンのリストです。

- *#### (.) ########* — ホスト名の始めにピリオドを置くことにより、リストされたコンポーネント名を共有するすべてのホストにマッチします。以下の例はexample.com ドメインにあるすべてのホストに適用されます。:

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

- *#### (.) #### IP ####* — IP アドレスの最後にピリオドを置くことにより、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マッチします。以下の例は 192.168.x.x ネットワークにあるすべてのホストに適用されます:

```
ALL : 192.168.
```

- *IP ####/##########* — ネットマスク表現は、IP アドレスの特定のグループへのアクセスを制御するためのパターンとしても使われます。192.168.0.0 から 192.168.1.255 までのアドレス範囲を持つすべてのホストに適用されます:

```
ALL : 192.168.0.0/255.255.254.0
```

重要

IPv4 アドレス空間で動作しているとき、アドレス/プレフィックス長 (*prefixlen*) ペアの宣言 (CIDR 表記) はサポートされません。IPv6 ルールのみがこの形式を利用できます。

- *[IPv6 ####]/###########* — [ネット]/プレフィックス長は IPv6 アドレスの特定のグループに対するアクセスを制御するためにパターンとして使われます。以下の例は、3ffe:505:2:1:: から 3ffe:505:2:1:ffff:ffff:ffff:ffff までのアドレス範囲を持つすべてのホストに適用されます:

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

- *###### (*)* — アスタリスクは、他の形式のパターンを含むクライアント・リストと混在されない限り、ホスト名または IP アドレスのグループ全体にマッチするために使用されます。以下の例は example.com ドメインの中にあるホストすべてに:

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

- ##### *(/)* — クライアント・リストがスラッシュで始まっていると、ファイル名として取り扱われます。多数のホストを指定するルールが必要ならば、これは有用です。以下の例は TCP Wrappers がすべての Telnet コネクションに対して /etc/telnet.hosts ファイルを参照します:

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

他にも、あまり使われないパターンも TCP Wrappers により受け付けられます。詳細は hosts_access マニュアル 5 ページを参照してください。

警告

ホスト名とドメイン名を使用するときは非常に注意してください。攻撃者は、正確な名前解決を避けるためにさまざまな技を使用できます。さらに、DNS サービスへの妨害により認可されたユーザーがネットワーク・サービスを使用することを妨害します。そのため、可能なときは必ず IP アドレスを使用するのが一番です。

3.6.2.1.3. Portmap と TCP Wrappers

Portmap の TCP Wrappers の実装は、ホスト名検索をサポートしません。このことは、portmap がホストを識別するためにホスト名を使えないことを意味します。結果として、hosts.allow や hosts.deny における portmap に対するアクセス制御ルールは、ホストを指定するために、IP アドレスを使用するか、キーワード ALL を使用しなければいけません。

portmap アクセス制御ルールへの変更はすぐに反映されないかもしれません。portmap サービスを再起動する必要があるかもしれません。

NIS や NFS のような広く使われるサービスは、動作するために portmap に依存します。そのため、これらの制限を意識してください。

3.6.2.1.4. 演算子

現在、アクセス制御ルールは1つの演算子 EXCEPT を受け付けます。ルールのデーモン・リストとクライアント・リストどちらも使用できます。

EXCEPT 演算子は、同じルールの中でより広くマッチさせるために、特定の例外を許可します。

hosts.allow ファイルからの以下の例は、すべての example.com ホストは、cracker.example.com は除き、すべてのサービスに接続を許可されます:

```
ALL: .example.com EXCEPT cracker.example.com
```

hosts.allow ファイルの他の例では、192.168.0.*x* ネットワークのクライアントは FTP を除くすべてのサービスを使用できます。

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

注記

組織的に、EXCEPT 演算子を使用することを避けることは、しばしばより簡単です。これにより、どのホストがサービスにアクセスを許可または拒否をされるかを見るために、EXCEPT 演算子をより分けることなく、他の管理者が適切なファイルを素早く検索できるようになります。

3.6.2.2. オプション・フィールド

アクセスを許可または拒否する基本的なルールに加えて、TCP Wrappers の Fedora 実装は、##########を通してアクセス制御言語への拡張をサポートします。hosts アクセス・ルールにおけるオプション・フィールドを使用することにより、ログ動作の変更、アクセス制御の統合、シェル・コマンドの実行などのさまざまな作業を管理者は達成することができます。

3.6.2.2.1. ログ取得

オプション・フィールドは、severity ディレクティブを使用することにより、管理者がルールに対するログ・ファシリティおよびプライオリティ・レベルをより簡単に変更できるようにします。

以下の例では、example.com ドメインにあるすべてのホストから SSH デーモンへの接続は、デフォルトの authpriv syslog ファシリティ（ファシリティ値が指定されていないため）にプライオリティ emerg で記録されます:

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

severity オプションを使用してファシリティを指定することも可能です。以下の例は、example.com ドメインのホストによるすべての SSH コネクション試行が local0 ファシリティに alert プライオリティで記録されます:

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

注記

実際には、syslog デーモン (syslogd) が local0 ファシリティを記録するよう設定されるまで、この例はうまく働きません。カスタムログ・ファシリティの設定に関する詳細は syslog.conf マニュアル・ページを参照してください。

3.6.2.2.2. アクセス制御

オプション・フィールドは管理者が、最後のオプションとして allow または deny ディレクティブを追加することにより、1つのルールにおいてホストの許可または拒否を明示的にできるようにすることができます。

たとえば、以下の2つのルールは、client-1.example.com からの SSH 接続を許可しますが、client-2.example.com からの接続は拒否します:

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

ルールごとを基本としたアクセス制御を許可することにより、オプション・フィールドは、管理者が1つのファイルの中ですべてのアクセス・ルールを統合できるようにします: hosts.allow または hosts.deny。何人かの管理者はこれがアクセス・ルールを編成する最も簡単な方法と考えます。

3.6.2.2.3. シェル・コマンド

オプション・フィールドはアクセス・ルールが以下の2つのディレクティブによりシェル・コマンドを起動できるようにします。

- spawn — 子プロセスとしてシェルコマンドを起動します。リクエストしているクライアントのより詳しい情報を得るために /usr/sbin/safe_finger を使用するようなタスクを実行できます、もしくはecho コマンドを用いて特別なログファイルを作成できます。

  以下の例では、example.com ドメインからの Telnet サービスにアクセスしようとしているクライアントは特別なファイルにひそかに記録されます:

  ```
  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特別なコマンドで置き換えます。このディレクティブはしばしば、侵入者に対するトラップ(「ハニーポット」とも呼ばれます)をセットアップするために使用されます。接続しているクライアントにメッセージを送るためにも使えます。twistディレクティブはルール行の最後に表れなければいけません。

  以下の例では、example.com ドメインからの FTP サービスへのアクセスを試みているクライアントは、echo コマンドを用いてメッセージを送られます:

  ```
  vsftpd : .example.com ¥
   : twist /bin/echo ”421 This domain has been black-listed. Access denied!”
  ```

シェル・コマンド・オプションの詳細は hosts_options マニュアル・ページを参照してください。

3.6.2.2.4. 拡張

拡張は、spawn および twist ディレクティブとともに使用されるとき、クライアント、サーバ、および関連する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リストです:

- %a — クライアントの IP アドレスを返します。
- %A — サーバの IP アドレスを返します。
- %c — ユーザ名ーとホスト名、またはユーザー名と IP アドレスのようなクライアントのさまざまな情報を返します。
- %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ければ 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ければ 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。もし利用できなければ、unknown が表示されます。クライアントのホスト名とホストのアドレスが一致しなければ、paranoid が表示されます。

- %N — サーバーのホスト名を返します。もし利用できなければ、unknown が表示されます。サーバーのホスト名とホストのアドレスが一致しなければ、paranoid 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・プロセスおよびサーバーのホストまたは IP アドレスのような、さまざまな種類のサーバーの情報を返します。
- %u — クライアントのユーザー名を返します。もし利用できなければ、unknown が表示されます。

以下のサンプル・ルールはカスタマイズされたログファイルにおいてクライアント・ホストを識別するために spawn コマンドとともに拡張を使用します。

SSH デーモン (sshd) へのコネクションが example.com ドメインにあるホストから試行されるとき、特別なファイルに (%h 表現を使用することにより) クライアントのホスト名を含めて、試行を記録するために echo コマンドを実行します。

```
sshd : .example.com  ¥
 : spawn /bin/echo `/bin/date` access denied to %h>>/var/log/sshd.log ¥
 : deny
```

同様に、拡張はクライアントに返すメッセージをカスタマイズするために使用できます。以下の例では、example.com ドメインから FTP サービスにアクセスを試行しているクライアントは、サーバから禁止されていることを通知されます。

```
vsftpd : .example.com ¥
: twist /bin/echo "421 %h has been banned from this server!"
```

利用可能な拡張の完全な説明、および追加のアクセス制御オプションは、hosts_access のマニュアル・ページのセクション5 (man 5 hosts_access) および hosts_options のマニュアル・ページを参照してください。

TCP Wrappers に関する詳細は ########## を参照してください。

3.6.3. xinetd

xinetd デーモンは、FTP, IMAP, および Telnet を含む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 ラップされた##########です。アクセス制御、高度なロギング、バインド、リダイレクト、およびリソース使用量制御に対する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 により制御されているネットワーク・サービスに接続しようとしているとき、スーパー・サービスがリクエストを受け取り、すべての TCP Wrappers アクセス制御ルールをチェックします。

アクセスが許可されると、xinetd はコネクションがそのサービスに対するそれ自身のアクセス・ルール下で許可されます。サービスがより多くのリソースを割り当てられ、すべての定義されたルールに違反していないこともチェックします。

これらの条件すべてが満たされた (つまり、サービスへのアクセスが許可され、サービスがそのリソース制限に届かず、そして、サービスが定義されたルールすべてに違反していない) ならば、xinetd はリクエストされたインスタンスを開始して、それへのコネクションの制御を認めます。コネクションが確立された後、xinetd は、クライアントとサーバ間のコミュニケーションにそれ以上参加しません。

3.6.4. 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 — 全体の xinetd 設定ファイル。

- /etc/xinetd.d/ — サービス固有のすべてのファイルを含むディレクトリ。

3.6.4.1. /etc/xinetd.conf ファイル

/etc/xinetd.conf ファイルは xinetd の制御下ですべてのサービスに影響する一般的な設定を含みます。xinetd サービスが最初に起動するときに読み込まれます。そのため、設定の変更を反映するためには、xinetd サービスを再起動する必要があります。以下は /etc/xinetd.conf ファイルのサンプルです:

```
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 の以下の観点を制御します:

- instances — xinetd が処理できる同時リクエストの最大数を指定します。
- log_type — xinetd が authpriv ログ・ファシリティを使用するよう設定します。それはログ・エントリーを /var/log/secure ファイルに書き込みます。FILE /var/log/xinetdlog のようなディレクティブを追加することにより、/var/log/ ディレクトリにある xinetdlogというカスタムログファイルを作成します。
- log_on_success — 成功したコネクション試行を記録するよう xinetd を編集します。デフォルトで、リクエストを処理するサーバのリモートホストの IP アドレスおよびプロセス ID が記録されます。
- log_on_failure — コネクションが拒否されると、失敗したコネクション試行を記録するために xinetd を設定します。
- cps — あらゆる与えられたサービスに1秒あたり25コネクションだけを許可するように xinetd を設定します。サービスが 30 秒間待たされます。
- includedir /etc/xinetd.d/ — /etc/xinetd.d/ ディレクトリにあるサービス固有の設定ファイルで宣言されたオプションを取り込みます。詳細は *#/etc/xinetd.d/ #######* を参照してください。

注記

しばしば、/etc/xinetd.conf にある log_on_success および log_on_failure の設定は、サービス固有の設定ファイルにおいてさらに修正されます。そのため、詳細は、/etc/xinetd.conf ファイルが示すところより、与えられたサービスのログファイルに表れるかもしれません。詳細は *########## を参照してください。

3.6.4.2. /etc/xinetd.d/ ディレクトリ

/etc/xinetd.d/ ディレクトリは xinetd により管理される各サービスに対する設定ファイルを含み、ファイルの名前はサービスと一致します。xinetd.conf にあるように、このディレクトリは xinetd サービスが起動するときのみ読み込まれます。あらゆる変更は効果を持たせるために、管理者が xinetd サービスを再起動しなければいけません。

/etc/xinetd.d/ ディレクトリにあるファイルのフォーマットは、/etc/xinetd.conf と同じ規約を使用します。各サービスに対する設定が別々のファイルに保存される一番の理由は、より簡単にカスタマイズでき、他のサービスに影響を与えないようにするためです。

これらのファイルがどのような構造であるかを理解するために、/etc/xinetd.d/krb5-telnet ファイルを検討します:

```
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 サービスをさまざまな観点で制御します:

- service — サービス名を指定します、通常 /etc/services ファイルにおいてリストされるものの1つです。
- flags — コネクションに対するいくつかの属性のどれかをセットします。REUSE は Telnet 接続に対するソケットを再利用するよう xinetd に指示します。

注記

REUSE フラグは廃止されました。現在、すべてのサービスは暗黙的に REUSE フラグを使用します。

- socket_type — ネットワーク・ソケットの種類を stream にセットします。
- 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) のどちらであるかを指定します。
- user — プロセスが実行されるユーザー ID を指定します。
- server — 起動するためにバイナリ実行可能なものを指定します。
- log_on_failure — xinetd.conf においてすでに定義されているものに加えて、log_on_failure に対するログのパラメータを指定します。
- disable — サービスが無効化 (yes) または有効化 (no) されるかを指定します。

これらのオプションとその使用法に関する詳細は xinetd.conf マニュアル・ページを参照してください。

3.6.4.3. xinetd 設定ファイルの変更

ディレクティブの範囲は xinetd により保護されたサービスに対して利用可能です。このセクションは、より一般的に使用されるオプションのいくつかにハイライトします。

3.6.4.3.1. ログ取得オプション

以下のロギング・オプションは /etc/xinetd.conf および /etc/xinetd.d/ ディレクトリにあるサービス固有の設定ファイルに対して利用可能です。

以下は、より一般的に使われるロギング・オプションのいくつかのリストです:

- ATTEMPT — 失敗した試行がなされたという事実を記録します (log_on_failure)。
- DURATION — サービスがリモート・システムにより使用された時間の長さを記録します (log_on_success)。

- EXIT — 終了ステータスまたはサービスの終了シグナルを記録します (log_on_success)。
- HOST — リモート・ホストの IP アドレスを記録します (log_on_failure および log_on_success)。
- PID — リクエストを受け取ったサーバのプロセス ID を記録します (log_on_success)。
- USERID — すべてのマルチ・スレッド stream サービスに対して RFC 1413 で定義された方式を使用してリモート・ユーザーを記録します (log_on_failure および log_on_success)。

ロギング・オプションの完全なリストは xinetd.conf マニュアル・ページを参照してください。

3.6.4.3.2. アクセス制御オプション

xinetd サービスのユーザーは、TCP Wrappers の hosts アクセスルールを使うことを選択する、xinetd 設定ファイル経由のアクセス制御を提供する、もしくは両方の混在をすることができます。TCP Wrappers hosts アクセス制御ファイルの詳細は *#TCP Wrappers ########* を参照してください。

このセクションはサービスへのアクセスを制御するために xinetd を使用することについて議論します。

注記

TCP Wrappers と違い、xinetd 管理者が xinetd サービスを再起動すると、アクセス制御の変更が効果を持ちます。

また、TCP Wrappers と違い、xinetd を通したアクセス制御は xinetd により制御されるサービスのみが効果を持ちます。

xinetd ホスト・アクセス制御は、TCP Wrappers により使われる方式とは異なります。TCP Wrappers は2つの設定ファイル /etc/hosts.allow および /etc/hosts.deny の¥n中ですべてのアクセス設定がされますが、xinetd のアクセス制御は /etc/xinetd.d/ ディレクトリにある各サービスの設定ファイルに見られます。

以下のホスト・アクセス・オプションは xinetd によりサポートされます:

- only_from — 指定されたホストのみがサービスを使用することを許可されます。
- no_access — リストされたホストがサービスを使用することをブロックされます。
- access_times — 特定のサービスが使用される可能性がある時間帯を指定します。時間帯は24時間表記 HH:MM-HH:MM で記載されなければいけません。

only_from と no_access オプションは、IP アドレスまたはホスト名のリストを使用できます。もしくは、ネットワーク全体を指定できます。TCP Wrappers のように、xinetd アクセス制御と高度なロギング設定を組み合わせることは、各コネクションの試行を冗長に記録しながら、禁止されたホストからのリクエストをブロックすることにより、セキュリティを向上させることができます。

たとえば、以下の /etc/xinetd.d/telnet ファイルは特定のネットワークグループからの Telnet アクセスを拒否して、許可されたユーザーがログインできる時間帯を制限できます:

```
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、172.16.45.2 のような 172.16.45.0/24 ネットワーク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き、以下のメッセージを受け取ります。

```
Connection closed by foreign host.
```

さらに、ログイン試行が以下のように /var/log/messages に記録されます:

```
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 Wrappers を使用するとき、2つのアクセス制御メカニズムの関係を理解することは重要です。

以下は、クライアントが接続を要求するとき、xinetd により実行される一連のイベントです。

1. xinetd デーモンは libwrap.a ライブラリコールを用いて TCP Wrappers hosts アクセスルールにアクセスします。拒否ルールがクライアントにマッチすると、コネクションは廃棄されます。許可ルールがクライアントにマッチすると、コネクションが xinetd に渡されます。

2. xinetd デーモンは、xinetd サービスおよびリクエストされたサービスどちらに対しても自身のアクセス制御ルールをチェックします。拒否ルールがクライアントにマッチすると、コネクションは廃棄されます。そうでなければ、xinetd はリクエストされたサービスのインスタンスを起動し、サービスへのコネクションを認めます。

重要

xinetd アクセス制御とともに TCP Wrappers を使用するときは注意する必要があります。設定誤りが意図しない効果を引き起こす可能性があります。

3.6.4.3.3. バインドとリダイレクトのオプション

xinetd の設定ファイルは、サービスの IP アドレスへのバインド、およびサービスの入力リクエストを他の IP アドレス、ホスト名、またはポートへのリダイレクトをサポートします。

バインドは、サービス固有の設定ファイルにおいて bind オプションを用いて制御され、サービスをシステムにおける1つの IP アドレスにリンクします。これが設定されるとき、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へのアクセスを許可されます。リクエストに基づいて、異なるネットワークインタフェースに異なるサービスをバインドするために、この方式を使用することができます。

これは複数のネットワークアダプタまたは複数の IP アドレスを持つシステムにとってとくに有用です。そのようなシステムにおいて、セキュアではないサービス（たとえば、Telnet）は、プライベート・ネットワークに接続されたインタフェースにおいてのみリッスンして、インターネットに接続されたインタフェースではそうしないよう設定できます。

redirect オプションは、ポート番号を後ろにつけた IP アドレスまたはホスト名を受け付けます。このサービスに対するすべてのリクエストを、指定されたホストとポート番号へとリダイレクトするよう、サービスを設定します。

同じシステムにある別のポート番号を指し示す、リクエストを同じマシンにある別の IP アドレスにリダイレクトする、リクエストを全体的に異なるシステムとポート番号に変換する、もしくはこれらのオプションすべての組み合わせをするためにこれらの機能を使用できます。それゆえ、システムにおける特定のサービスに接続しているユーザーは中断することなく他のシステムに再ルートされるかもしれません。

xinetd デーモンは、クライアントマシンと実際にサービスを提供するホストの間でコネクションの間中ずっと継続し続けるプロセスを生み出し、2つのシステム間でデータを転送することにより、このリダイレクトを達成できます。

bind および redirect オプションの利点は、一緒に使われたときに、最も明確にわかりやすいです。あるシステムにおいて特定の IP アドレスへとサービスをバインドして、このサービスに対するリクエストを1番目のマシンが見える2番目のマシンへとリダイレクトすることにより、内部システムが全体的に異なるネットワークに対してサービスを提供するために使用できます。代わりに、これらのオプションは、既知の IP アドレスへと複数ホームのマシンにおける特定のサービスの露出を制限して、そのサービスに対するすべてのリクエストをその目的のために特別に設定された他のマシンへとリダイレクトするためにも使用できます。

たとえば、システム Telnet サービスに対して、この設定を持つファイアウォールとして使用されることを考えます:

```
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
}
```

このファイルにある bind および redirect オプションはマシンにある Telnet サービスは外部 IP アドレス（インターネットに接しているもの）に結び付けらることを確実にします。加えて、123.123.123.123 に送られた Telnet サービスに対するすべてのリクエストは、2つ目のネットワーク・アダプターを経由して、ファイアウォールと内部システムだけがアクセスできる内部 IP アドレス (10.0.1.13) に送られます。そして、ファイアウォールを2つのシステム間で通信を送り、接続しているシステムは実際に別のマシンに接続しているとき、123.123.123.123 へと接続していると考えます。

ブロードバンド接続と固定 IP アドレスを1つだけ持つユーザーにとって、この機能はとくに有用です。Network Address Translation (NAT) を使用するとき、内部専用 IP アドレスを使用するゲートウェイマシンの後ろにあるシステムはゲートウェイシステムの外側から利用可能ではありません。しかしながら、xinetd により制御される特定のサービスが bind および redirect オプションを用いて設定されているとき、ゲートウェイマシンは、外側のシステムと、サービスを提供するよう設定された特定の内部マシンの間でプロキシとして動作できます。さらに、xinetd のアクセス制御およびロギングのさまざまなオプションがさらなる保護のために利用可能です。

3.6.4.3.4. リソース管理オプション

xinetd デーモンは Denial of Service (DoS) 攻撃から基本的なレベルの保護を与えられます。以下はそのような攻撃の有効性を制限するのに役立つディレクティブのリストです:

- per_source — ソース IP アドレスあたりのサービスに対するインスタンスの最大数を定義します。引数として整数のみを受け付け、xinetd.conf および xinetd.d/ ディレクトリにあるサービス固有の設定ファイルにおいて使用できます。

- cps — 秒あたりの最大コネクション数を定義します。このディレクティブは空白で区切られた2つの整数を受け付けます。1番目の引数は秒あたりにサービスに許可されたコネクションの最大数です。2番目の引数は xinetd がサービスを再び有効化するまでに待たなければいけない秒数です。引数として整数のみを受け付け、xinetd.conf および xinetd.d/ ディレクトリにあるサービス固有の設定ファイルにおいて使用できます。

- max_load — サービスに対する CPU 利用率またはロード・アベレージの閾値を定義します。浮動小数点の引数を受け付けます。

  ロード・アベレージはある時点においてどのくらいのサービスがアクティブであるかを大まかに測定する方法です。ロード・アベレージの詳細は uptime, who, および procinfo コマンドを参照してください。

xinetd に対して利用可能なより多くのリソース管理オプションがあります。詳細は xinetd.conf マニュアル・ページを参照してください。

3.6.5. 追加のリソース

TCP Wrappers と xinetd に関する詳細は、システムのドキュメントとインターネットにおいて入手可能です。

3.6.5.1. インストールされた TCP Wrappers ドキュメント

システムにあるドキュメントは、TCP Wrappers, xinetd, およびアクセス制御に対する、追加の設定オプションを探し始めるよい場所です。

- /usr/share/doc/tcp_wrappers-***<version>***/ — このディレクトリは README ファイルを含みます。これは、TCP Wrappers がどのように働き、さまざまなホスト名やホスト・アドレスのありえる偽装リスクについて議論しています。

- /usr/share/doc/xinetd-***<version>***/ — このディレクトリは README ファイルを含みます。これは、/etc/xinetd.d/ ディレクトリ¥nにあるサービス固有の設定ファイルを変更することに対するさまざまなアイディアとともに、アクセス制御や sample.conf ファイルの観点を議論しています。

- TCP Wrappers および xinetd に関連するマニュアル・ページ — TCP Wrappers および xinetd に関連するさまざまなアプリケーションや設定ファイルに対する多くのマニュアル・ページが存在します。以下はより重要なマニュアル・ページのいくつかです。

  サーバ・アプリケーション
  - man xinetd — xinetd のマニュアル・ページ

  設定ファイル
  - man 5 hosts_access — TCP Wrappers の hosts access 制御ファイルのマニュアル・ページ。
  - man hosts_options — TCP Wrappers オプション・フィールドのマニュアル・ページ。
  - man xinetd.conf — xinetd 設定オプションを一覧にしているマニュアル・ページ。

3.6.5.2. 有用な TCP Wrappers ウェブサイト

- *http://www.xinetd.org/*[3] — xinetd のホーム、サンプル設定ファイル、機能の完全な一覧、および有益な FAQ。

- *http://www.docstoc.com/docs/2133633/An-Unofficial-Xinetd-Tutorial* — 具体的なセキュリティ目標を達成するために、デフォルトの xinetd 設定ファイルを最適化する多くの異なる方法を議論する、完全なチュートリアル。

3.6.5.3. 関連書籍

- Brian Hatch, James Lee, および George Kurtz による *Hacking Linux Exposed*; Osbourne/McGraw-Hill — TCP Wrappers および xinetd に関する情報を持つ優れたセキュリティ・リソース。

[3] http://www.xinetd.org

3.7. Kerberos

ネットワークの中におけるシステムのセキュリティと完全性は扱いにくいです。どのサービスがネットワークにおいて実行されているか、これらのサービスがどのような方法で使用されているか、を追いかけ続けるために何人かの管理者の時間を消費します。

さらに、ネットワーク・サービスに認証しているユーザーは、従来の FTP や Telnet プロトコルを用いてネットワーク上に暗号化されないパスワードの転送により証明されるように、プロトコルにより使用されている方式が本質的にセキュアではないとき、危険であることを証明できます。

Kerberos は、危険な認証の方式を許可するプロトコルに対する必要性を取り除き、それによりネットワーク・セキュリティ全体を強化する方法です。

3.7.1. Kerberos とは何でしょうか?

Kerberos は MIT により作成されたネットワーク認証プロトコルです。そして、ネットワーク・サービスにユーザーを認証するために対象暗号鍵 [4] を使用します。これは、パスワードがネットワーク上で実際には決して送られないことを意味します。

したがって、ユーザーが Kerberos を使用してネットワーク・サービスに認証するとき、ネットワーク・トラフィックを監視することによりパスワードを集めようとしている認可されないユーザーは効果的に挫折させられます。

3.7.1.1. Kerberos の利点

多くの慣習的なネットワーク・サービスは、パスワード・ベースの認証スキームを使用します。そのようなスキームは、ユーザー名とパスワードを供給することにより、与えられたネットワーク・サーバーへと認証するためにユーザーに要求します。不幸にも、多くのサービスに対する認証情報の転送は暗号化されません。そのようなスキームをセキュアにするために、ネットワークは外部者からアクセス不能にしなければいけません。そして、ネットワークにあるすべてのコンピュータとユーザーが信頼され、信頼できなければいけません。

たとえこれが問題であるとしても、インターネットに接続されたネットワークはもはやセキュアであるとは見なされません。ネットワークへのアクセス権を得た攻撃者は、ユーザー・アカウントとセキュリティ基盤全体を危険にさらす、ユーザー名とパスワードを横取りするために、パケット・スニファーとしても知られるシンプルなパケット・アナライザーを使用できます。

Kerberos の一番の設計目標は、ネットワークを通した暗号化されないパスワードの転送を減らすことです。適切に使用されれば、Kerberos はパケット・スニファーがそうしないとネットワークに配置される脅威を効果的に減らします。

3.7.1.2. Kerberos の欠点

Kerberos は一般的かつ深刻なセキュリティ脅威を取り除きますが、さまざまな理由により導入することが難しいかもしれません:

- `/etc/passwd` や `/etc/shadow` のような標準的な UNIX パスワード・データベースから、Kerberos パスワード・データベースにユーザーのパスワードを移行することは、このタスクを実行する自動化されたメカニズムがないため、時間がかかる可能性があります。オンライン Kerberos FAQ の Question 2.23 を参照してください:

  http://www.nrl.navy.mil/CCS/people/kenh/kerberos-faq.html[5]

- Kerberos は多くの Fedora サーバにより使用される Pluggable Authentication Modules (PAM) システム¥nと部分的な互換性のみがあります。この問題の詳細は *#Kerberos # PAM#* を参照してください。

[4] ネットワーク通信を暗号化および復号するために使用される共通のキーをクライアントとサーバーが共有するシステム。
[5] http://www.nrl.navy.mil/CCS/people/kenh/kerberos-faq.html#pwconvert

- Kerberos は、それぞれのユーザーが信頼されますが、信頼されないネットワークにある信頼されないホストを使用します。その主要な目標は、暗号化されないパスワードがネットワークを越えて転送されるのを防ぐことです。しかしながら、適切なユーザー以外の誰かが認証のために使用されるチケットを発行する1つのホスト、キー配布センター (*KDC*: *key distribution center*) 、にアクセスするならば、Kerberos 認証システム全体がリスクにさらされます。

- Kerberosを利用するアプリケーションにとって、そのソースは Kerberos ライブラリの中にある適切なコールをするために、修正されなければいけません。この方法で修正されたアプリケーションは *Kerberos ##*, あるいは *kerberos ####*と考えられます。いくつかのアプリケーションに対して、これはアプリケーションの大きさやその設計のために極めて問題である可能性があります。他の互換性のないアプリケーションに対しては、変更はサーバーとクライアントがコミュニケートする方法にならなければいけません。さらにまた、これは広範囲なプログラミングを必要とします。デフォルトで Kereros に対応していないクローズ・ソースのアプリケーションはしばしば最も問題があります。

- Kerberos は全か無かのソリューションです。Kerberos がネットワークにおいて使用されるならば、Kerberos に対応していないサービスに転送される暗号化されないパスワードはすべてリスクになります。このように、ネットワークは Kerberos の使用から何も利益を得ません。Kerberos を用いてネットワークをセキュアにするために、暗号化されないパスワードを転送する###のクライアント/サーバー・アプリケーションの Kerberos 対応バージョンを使用する、もしくは、そのようなクライアント/サーバー・アプリケーションを###使用しないようにしなければいけません。

3.7.2. Kerberos の用語

Kerberos はサービスのさまざまな特徴を定義するためにそれ自身の用語を持ちます。Kerberos がどのように機能するかを学ぶ前に、以下の用語を学ぶことは重要です。

認証サーバ (AS: authentication server)
: 次々にユーザーにサービスへのアクセス権を与える専用のサービスに対するチケットを発行するサーバー。AS は、リクエストとともにクレデンシャルを持っていない、または送っていないクライアントからのリクエストに応答します。ticket-granting ticket (TGT) を発行することにより、ticket-granting server (TGS) サービスへのアクセス権を得るために一般的に使用されます。AS は一般的にキー配布センター (KDC) を同じホストにおいて実行します。

暗号文
: 暗号化されたデータ。

クライアント
: ネットワークにおいて Kerberos からチケットを受け取るエンティティ（ユーザー、ホストまたはアプリケーション）。

クレデンシャル
: 特定のサービスに対するクライアントのアイデンティティを確認する電子的なクレデンシャルの一時的なセット。チケットとも呼ばれます。

クレデンシャル・キャッシュまたはチケット・ファイル
: ユーザーとさまざまなネットワーク・サービスの間の暗号化されたコミュニケーションに対するキーを含むファイル。Kerberos 5 は共有メモリーのような他のキャッシュ形式を使用するためのフレームワークをサポートしますが、ファイルはより全体的にサポートされます。

暗号ハッシュ
: ユーザーを認証するために使われる一方向ハッシュ。暗号化されていないデータを使うよりはセキュアですが、経験のあるユーザーが復号することはまだ比較的易しいです。

GSS-API
: Generic Security Service Application Program Interface (Internet Engineering Task Force により発行された RFC-2743 で定義されます) は、セキュリティ・サービスを提供する一連の関数です。この API は、それぞれのプログラムが基礎となるメカニズムの具体的な知識なしでお互いを認証するために、クライアントとサービスにより使用されます。ネットワーク・サービス (cyrus-IMAPのような) が GSS-API を使用するならば、Kerberos を用いて認証できます。

ハッシュ
: #####としても知られます。######に文字列を渡すことにより生成された値。これらの値は一般的に、転送されたデータが改ざんされていないことを保証にするために使われます。

ハッシュ関数
: 入力データからデジタルな "フィンガープリント" を生成する方法。これらの関数は、#####を作成するために、データを再配置、転置、または他に変更します。

キー
: 他のデータを暗号化または複合するときに使われるデータ。暗号化されたデータは、正しいデータもしくはクラッカー側で極めて幸運がなければ複合できません。

キー配布センター (KDC: key distribution center)
: kerberos チケットを発行するサービス、また一般的に ticket-granting server (TGS) として同じホストにおいて実行されます。

keytab (またはキー・テーブル)
: プリンシパルとそのキーの暗号化されていないリストを含むファイル。サーバーは kinit を使用する代わりに keytab ファイルから必要とするキーを取得します。デフォルトの keytab ファイルは /etc/krb5.keytab です。KDC 管理サーバー /usr/kerberos/sbin/kadmind は、(/var/kerberos/krb5kdc/kadm5.keytab を使用する) 他のすべてのファイルを使用する唯一のサービスです。

kinit
: kinit コマンドは、すでにログインしたプリンシパルが、初期 TGT (ticket-granting ticket) を手に入れてキャッシュできるようにします。詳細は kinit マニュアル・ページを参照してください。

プリンシパル (またはプリンシパル名)
: プリンシパルは、Kerberos を使用する認証を許可されたユーザーまたはサービスの一意な名前です。プリンシパルは root[/instance]@REALM の形式に従います。一般的なユーザーに対して、root はログイン ID を同じです。instance はオプションです。プリンシパルがインスタンスを持つならば、スラッシュ ("/") を用いて root から分離されます。空の文字 ("") は (デフォルトの NULL インスタンスを異なる) 有効なインスタンスを見なされますが、それを使用することは混乱を招きます。レルムにあるすべてのプリンシパルは自分自身のキーを持ち、ユーザーに対してパスワードから導き出されるか、サービスに対してランダムにセットされます。

レルム
: Kerberos を使用するネットワーク。1つかそれより多い KDC と呼ばれるサーバー、および潜在的に多くのクライアントから構成されます。

サービス
: ネットワーク上でアクセスされるプログラム。

チケット
: 特定のサービスに対するクライアントのアイデンティティを確認する電子的なクレデンシャルの一時的なセット。クレディンシャルとも呼ばれます。

ticket-granting server (TGS)
: サービスにアクセスするためにユーザーへ交互に与えられる、希望するサービスに対してチケットを発行するサーバー。TGS は一般的に KDC と同じホストにおいて実行されます。

ticket-granting ticket (TGT)
: クライアントが KDC から適用されることなく追加のチケットを得られるようにする特別なチケット。

暗号化されていないパスワード
: プレイン・テキスト、人間が読めるパスワード。

3.7.3. Kerberos はどのように動作しますか

Kerberos はユーザー/パスワードの認証方式とは異なります。各ユーザーが各ネットワーク・サービスに認証する代わりに、ユーザーを一連のネットワーク・サービスに認証するために、Kerberos は対象鍵暗号と信頼された第三者 (KDC) を使用します。ユーザーが KDC に認証するとき、KDC はそのセッションに特有のチケットをユーザーのマシンに送り戻します。そして、すべての Kerberos 対応サービスは、ユーザーにパスワードを使用した認証よりも、ユーザーのマシンにおけるチケットを期待します。

Kerberos 対応のネットワークにいるユーザーが自身のワークステーションにログインするとき、プリンシパルがアプリケーション・サーバーからの TGT をリクエストの一部として KDC に送られます。このリクエストは、ユーザーへと透過的になるようにログイン・プログラムにより送られます。もしくは、ユーザーがログインした後に `kinit` により送られます。

その後、KDC はそのデータベースにあるプリンシパルに対してチェックします。プリンシパルが見つかると、KDC は TGT を作成します。それは、ユーザーのキーを使用して暗号化され、ユーザーへと返されます。

クライアントにあるログインまたは `kinit` プログラムはユーザーのキーを使用して TGT を復号します。そして、それはユーザーのパスワードから計算します。ユーザーのキーはクライアントマシンにおいてのみ使用され、ネットワーク上で転送され###

TGT は一定時間後（通常は10から24時間）に期限切れするようセットされ、クライアント・マシンのクレデンシャルに保存されます。漏えいした TGT が攻撃者に短い時間のみ使用されるよう、期限切れ時間がセットされます。TGT が発行された後、ユーザーは TGT が期限切れするまで、またはログアウトして再びログインするまでパスワードを再入力する必要はありません。

ユーザーがネットワーク・サービスにアクセスする必要があるときはいつでも、クライアント・ソフトウェアが TGS からその特定のサービスに対する新しいチケットを要求するために TGT を使用します。

警告

ネットワークにいるユーザーが平文でパスワードを転送することにより Kerberos に対応していないサービスに認証するならば、Kerberos システムは危険にさらされる可能性があります。Kerberos に対応していないサービスを使用することは高く思いとどまらせます。そのようなサービスは Telnet や FTP を含みます。しかしながら、SSH や SSL 化されたサービスのような他の暗号化プロトコルの使用は好まれますが、理想的ではありません。

これは Kerberos 認証がどのように機能するかの幅広い概要です。詳細については ######### を参照してください。

注記

Kerberos は正しく機能するために以下のネットワーク・サービスに依存します。

- ネットワークにあるマシン間でクロック同期を近づけます。

  クロック同期プログラムは ntpd のようにネットワークに対してセットアップされるべきです。Network Time Protocol サーバーのセットアップに関する詳細は /usr/share/doc/ntp-***<version-number>***/index.html を参照してください（ここで *<version-number>* は、システムにインストールされた ntp パッケージのバージョン番号です）。

- Domain Name Service (DNS)

  ネットワークにおける DNS エントリーと hosts がすべて正しく設定されていることを確実にすべきです。詳細は /usr/share/doc/krb5-server-***<version-number>*** にある *Kerberos V5 System Administrator's Guide* を参照してください（ここで *<version-number>* は、システムにインストールされた krb5-server パッケージのバージョン番号です）。

3.7.4. Kerberos と PAM

Kerberos 対応サービスは現在 Pluggable Authentication Modules (PAM) を使用しません — これらのサービスは完全に PAM を回避します。しかしながら、PAM を使用するアプリケーションは、pam_krb5 モジュール（pam_krb5 で提供されます）がインストールされていると、認証のために Kerberos を使用できます。pam_krb5 パッケージは、login や gdm のようなサービスがユーザーを認証するとともにそれらのパスワードを用いて初期クレデンシャルを得られるようにする、サンプル設定ファイルを含みます。ネットワーク・サービスへのアクセスが常に Kerberos 対応サービスまたは IMAP のような GSS-API を使用するサービスを用いて実行されるならば、ネットワークは相当に安全であると考えられます。

重要

管理者はユーザーが Kerberos パスワードを用いて多くのネットワーク・サービスに認証できないことに注意すべきです。これらのサービスにより使用される多くのプロトコルは、ネットワーク上でそれを送信して、Kerberos システムの利益を破壊する前にパスワードを暗号化しません。たとえば、ユーザーは Kerberos 認証のために使用するものと同じパスワードを用いて、Telnet サービスへと認証することが許可されるべきではありません。

3.7.5. Kerberos 5 サーバーの設定

Kerberos をセットアップするとき、まず KDC をインストールします。スレーブサーバーをセットアップする必要があれば、まずマスターをインストールします。

最初の Kerberos KDC を設定するために、これらの手順に従います:

1. Kerberos を設定する前に時刻同期と DNS がすべてのクライアントとサーバーマシンにおいて正しく機能していることを確実にします。Kerberos サーバーとそのクライアントの間の時刻同期については特に注意をします。サーバーとクライアントの間で時刻が5分よりもずれていると（これは Kerberos 5 で設定可能です）、Kerberos クライアントはサーバーに認証することができません。この時刻同期は攻撃者が正当なユーザーになりすますために古い Kerberos チケットを使用するのを防ぐために不可欠です。

Kerberos が使用されていないときでも、Network Time Protocol (NTP) 互換のクライアント/サーバー・ネットワークをセットアップすることが望ましいです。Fedora はこの目的のために ntp パッケージを含みます。Network Time Protocol サーバーをセットアップする方法に関する詳細は /usr/share/doc/ntp-**<*version-number*>**/index.html を (<*version-number*> はシステムにインストールされた ntp パッケージのバージョン番号です)、NTP に関する詳細は *http://www.ntp.org* を参照してください。

2. KDC を実行する専用のマシンにおいて krb5-libs, krb5-server, および krb5-workstation パッケージをインストールします。このマシンは非常にセキュアである必要があります — 可能ならば、KDC 以外のあらゆるサービスを実行すべきではありません。

3. レルム名とドメイン—レルム・マッピングを反映するために、/etc/krb5.conf および /var/kerberos/krb5kdc/kdc.conf 設定ファイルを編集します。シンプルなレルムは、*EXAMPLE.COM* と *example.com* を正しいドメイン名に置き換え、— 正しい形式において大文字と小文字を確実に保ってください — また、KDC を *kerberos.example.com* から Kerberos サーバーの名前に変えることにより構築できます。便宜上、すべてのレルム名は大文字で、すべての DNS ホスト名とドメイン名は小文字にします。これらの設定ファイルの形式に関する詳細はそれぞれのマニュアル・ページを参照してください。

4. シェル・プロンプトから kdb5_util ユーティリティを用いてデータベースを作成します:

```
/usr/sbin/kdb5_util create -s
```

create コマンドは Kerberos レルムのためのキーを保存するデータベースを作成します。-s スイッチはマスター・サーバー・キーが保存される *stash* ファイルの作成を強制します。キーを読み込むために存在する隠しファイルが存在しなければ、Kerberos サーバー (krb5kdc) は、起動するときに毎回ユーザーにマスター・サーバー・キー (キーを再生成するために使用されます) を要求します。

5. /var/kerberos/krb5kdc/kadm5.acl ファイルを編集します。このファイルは、どのプリンシパルが Kerberos データベースとそのアクセスレベルを持つかを決めるために kadmind により使用されます。多くの組織は1行でうまくやっていけます:

```
*/admin@EXAMPLE.COM  *
```

大抵のユーザーは、データベースにおいて単一プリンシパル (*NULL*、空、または *joe@EXAMPLE.COM* のようなインスタンス) により表現されます。この設定において、*admin* のインスタンスという2つ目のプリンシパルを持つユーザー (たとえば、*joe/admin@EXAMPLE.COM*) は、レルムの Kerberos データベース上でフルパワーを行使できます。

kadmind がサーバーにおいて起動された後、すべてのユーザーは、レルムにあるすべてのクライアントとサーバーにおいて kadmin を実行することによりそのサービスにアクセスできます。しかしながら、kadm5.acl ファイルにリストされたユーザーのみが、自身のパスワードを変更することを除いて、なんらかの方法でデータベースを変更できます。

注記

kadmin ユーティリティはネットワーク上で kadmind サーバーをコミュニケーションして、認証を処理するために Kerberos を使用します。その結果として、最初のプリンシパルは、それを管理するためにネットワーク上でサーバーに接続する前に、すでに存在しなければいけません。kadmin.local コマンドを用いて最初のプリンシパルを作成します。これは、KDC として同じホストにおいて使用されるための具体的に設定されたもので、認証のために Kerberos を使用しません。

最初のプリンシパルを作成するために、KDC ターミナルにおいて以下の kadmin.local コマンドを入力します:

```
/usr/kerberos/sbin/kadmin.local -q "addprinc username/admin"
```

6. 以下のコマンドを使用して Kerberos を起動します:

```
/sbin/service krb5kdc start
/sbin/service kadmin start
```

7. kadmin の中にある addprinc コマンドを使用してユーザーに対するプリンシパルを追加します。kadmin と kadmin.local は KDC に対するコマンドライン・インターフェースです。それ自体は、多くのコマンド — addprinc のような — が kadmin プログラムを起動した後で利用可能です。詳細は kadmin マニュアル・ページを参照してください。

8. KDC がチケットを発行していることを確認します。まず、チケットを取得して、それをクレデンシャル・キャッシュファイルに保存するために、kinit を実行します。次に、キャッシュにあるクレデンシャルのリストを表示するために klist を使用して、キャッシュおよびそれを含むクレデンシャルを廃棄するために kdestroy を使用します。

> **注記**
>
> デフォルトで、kinit は、同じシステムログインユーザー名（Kerberos サーバーではありません）を使用して認証をしようとします。そのユーザー名が Kerberos データベースにあるプリンシパルと一致しなければ、kinit はエラーメッセージを発行します。それが起きると、コマンドラインにおける引数として正しいプリンシパル名をとともに kinit を供給します (kinit ***<principal>***)。

これらの手順が完了すると、Kerberos サーバーは稼働可能になります。

3.7.6. Kerberos 5 クライアントの設定

Kerberos 5 クライアントをセットアップすることは、サーバーをセットアップするほどではありません。最低限、クライアント・パッケージをインストールして、各クライアントに適切な krb5.conf 設定ファイルを提供します。ssh と slogin はクライアントシステムにリモートでログインする方式を好む一方、デプロイするのにもう少し多くの設定変更を必要とするにも関わらず、Kerberos 化されたバージョンの rsh と rlogin はまだ利用可能です。

1. 時刻同期は Kerberos クライアントと KDC の間で適切であることを確実にします。詳細は *#Kerberos 5 ######## を参照してください。さらに、Kerberos クライアント・プログラムを設定する前に Kerberos クライアントにおいて DNS が適切に動作することを確認します。

2. すべてのクライアント・マシンにおいて krb5-libs および krb5-workstation パッケージをインストールします。各クライアントに対して 適切な /etc/krb5.conf ファイルを供給します（通常は KDC により使用される krb5.conf ファイルと同じです）。

3. レルムにあるワークステーションが ssh または Kerberos 化された rsh や rlogin を使用して接続するユーザーを認証するために Kerberos を使用できる前に、それ自身のホストプリンシパルを Kerberos データベースに持たなければいけません。sshd, kshd, および klogind サーバー・プログラムはすべて、### #のサービスのプリンシパルに対するキーにアクセスする必要があります。加えて、Kerberos 化された rsh と rlogin を使用するために、そのワークステーションは xinetd パッケージがインストールされていなければいけません。

kadmin を使用すると、KDC におけるワークステーションに対するホストプリンシパルが追加されます。このケースにおけるインスタンスはワークステーションのホスト名です。プリンシパルを作成して、それにランダムなキーを割り当てるために、kadmin の addprinc コマンドに対して -randkey オプションを使用します:

```
addprinc -randkey host/blah.example.com
```

これでプリンシパルが作成されたので、キーは ################ kadmin を実行して、kadmin を用いて ktadd コマンドを使用することによりワークステーションのために抽出されます:

```
ktadd -k /etc/krb5.keytab host/blah.example.com
```

4. 他の Kerberos 化されたネットワーク・サービスを使用するためには、まずそれらが起動されていなければいけません。以下は、一般的な Kerberos 化されたサービスとそれらを有効にすることに関する説明の一覧です:

   - ssh — クライアントとサーバーの設定がどちらも GSSAPIAuthentication を有効にしているならば、OpenSSH はユーザーをサーバーへ認証するために GSS-API を使用します。クライアントが GSSAPIDelegateCredentials も有効にしていると、ユーザの証明書がリモート・システムにおいて利用可能になります。

   - rsh および rlogin — Kerberos 化されたバージョンの rsh および rlogin を使用するために、klogin, eklogin, および kshell を有効にします。

   - Telnet — Kerberos 化された Telnet を使用するために、krb5-telnet が有効にされなければいけません。

   - FTP — FTP アクセスを提供するために、ftp の root とともにプリンシパルに対するキーを作成および解凍する必要があります。インスタンスに FTP サーバーの完全修飾ホスト名を確実にセットしてください、そして gssftp を有効にします。

   - IMAP — Kerberos 化された IMAP サーバーを使用するために、cyrus-sasl-gssapi パッケージもインストールされているならば、cyrus-imap パッケージは Kerberos 5 を使用します。cyrus-sasl-gssapi パッケージは GSS-API 認証をサポートする Cyrus SASL プラグインを含みます。Cyrus IMAP は cyrus ユーザーが /etc/krb5.keytab に適切なキーを見つけられ、プリンシパルに対する root が imap (kadmin を用いて作成されます) にセットされる限り、Kerberos を用いて適切に機能すべきです。

     cyrus-imap の代替は、Fedora にも含まれる dovecot パッケージで見つけられます。このパッケージは IMAP サーバーを含みますが、現在まで GSS-API と Kerberos をサポートしていません。

   - CVS — Kerberos 化された CVS サーバーを使用するために、gserver は cvs の root とともにプリンシパルを使用します。そうでなければ、CVS pserver を同一です。

3.7.7. ドメイン−レルムのマッピング

クライアントが特定のサーバーで実行しているサービスにアクセスしようとするとき、サービスの名前(*host*) とサーバーの名前 (*foo.example.com*) を知ります。しかし、1つより多いレルムがネットワークにデプロイされているかもしれないので、サービスが存在するレルムの名前で推測しなければいけません。

レルムの名前はデフォルトで、サーバーの DNS ドメイン名が大文字で使用されます。

```
foo.example.org → EXAMPLE.ORG
 foo.example.com → EXAMPLE.COM
 foo.hq.example.com → HQ.EXAMPLE.COM
```

いくつかの設定において、これは十分ですが、他では導かれたレルム名は存在しないレルムの名前でしょう。これらの場合、サーバーの DNS ドメイン名からそのレルム名へのマッピングが、クライアントシステムの krb5.conf の *domain_realm* セクションにおいて指定されなければいけません。たとえば:

```
[domain_realm]
.example.com = EXAMPLE.COM
example.com = EXAMPLE.COM
```

上の設定は2つのマッピングを指定します。最初のマッピングは "example.com" DNS ドメインにあるすべてのシステムが *EXAMPLE.COM* レルムに所属するということを指定します。2つ目は正確に "example.com" という名前を持つシステムもレルムにあることを指定します。(ドメインと具体的なホストの区別は最初の "." の有無により区別されます。) マッピングは DNS にも直接保存されます。

3.7.8. セカンダリ KDC のセットアップ

多くの理由のため、与えられたレルムに対して複数の KDC を実行することを選択するかもしれません。このシナリオでは、1つの KDC (#### *KDC*) がレルム・データベースの書き込み可能なコピーを維持して、kadmind を実行します (それはレルムの ######でもあります)。また、1つかそれより多い KDC (#### *KDC*) はデータベースの読み込み専用のコピーを維持して、kpropd を実行します。

マスター—スレーブの伝搬手順は、マスター KDC がそのデータベースを一時的なダンプファイルにダンプして、そのファイルを各スレーブに転送するようにします。これは、それらの以前に受け取ったデータベース読み込み専用コピーをダンプファイルの内容で上書きします。

スレーブ KDC をセットアップするために、マスター KDC の krb5.conf および kdc.conf ファイルがスレーブ KDC に確実にコピーします。

マスター KDC において root シェルで kadmin.local を起動して、マスター KDC の *host* サービスに対する新しいエントリーを作成するために、その add_principal コマンドを使用します。そして、同時にサービスに対するランダムなキーをセットして、ランダムキーをマスターのデフォルト keytab ファイルに保存するために、その ktadd を使用します。このキーはスレーブサーバーを認証するために kprop コマンドにより使用されます。どのくらいのスレーブサーバーをインストールするかに関わらず、これを一度だけ実行する必要があります。

```
# kadmin.local -r EXAMPLE.COM

Authenticating as principal root/admin@EXAMPLE.COM with password.

kadmin: add_principal -randkey host/masterkdc.example.com

Principal ”host/host/masterkdc.example.com@EXAMPLE.COM” created.

kadmin: ktadd host/masterkdc.example.com

Entry for principal host/masterkdc.example.com with kvno 3, encryption type Triple DES cbc mode with HMAC/sha1
 added to keytab WRFILE:/etc/krb5.keytab.

Entry for principal host/masterkdc.example.com with kvno 3, encryption type ArcFour with HMAC/md5 added to
 keytab WRFILE:/etc/krb5.keytab.

Entry for principal host/masterkdc.example.com with kvno 3, encryption type DES with HMAC/sha1 added to keytab
 WRFILE:/etc/krb5.keytab.

Entry for principal host/masterkdc.example.com with kvno 3, encryption type DES cbc mode with RSA-MD5 added to
 keytab WRFILE:/etc/krb5.keytab.

kadmin: quit
```

スレーブ KDC において root シェルから kadmin を起動して、スレーブ KDC の *host* サービスに対する新しいエントリーを作成するために、その add_principal コマンドを使用します。そして、同時にサービスに対するラ

ンダムなキーをセットして、ランダムキーをスレーブのデフォルト keytab ファイルに保存するために、kadmin の ktadd を使用します。このキーはクライアントを認証するときに kpropd サービスにより使用されます。

```
# kadmin -p jimbo/admin@EXAMPLE.COM -r EXAMPLE.COM

Authenticating as principal jimbo/admin@EXAMPLE.COM with password.

Password for jimbo/admin@EXAMPLE.COM:

kadmin: add_principal -randkey host/slavekdc.example.com

Principal "host/slavekdc.example.com@EXAMPLE.COM" created.

kadmin: ktadd host/slavekdc.example.com@EXAMPLE.COM

Entry for principal host/slavekdc.example.com with kvno 3, encryption type Triple DES cbc mode with HMAC/sha1
 added to keytab WRFILE:/etc/krb5.keytab.

Entry for principal host/slavekdc.example.com with kvno 3, encryption type ArcFour with HMAC/md5 added to
 keytab WRFILE:/etc/krb5.keytab.

Entry for principal host/slavekdc.example.com with kvno 3, encryption type DES with HMAC/sha1 added to keytab
 WRFILE:/etc/krb5.keytab.

Entry for principal host/slavekdc.example.com with kvno 3, encryption type DES cbc mode with RSA-MD5 added to
 keytab WRFILE:/etc/krb5.keytab.

kadmin: quit
```

そのサービスキーを用いると、スレーブ KDC はそれに接続するすべてのクライアントを認証できます。明らかに、それらのすべてが新しいレルム・データベースを持つスレーブの kprop サービスを提供することが許可されるわけではありません。アクセスを制限するために、スレーブ KDC における kprop サービスは、/var/kerberos/krb5kdc/kpropd.acl にリストされたプリンシパル名であるクライアントからの更新のみを受け付けます。マスター KDC の host サービスの名前をそのファイルに追加します。

```
# echo host/masterkdc.example.com@EXAMPLE.COM > /var/kerberos/krb5kdc/kpropd.acl
```

一度スレーブ KDC がデータベースのコピーを取得すると、それを暗号化するために使用されるマスターキーが必要になります。KDC データベースのマスターキーが、マスター KDC (一般的に /var/kerberos/krb5kdc/.k5.REALM という名前) における *stash* ファイルに保存されると、利用可能なセキュアなあらゆる方法を用いてスレーブ KDC にコピーするか、ダミーのデータベースを作成して、kdb5_util create -s を実行して、同じパスワードを供給することによりスレーブ KDC に同一の stash ファイルを作成するかします。

スレーブ KDC のファイアウォールはマスター KDC がポート 754 の TCP を使用して接続できるようにしていることを確実にして、kprop サービスを起動します。そして、kadmin サービスが##にされていることを二重チェックします。

今、マスター KDC においてレルム・データベースを、kprop コマンドが読み込むデフォルトのデータファイル (/var/kerberos/krb5kdc/slave_datatrans) に、ダンプすることにより、手動のデータベース伝搬テストを実行します。そして、その内容をスレーブ KDC に転送するために kprop コマンドを使用します。

```
# /usr/sbin/kdb5_util dump /var/kerberos/krb5kdc/slave_datatrans# kprop slavekdc.example.com
```

kinit を使用すると、クライアントシステムの krb5.conf があなたのレルムに対して KDC のリストにあるスレーブ KDC のみリストしているものは、スレーブ KDC から初期クレデンシャルを正しく得られることを確認します。

単にレルム・データベースをダンプするスクリプトを作成して、データベースを各スレーブ KDC に順番に転送するために kprop コマンドを実行します。そして、定期的にスクリプトを実行するために cron サービスを設定します。

3.7.9. クロス・レルム認証のセットアップ

##########は、それらの自身以外のレルムが属するサービス(一般的に特定のサーバーシステムにおいて実行しているサーバープロセス)を認証するために、あるレルムのクライアント(一般的にユーザー)が Kerberos を使用する状況を記述するために使用される言葉です。

最も簡単な場合に対して、A.EXAMPLE.COM という名前のレルムのクライアントが B.EXAMPLE.COM レルムにあるサービスにアクセスするために、両方のレルムが krbtgt/B.EXAMPLE.COM@A.EXAMPLE.COM という名前のプリンシパルに対するキーを共有しなければならず、両方のキーがそれらに関連づけられた同じキーバージョン番号を持たなければいけません。

これを達成するために、非常に強いパスワードまたはパスフレーズを選択して、kadmin により使用される両方のレルムにおけるプリンシパルに対するエントリーを作成します。

```
 # kadmin -r A.EXAMPLE.COM kadmin: add_principal krbtgt/B.EXAMPLE.COM@A.EXAMPLE.COM Enter password
 for principal "krbtgt/B.EXAMPLE.COM@A.EXAMPLE.COM": Re-enter password for principal "krbtgt/
B.EXAMPLE.COM@A.EXAMPLE.COM": Principal "krbtgt/B.EXAMPLE.COM@A.EXAMPLE.COM" created. quit # kadmin -r
 B.EXAMPLE.COM kadmin: add_principal krbtgt/B.EXAMPLE.COM@A.EXAMPLE.COM Enter password for principal "krbtgt/
B.EXAMPLE.COM@A.EXAMPLE.COM": Re-enter password for principal "krbtgt/B.EXAMPLE.COM@A.EXAMPLE.COM": Principal
 "krbtgt/B.EXAMPLE.COM@A.EXAMPLE.COM" created. quit
```

両方のエントリーが対応するキー・バージョン番号 (kvno 値) と暗号化の種類を持つことを検証するために、get_principal コマンドを使用します。

データベースをダンプすることを実行しないでください。

セキュリティに注意深い管理者は、パスワードの代わりにランダムなキーを割り当てるために add_principal コマンドの -randkey オプションを使用して、最初のレルムのデータベースから新しいエントリーをダンプして、そしてそれを2番目にインポートしようとするかもしれません。データベースに含まれるキーがマスターキーを用いて暗号化されたそれ自身なので、これはレルム・データベースに対するマスターキーが同一でなければうまく動きません。

これで A.EXAMPLE.COM レルムにあるクライアントは B.EXAMPLE.COM レルムにあるサービスに認証できます。言い換えると、これで B.EXAMPLE.COM レルムは A.EXAMPLE.COM レルムを##します、もしくは、よりシンプルに言うと、B.EXAMPLE.COM は A.EXAMPLE.COM を##します。

これは重要な点をもたらします: クロス・レルム認証はデフォルトで一方向性です。B.EXAMPLE.COM レルムに対する KDC は、B.EXAMPLE.COM レルムにあるサービスに認証するために A.EXAMPLE.COM からのクライアントを信頼するかもしれません。しかし、B.EXAMPLE.COM レルムにクライアントがあってもなくても効果を持たないという事実は A.EXAMPLE.COM レルムにあるサービスに認証するために信頼されます。他の方向に信頼を確立するために、両方のレルムが krbtgt/A.EXAMPLE.COM@B.EXAMPLE.COM サービスに対するキーを共有する必要があります(上の例と比較して、2つのレルムの順番を反対にすることに注意してください)。

直接の信頼関係がレルム間の信頼を提供する唯一の方法であるならば、複数のレルムを含むネットワークはセットアップすることが非常に難しいです。幸運なことに、クロス・レルム認証は推移的です。A.EXAMPLE.COM からのクライアントが B.EXAMPLE.COM にあるサービスに認証でき、B.EXAMPLE.COM からのクライアントが C.EXAMPLE.COM にあるサービスに認証できるならば、***C.EXAMPLE.COM*** ### ***A.EXAMPLE.COM*** を信頼していなくても、A.EXAMPLE.COM にあるクライアントは C.EXAMPLE.COM にあるサービスにも認証できます。これは次の

ことを意味します。お互いにすべてを認証する必要がある、複数のレルムを持つネットワークにおいて、どの信頼関係をセットアップするかについて良い選択をすることは、必要とされる努力の量を非常に減らすことができます。

ここでより伝統的な問題に直面します: クライアントのシステムは、特定のサービスが属するレルムを適切に導き出されるように設定されなければいけません。また、そのレルムにあるサービスに対するクレデンシャルを取得する方法を決められなければいけません。

まず第一に: 与えられたレルムにおいて特定のサーバーシステムから提供されるサービスに対するプリンシパル名は、一般的にこのように見えます:

```
service/server.example.com@EXAMPLE.COM
```

この例において、*service* は一般的に、使用するプロトコルの名前（他の一般的な値は *ldap*, *imap*, *cvs*, および *HTTP* を含みます）もしくは *host* を使用します。*server.example.com* はサービスを実行しているシステムの完全修飾ドメイン名（FQDN）です。また、EXAMPLE.COM はレルムの名前です。

サービスが属するレルムを導き出すために、クライアントはしばしば、ホスト名 (*server.example.com*) または DNS ドメイン名 (*.example.com*) をレルム名 (*EXAMPLE.COM*) に対応付けるために、DNS または /etc/krb5.conf のdomain_realm セクションを参照します。

サービスがどのレルムに属するかを決めると、サービスに認証することに使用するためのクレデンシャルを得るために、クライアントはコンタクトする必要があるレルムの組を、またどの順番でコンタクトしなければいけないかを決めなければいけません。

これは2つの方法の内1つで実行されます。

明示的な設定を必要としないデフォルトの方式は、共有された階層の中でレルム名を与えることです。例として、A.EXAMPLE.COM, B.EXAMPLE.COM, および EXAMPLE.COM という名前のレルムを考えます。A.EXAMPLE.COM レルムにあるクライアントが B.EXAMPLE.COM にあるサービスに認証しようとするとき、デフォルトでまず EXAMPLE.COM レルムに対するクレデンシャルを取得しようとします。そして、B.EXAMPLE.COM レルムにおいて使用するためのクレデンシャルを取得するためにそれらのクレデンシャルを使用しようとします。

このシナリオにおけるクライアントは、あるものが DNS 名を取り扱っているかのようにレルム名を取り扱います。サービスのレルムの "上" でもあるポイントにたどり着くまで、階層においてそれの "上" であるレルムの名前を生成するために、それ自身のレルムの名前のコンポーネントを繰り返し取り除きます。その時点で、サービスのレルムにたどり着くまでサービスのレルム名のコンポーネントを先頭につけるもので始めます。プロセスに関連する各レルムは他の "ホップ" です。

たとえば、A.EXAMPLE.COM にあるクレデンシャルを使用して、B.EXAMPLE.COM にあるサービスを認証する方法¥nA.EXAMPLE.COM → EXAMPLE.COM → B.EXAMPLE.COM

- A.EXAMPLE.COM と EXAMPLE.COM は krbtgt/EXAMPLE.COM@A.EXAMPLE.COM に対するキーを共有します
- EXAMPLE.COM と B.EXAMPLE.COM は krbtgt/B.EXAMPLE.COM@EXAMPLE.COM に対するキーを共有します

もう1つの例は、SITE1.SALES.EXAMPLE.COM にあるクレデンシャルを使用して、EVERYWHERE.EXAMPLE.COM にあるサービスを認証する方法¥nSITE1.SALES.EXAMPLE.COM → SALES.EXAMPLE.COM → EXAMPLE.COM → EVERYWHERE.EXAMPLE.COM

- SITE1.SALES.EXAMPLE.COM と SALES.EXAMPLE.COM は krbtgt/SALES.EXAMPLE.COM@SITE1.SALES.EXAMPLE.COM に対するキーを共有します
- SALES.EXAMPLE.COM と EXAMPLE.COM は krbtgt/EXAMPLE.COM@SALES.EXAMPLE.COM に対するキーを共有します
- EXAMPLE.COM と EVERYWHERE.EXAMPLE.COM は krbtgt/EVERYWHERE.EXAMPLE.COM@EXAMPLE.COM に対するキーを共有します

もう一つの例は、その名前が共通のサフィックスを共有しないレルム名を使用するよう、これが調整します (DEVEL.EXAMPLE.COM および PROD.EXAMPLE.ORG DEVEL.EXAMPLE.COM → EXAMPLE.COM → COM → ORG → EXAMPLE.ORG → PROD.EXAMPLE.ORG

- DEVEL.EXAMPLE.COM と EXAMPLE.COM は krbtgt/EXAMPLE.COM@DEVEL.EXAMPLE.COM に対するキーを共有します
- EXAMPLE.COM および COM は krbtgt/COM@EXAMPLE.COM に対するキーを共有します
- COM および ORG は krbtgt/ORG@COM に対するキーを共有します
- ORG および EXAMPLE.ORG は krbtgt/EXAMPLE.ORG@ORG に対するキーを共有します
- EXAMPLE.ORG および PROD.EXAMPLE.ORG は krbtgt/PROD.EXAMPLE.ORG@EXAMPLE.ORG に対するキーを共有します

より複雑でより柔軟な方法は、あるレルムに対するクレデンシャルを持つクライアントがどのレルムがサーバーに認証できることに導くチェインの次にあるかを探すことができるように、/etc/krb5.conf の capaths セクションを設定することに関連します。

capaths セクションの形式は比較的素直です: セクションの各エントリーはクライアントが存在するかもしれないレルムの後ろに名前がつけられます。このサブセクションの中で、クライアントがクレデンシャルを得なければいけない中間レルムのセットは、サービスが存在するかもしれないレルムと対応するキーの値としてリストされます。もし中間レルムがなければ、値 "." が使用されます。

これは例です:

```
[capaths]
A.EXAMPLE.COM = {
B.EXAMPLE.COM = .
C.EXAMPLE.COM = B.EXAMPLE.COM
D.EXAMPLE.COM = B.EXAMPLE.COM
D.EXAMPLE.COM = C.EXAMPLE.COM
}
```

この例において、A.EXAMPLE.COM レルムにあるクライアントは、B.EXAMPLE.COM に対するクレデンシャルを直接 A.EXAMPLE.COM KDC からクロス・レルム・クレデンシャルを得ることができます。

それらのクライアントが C.EXAMPLE.COM レルムにあるサービスに問い合わせしたければ、まず B.EXAMPLE.COM レルムから必要なクレデンシャルを取得する必要があります(これは krbtgt/B.EXAMPLE.COM@A.EXAMPLE.COM が存在する必要があります)。そして、(krbtgt/C.EXAMPLE.COM@B.EXAMPLE.COM を使用して)C.EXAMPLE.COM レルムにおいて使用するためのクレデンシャルを取得するために、それらのクレデンシャルを使用します。

それらのクライアントが D.EXAMPLE.COM レルムにあるサービスに問い合わせたいならば、最終的に D.EXAMPLE.COM レルムを使用するためにクレデンシャルを得る前に、まず B.EXAMPLE.COM レルムから必要なクレデンシャルを、次に C.EXAMPLE.COM レルムからクレデンシャルを手に入れる必要があります。

注記

他の方法で capth エントリーを表示していなければ、Kerberos はクロス・レルム信頼関係が階層をなすと仮定します。

A.EXAMPLE.COM レルムにあるクライアントは B.EXAMPLE.COM レルムから直接クロスドメイン・クレデンシャルを得ることができます。これを意味する "." がなければ、クライアントは階層的なパスを使用するために代わりの試みをします。今回の場合:

```
A.EXAMPLE.COM → EXAMPLE.COM → B.EXAMPLE.COM
```

3.7.10. 追加のリソース

Kerberos に関する詳細は、以下のリソースを参照してください。

3.7.10.1. インストールされた Kerberos ドキュメント

- PostScript 形式および HTML 形式の *Kerberos V5 Installation Guide* および *Kerberos V5 System Administrator's Guide*。これらのドキュメントは `/usr/share/doc/krb5-server-<version-number>/` ディレクトリで見つけられます (ここで `<version-number>` はシステムにインストールされた `krb5-server` パッケージのバージョン番号です) 。

- PostScript および HTML 形式の *Kerberos V5 UNIX User's Guide*。これらは `/usr/share/doc/krb5-workstation-<version-number>/` ディレクトリ (ここで `<version-number>` はシステムにインストールされた `krb5-workstation` パッケージのバージョン番号です) で見つけられます。

- Kerberos マニュアル・ページ — Kerberos 実装と関連したさまざまなアプリケーションと設定ファイルに対する数多くのマニュアル・ページがあります。以下はより重要なマニュアル・ページのいくつかのリストです。

  クライアント・アプリケーション
  - `man kerberos` — どのようにクレデンシャルが機能して、Kerberos チケットを取得および廃棄するための推奨値を提供するかについて説明する、Kerberos システムへの導入。マニュアル・ページの最後に関連するマニュアル・ページの番号を参照します。
  - `man kinit` — ticket-granting ticket を取得およびキャッシュするためにこのコマンドを使用する方法について説明します。
  - `man kdestroy` — Kerberos クレデンシャルを廃棄するためにこのコマンドを使用する方法について説明します。
  - `man klist` — Kerberos キャッシュsれたクレデンシャルを表示するためにこのコマンドを使用する方法について説明します。

  管理アプリケーション
  - `man kadmin` — Kerberos V5 データベースを管理するためにこのコマンドを使用する方法について説明します。
  - `man kdb5_util` — Kerberos V5 データベースにおける低レベルの管理機能を作成および実行するためにこのコマンドを使用する方法について説明します。

サーバー・アプリケーション

- `man krb5kdc` — Kerberos V5 KDC に対して利用可能なコマンドライン・オプションを説明します。
- `man kadmind` — V5 管理サーバーに対して利用可能なコマンドライン・オプションを説明します。

設定ファイル

- `man krb5.conf` — Kerberos V5 ライブラリの設定ファイルにおける形式および利用可能なオプションを説明します。
- `man kdc.conf` — Kerberos V5 AS および KDC の設定ファイルにおける形式および利用可能なオプションを説明します。

3.7.10.2. 有用な Kerberos

- *http://web.mit.edu/kerberos/www/* — MIT の *Kerberos: The Network Authentication Protocol* ウェブページ。
- *http://www.nrl.navy.mil/CCS/people/kenh/kerberos-faq.html* — Kerberos の FAQ (Frequently Asked Questions)。
- *ftp://athena-dist.mit.edu/pub/kerberos/doc/usenix.PS* — Jennifer G. Steiner, Clifford Neuman, および Jeffrey I. Schiller による *Kerberos: An Authentication Service for Open Network Systems* の PostScript バージョン。このドキュメントは Kerberos を説明している原論文です。
- *http://web.mit.edu/kerberos/www/dialogue.html* — *Designing an Authentication System: a Dialogue in Four Scenes*、元々1988年 Bill Bryant により、1997年に Theodore Ts'o によります。このドキュメントは、Kerberos 形式の認証システムについて考え抜いている開発者2人の間の会話です。議論の会話形式は、Kerberos に完全になじみがない人々にとって素晴らしい開始地点になります。
- *http://www.ornl.gov/~jar/HowToKerb.html* — *How to Kerberize your site* はネットワークを Kerberos 化するための素晴らしい参考資料です。
- *http://www.networkcomputing.com/netdesign/kerb1.html* — *Kerberos Network Design Manual* は Kerberos システムの完全な概要です。

3.8. Using Firewalls

3.8.1. Introduction to firewalld

The dynamic firewall daemon `firewalld` provides a dynamically managed firewall with support for network zones to assign a level of trust to a network and its associated connections and interfaces. It has support for IPv4 and IPv6 firewall settings. It supports Ethernet bridges and has a separation of runtime and permanent configuration options. It also has an interface for services or applications to add firewall rules directly.

3.8.2. Understanding firewalld

A graphical configuration tool, firewall-config, is used to configure `firewalld`, which in turn uses iptables tool to communicate with Netfilter in the kernel which implements packet filtering.

To use the graphical firewall-config tool, press the super key and start typing `firewall`. The firewall icon will appear. Press enter once it is highlighted. The firewall-config tool appears. You will be prompted for your user password.

The firewall-config tool has drop a down selection menu labeled Current View. This enables selecting between Runtime Configuration and Permanent Configuration mode. Notice that if you select Permanent Configuration, an Edit Services button appears on the right hand side of the Services tab and an Edit ICMP Types button appears on the right hand side of the ICMP Filter tab. The reason these buttons only appear in permanent configuration mode is that runtime changes are limited to enabling or disabling a service. You cannot change a service's parameters in run time mode.

The firewall service provided by `firewalld` is dynamic rather than static because changes to the configuration can be made at anytime and are immediately implemented, there is no need to save or apply the changes. No unintended disruption of existing network connections occurs as no part of the firewall has to be reloaded.

There is also an applet, firewall-applet, which can be used to quickly launch the NetworkManager configuration tab for the network connection in use. From the General tab changes to the assigned firewall zone can be made. This applet is not installed by default in Fedora.

A command line client, firewall-cmd, is provided. It can be used to make permanent and non-permanent run-time changes as explained in `man firewall-cmd(1)`. Permanent changes need to be made as explained in `man firewalld(1)`.

The configuration for `firewalld` is stored in various XML files in `/usr/lib/firewalld/` and `/etc/firewalld/`. This allows a great deal of flexibility as the files can be edited, written to, backed up, used as templates for other installations and so on.

Other applications can communicate with `firewalld` using D-bus.

3.8.3. Comparison of Firewalld to system-config-firewall and iptables

The essential differences between `firewalld` and the iptables service are:

- The iptables service stores configuration in `/etc/sysconfig/iptables` while `firewalld` stores it in various XML files in `/usr/lib/firewalld/` and `/etc/firewalld/`. Note that the `/etc/sysconfig/iptables` file does not exist as `firewalld` is installed be default on Fedora.

- With the iptables service, every single change means flushing all the old rules and reading all the new rules from `/etc/sysconfig/iptables` while with `firewalld` there is no re-creating of all the rules; only the differences are applied. Consequently, `firewalld` can change the settings during run time without existing connections being lost.

Both use iptables tool to talk to the kernel packet filter.

3.8.4. Understanding Network Zones

Firewalls can be used to separate networks into different zones based on the level of trust the user has decided to place on the devices and traffic within that network. NetworkManager informs `firewalld` to which zone an interface belongs. An interface's assigned zone can be changed by NetworkManager or via the firewall-config tool which can open the relevant NetworkManager window for you.

The zone settings in `/etc/firewalld/` are a range of preset settings which can be quickly applied to a network interface. They are listed here with a brief explanation:

`drop (immutable)`
: Any incoming network packets are dropped, there is no reply. Only outgoing network connections are possible.

`block (immutable)`
: Any incoming network connections are rejected with an icmp-host-prohibited message for IPv4 and icmp6-adm-prohibited for IPv6. Only network connections initiated from within the system are possible.

`public`
: For use in public areas. You do not trust the other computers on the network to not harm your computer. Only selected incoming connections are accepted.

`external`
: For use on external networks with masquerading enabled especially for routers. You do not trust the other computers on the network to not harm your computer. Only selected incoming connections are accepted.

`dmz`
: For computers in your demilitarized zone that are publicly-accessible with limited access to your internal network. Only selected incoming connections are accepted.

`work`
: For use in work areas. You mostly trust the other computers on networks to not harm your computer. Only selected incoming connections are accepted.

`home`
: For use in home areas. You mostly trust the other computers on networks to not harm your computer. Only selected incoming connections are accepted.

`internal`
: For use on internal networks. You mostly trust the other computers on the networks to not harm your computer. Only selected incoming connections are accepted.

`trusted (immutable)`
: All network connections are accepted.

It is possible to designate one of these zones to be the default zone. When interface connections are added to NetworkManager, they are assigned to the default zone. On installation, the default zone in `firewalld` is set to be the public zone.

3.8.5. Choosing a Network Zone

The network zone names have been chosen to be self-explanatory and to allow users to quickly make a reasonable decision. However, a review of the default configuration settings should be made and unnecessary services disabled according to your needs and risk assessments.

3.8.6. Understanding Predefined Services

A service can be a list of local ports and destinations as well as a list of firewall helper modules automatically loaded if a service is enabled. The use of predefined services makes it easier for the user to enable and disable access to a service. Using the predefined services, or custom defined services, as opposed to opening ports or ranges or ports may make administration easier. Service configuration options and generic file information are described in the `firewalld.service(5)` man page. The services are specified by means of individual XML configuration files which are named in the following format: ***`service-name`***`.xml`.

To view the list of services using the graphical firewall-config tool, press the super key and start typing `firewall`. The firewall icon will appear. Press enter once it is highlighted. The firewall-

config tool appears. You will be prompted for your user password. You can now view the list of services under the Services tab.

To list the default predefined services available using the command line, issue the following command as root:

```
~]# ls /usr/lib/firewalld/services/
```

Files in /usr/lib/firewalld/services/ must not be edited. Only the files in /etc/firewalld/services/ should be edited.

To list the system or user created services, issue the following command as root:

```
~]# ls /etc/firewalld/services/
```

Services can be added and removed using the graphical firewall-config tool and by editing the XML files in /etc/firewalld/services/. If a service has not be added or changed by the user, then no corresponding XML file will be found in /etc/firewalld/services/. The files /usr/lib/firewalld/services/ can be used as templates if you wish to add or change a service. As root, issue a command in the following format:

```
~]# cp /usr/lib/firewalld/services/[service].xml /etc/firewalld/services/[service].xml
```

You may then edit the newly created file. firewalld will prefer files in /etc/firewalld/services/ but will fall back to /usr/lib/firewalld/services/ should a file be deleted, but only after a reload.

3.8.7. Understanding The Direct Interface

firewalld has a so called direct interface, which enables directly passing rules to iptables, ip6tables and ebtables. It is intended for use by applications and not users. It is dangerous to use the direct interface if you are not very familiar with iptables as you could inadvertently cause a breach in the firewall. firewalld still tracks what has been added, so it is still possible to query firewalld and see the changes made by an application using the direct interface mode. The direct interface is used by adding the --direct option to firewall-cmd.

The direct interface mode is intended for services or applications to add specific firewall rules during run time. The rules are not permanent and need to be applied every time after receiving the start, restart or reload message from firewalld using D-BUS.

3.8.8. Check if firewalld is installed

In Fedora firewalld and the graphical user interface configuration tool firewall-config are installed by default but firewall-applet is not. This can be checked by running the following command as root:

```
~]# yum install firewalld firewall-config
```

3.8.9. Disabling firewalld

To disable firewalld, run the following commands as root:

```
~]# systemctl disable firewalld # systemctl stop firewalld
```

3.8.9.1. Using the iptables service

To use the iptables service instead of `firewalld`, first disable `firewalld` by running the following command as root:

```
~]# systemctl disable firewalld # systemctl stop firewalld
```

Then install the *iptables-services* package by entering the following command as root:

```
~]# yum install iptables-services
```

Then, to start iptables service, run the following commands as root:

```
  # touch /etc/sysconfig/iptables
  # touch /etc/sysconfig/ip6tables
  # systemctl start iptables
  # systemctl start ip6tables
  # systemctl enable iptables
  # systemctl enable ip6tables
```

3.8.10. Start firewalld

To start `firewalld`, enter the following command as root:

```
~]# systemctl start firewalld
```

3.8.11. Check if firewalld is running

To check if `firewalld` is running, enter the following command:

```
~]$ systemctl status firewalld
firewalld.service - firewalld - dynamic firewall daemon
   Loaded: loaded (/usr/lib/systemd/system/firewalld.service; enabled)
   Active: active (running) since Sat 2013-04-06 22:56:59 CEST; 2 days ago
 Main PID: 688 (firewalld)
   CGroup: name=systemd:/system/firewalld.service
```

In addition, check if firewall-cmd can connect to the daemon by entering the following command:

```
~]$ firewall-cmd --state
running
```

3.8.12. Installing firewalld

To install `firewalld`, run the following command as root:

```
~]# yum install firewalld
```

To install the graphical user interface tool firewall-config, run the following command as root:

```
~]# yum install firewall-config
```

To install the optional firewall-applet, run the following command as root:

```
~]# yum install firewall-applet
```

3.8.13. Configuring the Firewall

The firewall can be configured using the graphical user interface tool firewall-config, using the command line interface tool firewall-cmd and by editing XML configuration files. These methods will be described in order.

3.8.13.1. Configuring the Firewall using the graphical user interface

3.8.13.1.1. Start the graphical firewall configuration tool

To start the graphical firewall-config tool, press the super key and start typing `firewall`. The firewall icon will appear. Press enter once it is highlighted. The firewall-config tool appears. You will be prompted for your user password.

To start the graphical firewall configuration tool using the command line, enter the following command as root user:

```
~]# firewall-config
```

The Firewall Configuration window opens. Note, this command can be run as normal user but you will then be prompted for the root password from time to time.

Look for the word Connected in the lower left corner. This indicates that the firewall-config tool is connected to the user space daemon, `firewalld`.

3.8.13.1.2. Change the firewall settings

To immediately change the current firewall settings, ensure the current view is set to Runtime Configuration. Alternatively, to edit the settings to be applied at the next system start, or firewall reload, select Permanent Configuration from the drop down list.

注記

When making changes to the firewall settings in Runtime Configuration mode, your selection takes immediate effect when you set or clear the check box associated with the service. You should keep this in mind when working on a system that may be in use by other users.

When making changes to the firewall settings in Permanent Configuration mode, your selection will only take effect when you reload the firewall or the system restarts. You can use the reload icon below the File menu, or click the Options menu and select Reload Firewall.

You can select zones in the left hand side column. You will notice the zones have some services enabled, you may need to resize the window or scroll to see the full list. You can customize the settings by selecting and deselecting a service except for the zones block, drop, and trusted as those zone settings are classified as immutable, they cannot be changed.

3.8.13.1.3. Add an Interface to a zone

To add or reassign an interface of a connection to zone, start firewall-config, select Options from the menu bar, select Change Zones of Connections from the drop down menu. The Network Connections window appears. Select the connection you wish to add or reassign and select Edit. The Editing a connection window appears. Select the General tab. Select the new firewall zone from the drop down menu and click Save.

3.8.13.1.4. Set the Default Zone

To set the default zone that new interfaces will be assigned to, start firewall-config, select Options from the menu bar, select Change Default Zone from the drop down menu. The System Default Zone window appears. Select the zone form the list that you want to be used as the default zone and click OK.

3.8.13.1.5. Configuring Services

To enable or disable a predefined or custom service, start the firewall-config tool and select the network zone whose services are to be configured. Select the Services tab and select the check box for each type of service you want to trust. Clear the check box to block a service.

To edit a service, start the firewall-config tool and then select Permanent Configuration mode from the drop-down selection menu labeled Current View. An Edit Services button appears on the right hand side of the ICMP Filter tab. Click Edit Services, the Service Settings window appears. Select the service you wish to configure. The Ports and Protocols tab enables adding, changing, and removing of ports and protocols for the selected service. The modules tab is for configuring Netfilter helper modules. The Destination tab enables limiting traffic to a particular destination address and Internet Protocol (IPv4 or IPv6.

3.8.13.1.6. Open Ports in the firewall

To permit traffic through the firewall to a certain port, start the firewall-config tool and select the network zone whose settings you want to change. Select the Ports tab and the click the Add button on the right hand side. The Port and Protocol window opens.

Enter the port number or range of ports to permit. Select tcp or udp from the drop down list.

3.8.13.1.7. Enable IP Address Masquerading

To translate IPv4 addresses to a single external address, start the firewall-config tool and select the network zone whose addresses are to be translated. Select the Masquerading tab and select the check box to enable the translation of IPv4 addresses to a single address.

3.8.13.1.8. Configure Port Forwarding

To forward inbound network traffic, or packets, for a specific port to an internal address or alternative port, first enable IP address masquerading, then select the Port Forwarding tab.

Select the protocol of the incoming traffic and the port or range of ports on the upper section of the window. The lower section is for setting details about the destination.

To forward traffic to a local port, that is to say to a port on the same system, select the Local forwarding check box. Enter the local port or range of ports for the traffic to be sent to.

To forward traffic to another IPv4 address, select the Forward to another port check box. Enter the destination IP address and port or port range. The default is to send to the same port if the port field is left empty. Click OK to apply the changes.

3.8.13.1.9. Configuring the ICMP Filter

To enable or disable an ICMP filter, start the firewall-config tool and select the network zone whose messages are to be filtered. Select the ICMP Filter tab and select the check box for each type of ICMP message you want to filter. Clear the check box to disable a filter. This setting is per direction and the default allows everything.

To edit an ICMP filter, start the firewall-config tool and then select Permanent Configuration mode from the drop-down selection menu labeled Current View. An Edit ICMP Types button appears on the right hand side of the ICMP Filter tab.

3.8.13.2. Configuring the Firewall using the command line tool, firewall-cmd

The command line tool firewall-cmd is part of the `firewalld` application which is installed by default. You can verify that it is installed by checking the version or displaying the help output. Enter the following command to check the version:

```
~]$ firewall-cmd -V, --version
```

Enter the following command to view the help output:

```
~]$ firewall-cmd -h, --help
```

We list a selection of commands below, for a full list please see the man page, `man firewall-cmd(1)`.

注記

In order to make a command permanent or persistent, add the `--permanent` option to all commands apart from the `--direct` commands (which are by their nature temporary). Note that this not only means the change will be permanent but that the change will only take effect after firewall reload, service restart, or after system reboot. Settings made with firewall-cmd without the `--permanent` option take effect immediately, but are only valid till next firewall reload, system boot, or `firewalld` service restart. Reloading the firewall does not in itself break connections, but be aware you are discarding temporary changes by doing so.

3.8.13.3. View the firewall settings using the CLI

To get a text display of the state of `firewalld`, enter the following command:

```
~]$ firewall-cmd --state
```

To view the list of active zones, with a list of the interfaces currently assigned to them, enter the following command:

```
~]$ firewall-cmd --get-active-zones
   public: em1 wlan0
```

To find out the zone that an interface, for example em1, is currently assigned to, enter the following command:

```
~]$  firewall-cmd --get-zone-of-interface=em1
public
```

To find out all the interfaces assigned to a zone, for example the public zone, enter the following command as root:

```
~]# firewall-cmd --zone=public --list-interfaces
     em1 wlan0
```

This information is obtained from NetworkManager and only shows interfaces not connections.

To find out all the settings of a zone, for example the public zone, enter the following command as root:

```
~]# firewall-cmd --zone=public --list-all
public
  interfaces:
  services: mdns dhcpv6-client ssh
  ports:
  forward-ports:
  icmp-blocks: source-quench
```

To view the network zones currently active, enter the following command as root:

```
~]# firewall-cmd --get-service
    cluster-suite pop3s bacula-client smtp ipp radius bacula ftp mdns samba dhcpv6-client dns openvpn imaps
 samba-client http https ntp vnc-server telnet libvirt ssh ipsec ipp-client amanda-client tftp-client nfs tftp
 libvirt-tls
```

This will list the names of the services in `/usr/lib/firewalld/services/`. Note that the configuration files themselves are named ***`service-name`***`.xml`.

To view the network zones that will be active after the next firewall reload, enter the following command as root:

```
~]# firewall-cmd --get-service --permanent
```

3.8.13.4. View the firewall settings using nmcli

To get a list of all the interfaces and actions assigned to a zone, enter the following command:

```
~]$  nmcli -f NAME,DEVICES,ZONE con status
NAME                      DEVICES    ZONE
my-little-wifi            wlan0      home
VPN connection 1          wlan0      work
System em1                em1        --
```

-- means the interface is assigned to the default zone.

3.8.13.5. Change the firewall settings using the Command Line Interface (CLI)

3.8.13.5.1. Drop All Packets (Panic Mode)

To start dropping all incoming and outgoing packets, enter the following command as root:

```
~]# firewall-cmd --panic-on
```

All incoming and outgoing packets will be dropped. Active connections will be terminated after a period of inactivity; the time taken depends on the individual session time out values.

To start passing incoming and outgoing packets again, enter the following command as root:

```
~]# firewall-cmd --panic-off
```

After disabling panic mode, established connections might work again if panic mode was enabled for a short period of time.

To get a text indication if panic mode is enabled or disabled, enter the following command:

```
~]$  firewall-cmd --query-panic && echo "enabled" || echo "Not enabled"
```

3.8.13.5.2. Reload the firewall using the CLI

To reload the firewall with out interrupting user connections, that is to say, with out losing state information, enter the following command as root:

```
~]# firewall-cmd --reload
```

To reload the firewall and interrupt user connections, that is to say, to discard state information, enter the following command as root:

```
~]# firewall-cmd --complete-reload
```

This command should normally only be used in case of severe firewall problems. For example, if there are state information problems and no connection can be established but the firewall rules are correct.

3.8.13.5.3. Add an Interface to a Zone using the CLI

To add an interface to a zone, for example to add em1 to the public zone, enter the following command as root:

```
~]# firewall-cmd --zone=public --add-interface=em1
```

To make this setting permanent, add the `--permanent` option and reload the firewall.

3.8.13.5.4. Add an Interface to a Zone by Editing the Interface Configuration File

To add an interface to a zone by editing the `ifcfg-em1` configuration file, for example to add em1 to the work zone, as root use an editor to add the following line to `ifcfg-em1`:

```
ZONE=work
```

Note that if you omit the `ZONE` option, or use `ZONE=`, or `ZONE=''`, then the default zone will be used.

NetworkManager will automatically reconnect and the zone will be set accordingly.

3.8.13.5.5. Configure the default zone by Editing the firewalld Configuration File

As root, open `/etc/firewalld/firewalld.conf` and edit the file as follows:

```
# default zone
# The default zone used if an empty zone string is used.
# Default: public
DefaultZone=home
```

Reload the firewall, by entering the following command as root:

```
~]# firewall-cmd --reload
```

This will reload the firewall without losing state information (TCP sessions will not be interrupted).

3.8.13.5.6. Set the default zone by using the CLI

To set the default zone, for example to public, enter the following command as root:

```
~]# firewall-cmd --set-default-zone=public
```

This change will take immediate effect and in this case it is not necessary to reload the firewall.

3.8.13.5.7. Open Ports in the Firewall using the CLI

List all open ports for a zone, for example dmz, by entering the following command as root:

```
~]# firewall-cmd --zone=dmz --list-ports
```

To add a port to a zone, for example to allow TCP traffic to port 8080 to the dmz zone, enter the following command as root:

```
~]# firewall-cmd --zone=dmz --add-port=8080/tcp
```

To make this setting permanent, add the `--permanent` option and reload the firewall.

To add a range of ports to a zone, for example to allow the ports from 5060 to 5061 to the public zone, enter the following command as root:

```
~]# firewall-cmd --zone=public --add-port=5060-5061/udp
```

To make this setting permanent, add the `--permanent` option and reload the firewall.

3.8.13.5.8. Add a Service to a Zone using the CLI

To add a service to a zone, for example to allow SMTP to the work zone, enter the following command as root:

```
~]# firewall-cmd --zone=work --add-service=smtp
```

To make this setting permanent, add the `--permanent` option and reload the firewall.

3.8.13.5.9. Remove a Service from a Zone using the CLI

To remove a service from a zone, for example to remove SMTP from the work zone, enter the following command as root:

```
~]# firewall-cmd --zone=work --remove-service=smtp
```

Add the `--permanent` option to make the change persist after system boot. If using this option and you wish to make the change immediate, reload the firewall, by entering the following command as root:

```
~]# firewall-cmd --reload
```

Note, this will not break established connections. If that is your intention, you could use the `--complete-reload` option but this will break all established connections not just for the service you have removed.

3.8.13.5.10. Add a Service to a Zone by Editing XML files

To view the default zone files, enter the following command as root:

```
~]# ls /usr/lib/firewalld/zones/
block.xml  drop.xml      home.xml      public.xml   work.xml
dmz.xml    external.xml  internal.xml  trusted.xml
```

These files must not be edited. They are used by default if no equivalent file exists in the `/etc/firewalld/zones/` directory.

To view the zone files that have been changed from the default, enter the following command as root:

```
~]# ls /etc/firewalld/zones/
external.xml  public.xml  public.xml.old
```

In the example shown above, the work zone file does not exist. To add the work zone file, enter the following command as root:

```
~]# cp /usr/lib/firewalld/zones/work.xml /etc/firewalld/zones/
```

You can now edit the file in the `/etc/firewalld/zones/` directory. If you delete the file, `firewalld` will fall back to using the default file in `/usr/lib/firewalld/zones/`.

To add a service to a zone, for example to allow `SMTP` to the work zone, use an editor with root privileges to edit the `/etc/firewalld/zones/work.xml` file to include the following line:

```
<service name="smtp"/>
```

3.8.13.5.11. Remove a Service from a Zone by Editing XML files

An editor running with root privileges is required to edit the XML zone files. To view the files for previously configured zones, enter the following command as root:

```
~]# ls /etc/firewalld/zones/
external.xml  public.xml  work.xml
```

To remove a service from a zone, for example to remove `SMTP` from the work zone, use an editor with root privileges to edit the `/etc/firewalld/zones/work.xml` file to remove the following line:

```
<service name="smtp"/>
```

If no other changes have been made to the work.xml file, it can be removed and firewalld will use the default /usr/lib/firewalld/zones/work.xml configuration file after the next reload or system boot.

3.8.13.5.12. Configure IP Address Masquerading

To check if IP masquerading is enabled, for example for the external zone, enter the following command as root:

```
~]# firewall-cmd --zone=external --query-masquerade && echo "enabled" || echo "Not enabled"
```

If zone is omitted, the default zone will be used.

To enable IP masquerading, enter the following command as root:

```
~]# firewall-cmd --zone=external --add-masquerade
```

To make this setting permanent, add the --permanent option and reload the firewall.

To disable IP masquerading, enter the following command as root:

```
~]# firewall-cmd --zone=external --remove-masquerade
```

To make this setting permanent, add the --permanent option and reload the firewall.

3.8.13.5.13. Configure Port Forwarding using the CLI

To forward inbound network packets from one port to an alternative port or address, first enable IP address masquerading for a zone, for example external, by entering the following command as root:

```
~]# firewall-cmd --zone=external --add-masquerade
```

To forward packets to a local port, that is to say to a port on the same system, enter the following command as root:

```
~]# firewall-cmd --zone=external --add-forward-port=port=22:proto=tcp:toport=3753
```

In this example, the packets intended for port 22 are now forwarded to port 3753. The original destination port is specified with the port option. This option can be a port, or port range, together with a protocol. The protocol, if specified, must be one of either tcp or udp. The new local port, the port or range of ports to which the traffic is being forwarded to, is specified with the toport option. To make this setting permanent, add the --permanent option and reload the firewall.

To forward packets to another IPv4 address, usually an internal address, without changing the destination port, enter the following command as root:

```
~]# firewall-cmd --zone=external --add-forward-port=port=22:proto=tcp:toaddr=192.0.2.55
```

In this example, the packets intended for port 22 are now forwarded to the same port at the address given with the toaddr. The original destination port is specified with the port. This option

can be a port, or port range, together with a protocol. The protocol, if specified, must be one of either tcp or udp. The new destination port, the port or range of ports to which the traffic is being forwarded to, is specified with the toport. To make this setting permanent, add the --permanent option and reload the firewall.

To forward packets to another port at another IPv4 address, usually an internal address, enter the following command as root:

```
~]# firewall-cmd --zone=external --add-forward-port=port=22:proto=tcp:toport=2055:toaddr=192.0.2.55
```

In this example, the packets intended for port 22 are now forwarded to port 2055 at the address given with the toaddr. The original destination port is specified with the port. This option can be a port, or port range, together with a protocol. The protocol, if specified, must be one of either tcp or udp. The new destination port, the port or range of ports to which the traffic is being forwarded to, is specified with the toport. To make this setting permanent, add the --permanent option and reload the firewall.

3.8.13.6. Configuring The Firewall Using XML Files

The configuration settings for firewalld are stored in XML files in the /etc/firewalld/ directory. Do not edit the files in the /usr/lib/firewalld/ directory, they are for the default settings. You will need root user permissions to view and edit the XML files. The XML files are explained in three man pages:

- firewalld.icmptype(5) man page — Describes XML configuration files for ICMP filtering.
- firewalld.service(5) man page — Describes XML configuration files for firewalld service.
- firewalld.zone(5) man page — Describes XML configuration files for firewalld zone configuration.

The XML files can be created and edited directly or created indirectly using the graphical and command line tools. Organizations can distribute them in RPM files which can make management and version control easier. Tools such as Puppet can distribute such configuration files.

3.8.13.7. Using the direct interface

It is possible to add and remove chains during runtime by using the --direct option with the firewall-cmd tool. A few examples are presented here, please see the firewall-cmd(1) man page for more information.

It is dangerous to use the direct interface if you are not very familiar with iptables as you could inadvertently cause a breach in the firewall.

The direct interface mode is intended for services or applications to add specific firewall rules during run time. The rules are not permanent and need to be applied every time after receiving the start, restart or reload message from firewalld using D-BUS.

3.8.13.7.1. Adding a custom rule using the direct interface

To add a custom rule to the chain IN_ZONE_public_allow, issuing a command as root in the following format:

```
~]# firewall-cmd --direct --add-rule ipv4 filter IN_ZONE_public_allow 0 -m tcp -p tcp --dport 666 -j ACCEPT
```

3.8.13.7.2. Removing a custom rule using the direct interface

To remove a custom rule from the chain IN_ZONE_public_allow, issuing a command as root in the following format:

```
~]# firewall-cmd --direct --remove-rule ipv4 filter IN_ZONE_public_allow -m tcp -p tcp --dport 666 -j ACCEPT
```

3.8.13.7.3. Listing custom rules using the direct interface

To list the rules in the chain IN_ZONE_public_allow, issuing a command as root in the following format:

```
~]# firewall-cmd --direct --get-rules ipv4 filter IN_ZONE_public_allow
```

3.8.14. Additional Resources

The following sources of information provide additional resources regarding `firewalld`.

3.8.14.1. Installed Documentation

- `firewalld(1)` man page — Describes command options for `firewalld`.
- `firewalld.conf(5)` man page — Contains information to configure `firewalld`.
- `firewall-cmd(1)` man page — Describes command options for the `firewalld` command line client.
- `firewalld.icmptype(5)` man page — Describes XML configuration files for ICMP filtering.
- `firewalld.service(5)` man page — Describes XML configuration files for firewalld serivice.
- `firewalld.zone(5)` man page — Describes XML configuration files for `firewalld` zone configuration.

3.8.14.2. Useful Websites

https://fedoraproject.org/wiki/FirewallD

The website of the upstream project.

暗号化

保護されなければいけない、主な2種類のデータがあります: 静止しているデータと動作しているデータ。これらの異なる種類のデータは同じ技術を用いて同じ方法で保護されますが、実装は完全に異なります。同じ情報が静止していて、同時に異なる場所で動作しているかもしれないので、1つの保護の実装が、すべてのありえる漏えいの方法を防ぐことはできません。

4.1. 静止しているデータ

静止しているデータとは、ハードディスク、テープ、CD、DVD および他のメディアに保存されているデータです。この情報の最大の脅威は物理的な盗難から起こります。

4.1.1. 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良の方法の1つです。各ファイルが保護されるだけでなく、これらのファイルの一部が含まれるかもしれない一時的なストレージも保護されます。完全ディスク暗号化はすべてのファイルを保護するので、保護したいものを選択すること、およびファイルを見落とすかもしれないことについて心配する必要がありません。

Fedora は、ネイティブに LUKS 暗号化をサポートします。LUKS は、コンピュータがオフの間にデータを保護するよう、ハードディスクのパーティションを大量に暗号化します。これにより、攻撃者がコンピュータにログインするためにシングルユーザーモードを使用しようとしたり、他のアクセスを得ようとしたりすることからコンピュータを保護します。

LUKS のような完全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ときにだけデータを保護します。コンピューターがオンになり、LUKS がディスクを復号すると、ディスクにあるファイルはそれらに普通にアクセスできるすべての人が利用可能になります。コンピューターがオンのときにファイルを保護するために、ファイルベースの暗号化のような他のソリューションと組み合わせて完全ディスク暗号化を使用します。また、コンピューターから離れるときに、ロックすることを忘れないようにします。スクリーンセーバーを保護するパスフレーズが数分の未使用でアクティブになるよう設定することは、侵入者を追いやるために良い方法です。

4.1.2. ファイルベースの暗号化

GnuPG (GPG) は、ファイルや電子メールメッセージを署名かつ/または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。メッセージやファイルの完全性を維持するために有用です。また、ファイルや電子メールに含まれる情報の機密性も保護します。電子メールの場合、GPG は二重の保護を提供します。メッセージがネットワークを越えて送信されると、静止しているデータだけでなく動作しているデータを保護します。

ファイルベースの暗号化は、ファイルがコンピューターを離れた後（郵送で CD を送るときのように）保護することを意図しています。いくつかのファイルベースの暗号化ソリューションは、コンピューターへ物理的にアクセスした攻撃者がある環境において復元できる、暗号化されたファイルの残りをそのままにします。コンピューターにアクセスした攻撃者からそれらのファイルのコンテンツを保護するために、完全ディスク暗号化のような他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。

4.2. 動作しているデータ

動作しているデータとは、ネットワークを越えて転送されているデータです。動作しているデータに対する最大の脅威は盗聴と改ざんです。ユーザー名とパスワードは、なりすましをする、もしくは機密情報へのアクセスを得るために、誰かにより盗聴されて使用される可能性があるので、保護なしでネットワークを越えて転送されるべきではありません。銀行アカウント情報のような他のプライベートな情報もネットワークを越えて転送されるときに保護されるべきです。ネットワーク・セッションが暗号化されているならば、転送されているときにデータが危険にさらされていることをあまり心配する必要はありません。

動作しているデータは、攻撃者がデータが保存されているコンピューターの近くにいる必要がなく、経路のどこかにいればよいため、特に攻撃者へ脆弱です。暗号化トンネルはコミュニケーションの経路に沿ってデータを保護できます。

4.2.1. Virtual Private Networks (VPNs)

いくつかのサテライト・オフィスを持つ組織は、転送中に機密データの効率性と保護に対して専用線を用いてお互いに接続します。たとえば、多くの企業は、あるオフィスを他とつなぐためにエンド間のネットワーク・ソリューションとして、フレームリレーまたは *Asynchronous Transfer Mode* (ATM) 回線を使用します。これは高価な提案です、とくに高いコストを払うことなくエンタープライズ・レベルの専用デジタル回線を結びつける拡張を期待する中小企業 (SMB: small to medium sized businesses) にとってはそうです。

このニーズに取り組むために、*Virtual Private Networks* (VPN) が開発されました。専用線と同じ機能原則に従うことで、VPN は、既存の *Local Area Networks* (LAN) から *Wide Area Network* (WAN) を作成する、2者（またはネットワーク）間でセキュアなデジタル・コミュニケーションが可能になります。フレームリレーや ATM との違いは、その転送メディアです。VPN はトランスポート層としてデータグラムを使用して、意図した宛て先へとインターネットを経由してセキュアなトンネルにする IP 上で転送されます。最もフリーなソフトウェアの VPN 実装は、転送においてデータをさらにマスクするために、オープンで標準的な暗号方式を組み込んでいます。

いくつかの組織はセキュリティを強化するためにハードウェア VPN ソリューションを使用します。一方、それ以外の組織はソフトウェアもしくはプロトコル・ベースの実装を使用します。Cisco, Nortel, IBM, や Checkpoint のような、いくつかのベンダーはハードウェア VPN ソリューションを提供します。標準化された *Internet Protocol Security* (IPsec) 実装を利用した FreeS/Wan と呼ばれる Linux 向けのフリーのソフトウェア・ベースの VPN ソリューションがあります。これらの VPN ソリューションは、ハードウェアかソフトウェア・ベースかによらず、あるオフィスからもう1つへの IP コネクション間に存在する専用のルーターとして動作します。

4.2.1.1. VPN はどのように機能しますか?

パケットがクライアントから転送されるとき、ルーティングと認証のために *Authentication Header* (AH) を追加する、VPN ルーターまたはゲートウェイを通過して送られます。データは暗号化され、最終的に *Encapsulating Security Payload* (ESP) に囲い込まれます。後者は復号とハンドリング指示を取り扱います。

受信している VPN ルーターはヘッダー情報を分離して、データを復号して、そしてそれを意図した宛て先（ワークステーションもしくはネットワークにある他のノード）にルートします。ネットワーク−ネットワーク間のコネクションを使用していると、ローカルネットワークにある受信ノードは、すでに復号されて、処理する準備ができているパケットを受け取ります。ネットワーク−ネットワーク間の VPN コネクションにおいて暗号化/復号プロセスはローカルノードに透過的です。

そのように高くされたレベルのセキュリティを用いると、攻撃者はパケットを横取りしてはいけないだけでなく、パケットを復号してはいけません。サーバーとクライアント間で中間者攻撃を使用する侵入者は、認証セッションに対する秘密鍵を少なくとも1つにアクセスできなければいけません。認証と暗号化のいくつかの層を使用するので、VPN は単一化されたイントラネットとして動作するために、複数のリモートノードを接続するセキュアかつ効果的な手段です。

4.2.1.2. VPN および Fedora

Fedora は WAN をセキュアに接続するために、ソフトウェア・ソリューションの実装に関してさまざまなオプションを提供します。*Internet Protocol Security* (IPsec) は、Fedora のためのサポートされた VPN 実装です。また、支店やリモート・ユーザーとともに組織の利便性のニーズを十分に取り組みます。

4.2.1.3. IPsec

Fedora は、インターネットのような一般的なキャリア・ネットワークにおいてセキュアなトンネルを使用して、お互いにリモートのホストとネットワークを接続するために IPsec をサポートします。IPsec は、ホスト−ホスト間（あ

るコンピュータ・ワークステーションともう一方）またはネットワーク－ネットワーク間（ある LAN/WAN ともう一方）の設定を使用して導入されます。

Fedora における IPsec 実装は、接続しているシステム間で相互認証およびセキュアな関連づけのために使用される、Internet Engineering Task Force (IETF) により実装されたプロトコル、*Internet Key Exchange* (*IKE*) を使用します。

4.2.1.4. IPsec 接続の作成

IPsec コネクションは、2つの論理的なフェーズに分かれています。フェーズ1は、IPsec ノードがリモートのホストまたはネットワークとコネクションを初期化します。リモートのノードまたはネットワークは、リクエストしているノードのクレディンシャルをチェックして、両当事者はコネクションの認証方式をネゴシエーションします。

Fedora システムにおいては、IPsec コネクションは IPsec ノード認証の ###### 方式を使用します。事前共有キー IPsec コネクションにおいて、両方のホストは IPsec コネクションのフェーズ2に移行するために同じキーを使用しなければいけません。

IPsec コネクションのフェーズ2は、*Security Association* (SA) が IPsec ノード間に作成されるところです。このフェーズは、暗号化方式、秘密セッションキーの交換パラメーター、およびその他のような、設定情報を持つ SA データベースを確立します。このフェーズは、リモートノードとネットワーク間で実際の IPsec コネクションを管理します。

IPsec の Fedora 実装は、インターネットを越えるホスト間でキーを共有するために IKE を使用します。racoon キー管理デーモンは、IKE キーの配布と交換を取り扱います。このデーモンの詳細は racoon マニュアル・ページを参照してください。

4.2.1.5. IPsec のインストール

IPsec を実装するには、すべての IPsec ホスト（ホスト－ホスト間の設定なら）またはルーター（ネットワーク－ネットワーク間の設定なら）において、ipsec-tools RPM パッケージがインストールされている必要があります。RPM パッケージは、IPsec コネクションをセットアップするために、以下を含めて基本的なライブラリ、デーモンおよび設定ファイルを含みます。

- /sbin/setkey — カーネルにおける IPsec のキー管理およびセキュリティ属性を操作します。この実行コマンドは racoon キー管理デーモンにより制御されます。詳細は setkey(8) マニュアル・ページを参照してください。
- /usr/sbin/racoon — IKE キー管理デーモン。IPsec 接続されたシステム間でセキュリティ・アソシエーションとキー共有を管理および制御するために使用されます。
- /etc/racoon/racoon.conf — racoon デーモンの設定ファイル。コネクションに使用される認証方式および暗号化アルゴリズムを含む IPsec 接続のさまざまな観点を設定するために使用されます。利用可能なディレクティブの完全な一覧は racoon.conf(5) を参照してください。

Fedora において IPsec を設定するために、ネットワーク管理ツール を使用できます、もしくはネットワーク および IPsec 設定ファイルを手動で編集します。

- IPsec 経由でネットワーク接続された2つのホストを接続するために、######### *IPsec* #### を参照してください。
- IPsec 経由である LAN/WAN ともう一方を接続するために、################# *IPsec* ### を参照してください。

4.2.1.6. ホスト－ホスト間 IPsec の設定

IPsec は、あるデスクトップまたはワークステーション（ホスト）が他のものと、ホスト－ホスト間コネクションを使用して接続するために設定されます。この種類のコネクションは、各ホスト間でセキュアなトンネルを作成するた

めに、各ホストが接続されるネットワークを使用します。ホストーホスト間コネクションの必要要件は、各ホストにおいて IPsec の設定をする、という最小のものです。ホストは（インターネットのような）キャリアのネットワークへの専用のコネクションと IPsec コネクションを作成するための Fedora のみを必要とします。

4.2.1.6.1. ホストーホスト間コネクション

ホストーホスト間 IPsec コネクションは、どちらも同じ認証キーを用いて IPsec を実行している、2つのシステム間の暗号化されたコネクションです。IPsec コネクションをアクティブにすると、2つのホスト間のネットワーク・トラフィックはすべて暗号化されます。

ホストーホスト間 IPsec コネクションを設定するために、各ホストに対して以下の手順を使用します:

注記

設定している実際のマシンにおいて以下の手順を実行すべきです。リモートで設定して、IPsec 接続を確立しようとする試みは避けるべきです。

1. ネットワーク管理ツールを起動するために、コマンド・シェルにおいて system-config-network と入力します。
2. IPsec設定ウィザードを起動するために、IPsec タブにおいて新規をクリックします。
3. ホストーホスト間の IPsec コネクションの設定を開始するために進むをクリックします。
4. コネクションのための一意な名前、たとえば ipsec0 を入力します。必要に応じて、コンピュータが開始するときに自動的にコネクションを有効化するためにチェックボックスを選択します。続けるために進むをクリックします。
5. コネクションの種類として ホストーホスト間暗号化 を選択して、進むをクリックします。
6. 使用する暗号化の種類を選択します: 手動または自動。

   手動暗号化を選択すると、暗号キーが後のプロセスにおいて提供されなければいけません。自動暗号化を選択すると、racoon デーモンが暗号キーを管理します。自動暗号化を使用したいならば、ipsec-tools パッケージがインストールされていなければいけません。

   続けるために進むをクリックします。
7. リモートホストの IP アドレスを入力します。

   リモートホストの IP アドレスを決めるために、########以下のコマンドを使用します。:

   ```
   [root@myServer ~] # /sbin/ifconfig <device>
   ```

   ここで <*device*> は VPN 接続のために使用したいイーサネット・デバイスです。

   システムに1つだけイーサネットカードが存在するならば、デバイス名は一般的に eth0 です。以下の例はこのコマンドに関連する情報を表示します（これは出力のみの例であることに注意してください）。

   ```
   eth0      Link encap:Ethernet  HWaddr 00:0C:6E:E8:98:1D
             inet addr:172.16.44.192  Bcast:172.16.45.255  Mask:255.255.254.0
   ```

   IP アドレスは inet addr: ラベルに続く番号です。

注記

ホスト−ホスト間コネクションのために、どちらのホストもパブリックで、ルート可能なアドレスを持つ必要があります。代わりに、どちらも同じ LAN にある限り、プライベートで、ルート不可能なアドレス（たとえば、10.x.x.x または 192.168.x.x 範囲から）を持つことができます。

ホストが異なる LAN にあるならば、もしくは、一方がパブリック・アドレスを持ち、他方がプライベート・アドレスを持つならば、################## *IPsec* ### を参照してください。

続けるために進むをクリックします。

8. 手順 *6* において手動の暗号化を選択していると、使用する暗号キーを指定するか、それを生成するために生成をクリックします。

   a. 認証キーを指定します。もしくはそれを生成するために生成をクリックします。数字と文字のあらゆる組み合わせが可能です。

   b. 続けるために進むをクリックします。

9. IPsec — Summary ページにおいて情報を確認し、Apply をクリックします。

10. 設定を保存するために ファイル => 保存 をクリックします。

    変更を有効にするために、ネットワークを再起動する必要があるかもしれません。ネットワークを再起動するために、以下のコマンドを使用します:

```
[root@myServer ~]# service network restart
```

11. リストから IPsec コネクションを選択して、Activate ボタンをクリックします。

12. 他のホストに対しても手順全体を繰り返します。手順 *8* からの同じキーを他のホストにおいて使うことが不可欠です。さもなければ、IPsec はうまく動作しません。

IPsec コネクションを設定した後、*#4.1#IPsec ###* に示されるように IPsec リストに表示されます。

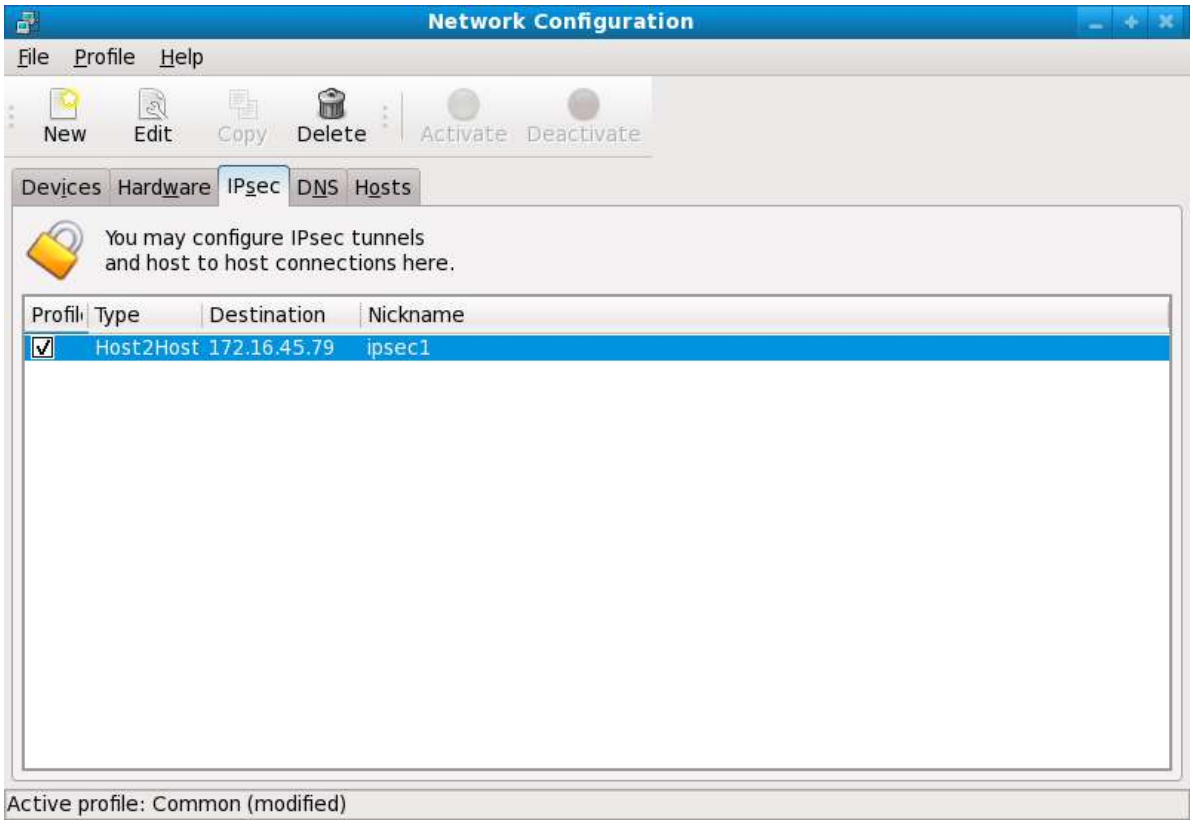

図4.1 IPsec 接続

IPsec 接続が設定されたとき、以下のファイルが作成されます:

- /etc/sysconfig/network-scripts/ifcfg-***<nickname>***
- /etc/sysconfig/network-scripts/keys-***<nickname>***
- /etc/racoon/***<remote-ip>***.conf
- /etc/racoon/psk.txt

自動暗号化が選択されていると、/etc/racoon/racoon.conf も作成されます。

インタフェースが起動するとき、***<remote-ip>***.conf を含めるために、/etc/racoon/racoon.confが修正されます。

4.2.1.6.2. 手動のホストーホスト間 IPsec の設定

コネクションを設定する第一歩は、各ワークステーションからシステムとネットワークの情報を集めることです。ホストーホスト間コネクションに対して、以下が必要になります:

- 各ホストの IP アドレス
- 一意な名前。たとえば、ipsec1。これは IPsec コネクションを識別し、他のデバイスまたはコネクションと区別するために使用されます。
- 固定された暗号キーまたは racoon により自動的に生成されたもの。
- コネクションの初期化段階で使用され、セッション中に暗号キーを交換するために事前共有された認証キー。

たとえば、ワークステーションAとワークステーションBが IPsec トンネルを通してお互いに接続していると仮定してください。Key_Value01 の値を持つ事前共有キーを用いて接続したく、racoon が各ホスト間の認証キーを自動的に生成および共有できるようにすることをユーザーが賛成します。どちらのホストもそのコネクションを ipsec1 と名づけることに決めます。

注記

大文字、小文字、数字および句読点の混在を使用する PSK を選択すべきです。推測が容易な PSK がセキュリティ・リスクを構成します。

各ホストに対して同じコネクション名を使用する必要はありません。インストールに便利でふさわしい名前を選択すべきです。

以下はワークステーションに対する IPsec 設定ファイルです。ワークステーション B とのホスト−ホスト間の IPsec コネクションのために A。この例においてコネクションを識別するための一意な名前が *ipsec1* です。そのため、結果ファイルは /etc/sysconfig/network-scripts/ifcfg-ipsec1 と呼ばれます。

```
DST=X.X.X.XTYPE=IPSEC
ONBOOT=no
IKE_METHOD=PSK
```

ワークステーション A に対して、*X.X.X.X* はワークステーション B の IP アドレスです。ワークステーション B に対して、*X.X.X.X* はワークステーション A の IP アドレスです。この接続はブート時に初期化するよう設定されていません (ONBOOT=no) 。また、事前共有キー認証方式を使用します (IKE_METHOD=PSK) 。

以下は、両方のワークステーションがお互いを認証するために必要となる、事前共有キーファイル (/etc/sysconfig/network-scripts/keys-ipsec1 と呼ばれます) のコンテンツです。このファイルのコンテンツは両方のワークステーションで同じであるべきです。また、root ユーザーだけがこのファイルを読み書きできるべきです。

```
IKE_PSK=Key_Value01
```

重要

root ユーザーのみがファイルを読み込みおよび編集できるように keys-ipsec1 ファイルを変更するために、ファイルを作成した後で以下のコマンドを使用します:

```
[root@myServer ~] # chmod 600 /etc/sysconfig/network-scripts/keys-ipsec1
```

いつでも認証キーを変更するために、両方のワークステーションにおいて keys-ipsec1 ファイルを編集します。##########。

次の例は、リモート・ホストへのフェーズ1コネクションに対する具体的な設定を示します。このファイルは ***X.X.X.X.***conf と呼ばれます。ここで、*X.X.X.X* はリモート IPsec ホストの IP アドレスです。このファイルは IPsec トンネルが有効化されるとき自動的に作成され、直接編集すべきではありません。

```
remote X.X.X.X{
        exchange_mode aggressive, main;
```

```
  my_identifier address;
  proposal {
   encryption_algorithm 3des;
  hash_algorithm sha1;
  authentication_method pre_shared_key;
  dh_group 2 ;
 }
}
```

IPsec コネクションが初期化されるときに作成される、デフォルトのフェーズ1設定ファイルは、IPsec の Fedora 実装により使用される以下の命令文を含みます。

remote *X.X.X.X*
: この設定ファイルの以降の節は *X.X.X.X* IP アドレスにより識別されるリモート・ホストに対してのみ適用されることを指定します。

exchange_mode aggressive
: Fedora における IPsec のデフォルトの設定は、複数のホストを用いたいくつかの IPsec コネクションの設定を許可する、コネクションのオーバーヘッドがより少ない、アグレッシブ認証モードを使用します。

my_identifier address
: ノードを認証するときに使用するための識別方式を指定します。Fedora はノードを識別するために IP アドレスを使用します。

encryption_algorithm 3des
: 認証の間に使用される暗号化の方式を指定します。デフォルトで *Triple Data Encryption Standard* (3DES) が使用されます。

hash_algorithm sha1;
: ノード間でフェーズ1ネゴシエーションの間に使用されるハッシュ・アルゴリズムを指定します。デフォルトで Secure Hash Algorithm バージョン 1 が使用されます。

authentication_method pre_shared_key
: ノードのネゴシエーション中に使用される認証方式を指定します。デフォルトで Fedora は認証のために事前共有キーを使用します。

dh_group 2
: 動的に生成されるセッション・キーのために Diffie-Hellman グループ番号を指定します。デフォルトで modp1024 (group 2) が使用されます。

4.2.1.6.2.1. Racoon 設定ファイル

/etc/racoon/racoon.conf ファイルは、include "/etc/racoon/***X.X.X.X***.conf" 命令文を###、すべての IPsec ノードにおいて同一でなければいけません。この命令文（および、それが参照するファイル）が、IPsec トンネルが有効化されるときに生成されます。ワークステーションAに対して、include 命令文¥nにおける *X.X.X.X* はワークステーションBの IP アドレスです。 以下は、IPsec コネクションが有効化されるとき、典型的な racoon.conf ファイルを示します。

```
# Racoon IKE daemon configuration file.
# See 'man racoon.conf' for a description of the format and entries.

path include "/etc/racoon";
path pre_shared_key "/etc/racoon/psk.txt";
path certificate "/etc/racoon/certs";

sainfo anonymous
{
        pfs_group 2;
```

```
        lifetime time 1 hour ;
        encryption_algorithm 3des, blowfish 448, rijndael ;
        authentication_algorithm hmac_sha1, hmac_md5 ;
        compression_algorithm deflate ;
}
include "/etc/racoon/X.X.X.X.conf";
```

このデフォルトの racoon.conf ファイルは、IPsec 設定、事前共有キーファイル、および証明書に対して定義されたパスを含みます。sainfo anonymous にあるフィールドは、IPsec ノード間でフェーズ2 SA を記載します — IPsec コネクション（使用することがサポートされた暗号化アルゴリズムを含めて）の性質およびキー交換の方式。以下のリストは、フェーズ2のフィールドを定義します:

sainfo anonymous
: SA が、IPsec クレディンシャルがマッチする、提供されたすべての相手を匿名で初期化できることを意味します。

pfs_group 2
: Diffie-Hellman キー交換プロトコルを定義します。これは、IPsec ノードが IPsec コネクションの第2フェーズに対する共通の一時的セッションを確立することにより、方式を決定します。デフォルトで、Fedora の IPsec 実装は、Diffie-Hellman 暗号キー交換グループのグループ2（または、modp1024）を使用します。グループ2は、秘密鍵が漏えいしたときさえ、攻撃者が以前の IPsec 転送を復号することを防ぐ、1024ビットのモジュールの累乗法を使用します。

lifetime time 1 hour
: このパラメーターは SA の有効期間を指定します。時間またはデータのバイトにより定量化されます。デフォルトの IPsec の Fedora 導入は1時間の有効期間を指定します。

encryption_algorithm 3des, blowfish 448, rijndael
: フェーズ2のためにサポートされる暗号化の方式を指定します。Fedora は 3DES, 448-bit Blowfish, および Rijndael (*Advanced Encryption Standard*, または AES で使用される暗号) をサポートします。

authentication_algorithm hmac_sha1, hmac_md5
: 認証のためにサポートされたハッシュ・アルゴリズムを表示します。サポートされるモードは sha1 および md5 の hashed message authentication codes (HMAC) です。

compression_algorithm deflate
: IP Payload Compression (IPCOMP) に対して Deflate 圧縮アルゴリズムを定義します。これは、低速なネットワークにおいて IP データグラムのより速い転送を潜在的にできるようにします。

接続を開始するために、各ホストにおいて以下のコマンドを使用します:

```
[root@myServer ~]# /sbin/ifup <nickname>
```

ここで <nickname> は IPsec 接続に対して指定した名前です。

IPsec コネクションをテストするために、ホスト間で転送されるネットワーク・パケットを表示して、IPsec 経由で暗号化されていることを検証するために、tcpdump ユーティリティを実行します。パケットは AH ヘッダーを含むべきであり、ESP パケットとして示されるべきです。ESP はそれが暗号化されていることを意味します。たとえば:

```
[root@myServer ~]# tcpdump -n -i eth0 host <targetSystem>⏎ ⏎ IP 172.16.45.107 > 172.16.44.192:
 AH(spi=0x0954ccb6,seq=0xbb): ESP(spi=0x0c9f2164,seq=0xbb)
```

4.2.1.7. ネットワークーネットワーク間の IPsec 設定

IPsec は、ネットワークーネットワーク間のコネクションを用いて、ネットワーク全体（LAN や WAN のような）をリモートネットワークを接続するために設定することもできます。ネットワークーネットワーク間のコネクション

は、LAN にあるノードからリモート LAN にあるノードへと情報を透過的に処理して中継するために、接続しているネットワークの両側においてIPsec ルーターのセットアップが必要となります。*#4.2################ IPsec ############* は、ネットワーク－ネットワーク間 IPsec トンネル・コネクションを示します。

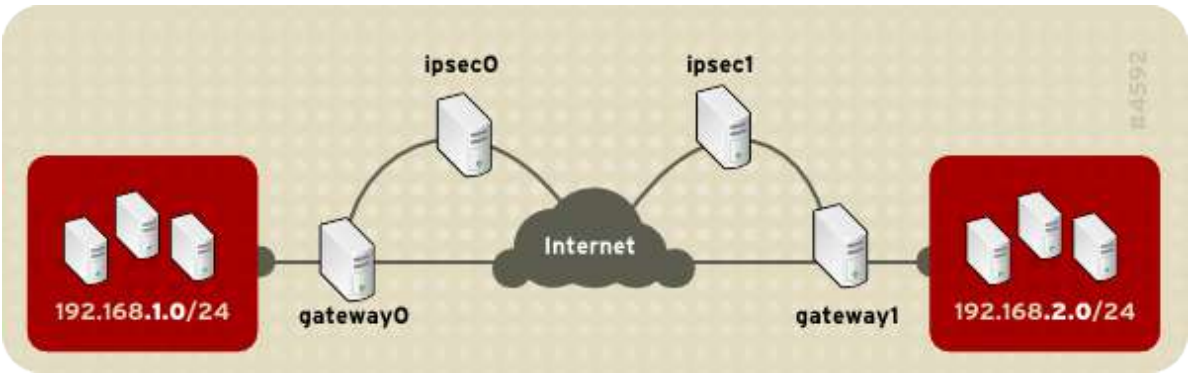

図4.2 ネットワーク－ネットワーク間の IPsec トンネル・コネクション

このダイアログは、2つの別々の LAN がインターネットにより分けられていることを示しています。これらの LAN は、インターネットを経由したセキュアなトンネルを用いてコネクションを認証および初期化するために、IPsec ルーターを使用します。転送中に横取りされたパケットは、これらの LAN の間でパケットを保護している暗号をクラックするために、ブルートフォース復号が必要となります。192.168.1.0/24 IP 範囲にあるノードから 192.168.2.0/24 範囲にあるもう一つのノードへとコミュニケートされるプロセスは、IPsec パケットの処理、暗号化/復号、およびルーティングが完全に IPsec ルーターで取り扱われるので、ノードに対して完全に透過的です。

ネットワーク－ネットワーク間接続に必要とされる情報は以下を含みます:

- 専用の IPsec ルーターの外部からアクセス可能な IP アドレス
- IPsec ルーターにより取り扱われる LAN/WAN のネットワーク・アドレス範囲 (192.168.1.0/24 や 10.0.1.0/24 のような) 。
- ネットワーク・ノードからインターネットへデータをルートするゲートウェイ・デバイスの IP アドレス
- 一意な名前。たとえば、ipsec1。これは IPsec コネクションを識別し、他のデバイスまたはコネクションと区別するために使用されます。
- 固定暗号キーまたは racoon により自動的に生成されたもの
- コネクションの初期化段階で使用され、セッション中に暗号キーを交換するために事前共有された認証キー。

4.2.1.7.1. ネットワーク－ネットワーク間の (VPN) コネクション

ネットワーク－ネットワーク間の IPsec コネクションは、プライベート・サブネットに対するネットワーク・トラフィックがルートされることを通して、お互いのネットワークのために、2つの IPsec ルーターを使用します。

たとえば、*#4.3################ IPsec#* に示されるように、192.168.1.0/24 プライベート・ネットワークが 192.168.2.0/24 プライベート・ネットワークにネットワーク・トラフィックを送信するならば、パケットは gateway0 を通り、ipsec0 へと、インターネットを経由して、ipsec1 へと、gateway1 へと、192.168.2.0/24 サブネットへと行きます。

IPsec ルーターは、公にアドレス可能な IP アドレスとそれぞれのプライベート・ネットワークに接続された2番目のイーサネット・デバイスを必要とします。もう一方の IPsec ルーターが暗号化されたコネクションを持つことを意図しているならば、トラフィックは IPsec ルーターを経由していきます。

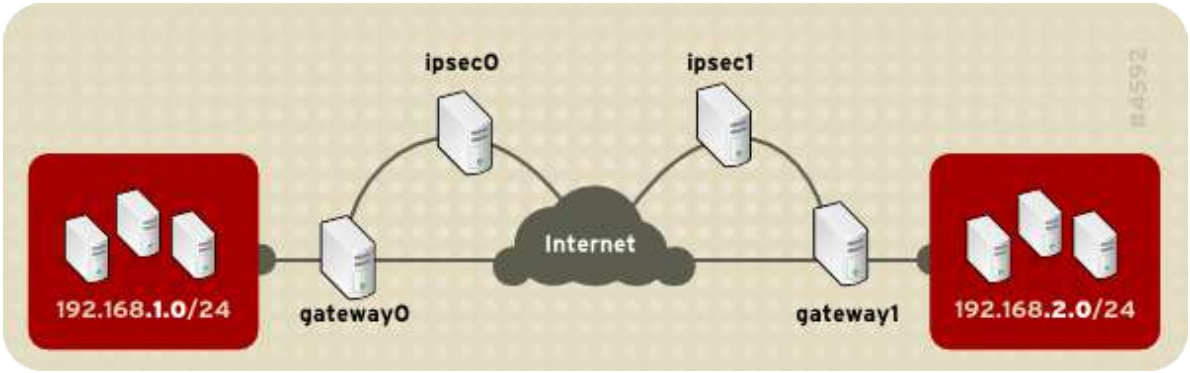

図4.3 ネットワーク−ネットワーク間の IPsec

代わりのネットワーク設定オプションは、各 IP ルーターおよびインターネット間のファイアウォール、各 IPsec ルーターおよびサブネット・ゲートウェイ間のイントラネット・ファイアウォールを含みます。IPsec ルーターおよびサブネットのゲートウェイは、2つのイーサネット・デバイスを持つ1つのシステムです。IPsec ルーターとして動作する パブリック IP アドレスを持つもの、およびプライベート・サブネットに対するゲートウェイとして動作するプライベート IP アドレスを持つもの。各 IPsec ルーターは、プライベート・ネットワークのためにゲートウェイを使用します。もしくは、他の IPsec ルーターにパケットを送るためにパブリック・ゲートウェイを使用します。

ネットワーク−ネットワーク間の IPsec コネクションを設定するために以下の手順を使用します:

1. ネットワーク管理ツールを起動するために、コマンド・シェルにおいて `system-config-network` と入力します。

2. IPsec設定ウィザードを起動するために、IPsec タブにおいて新規をクリックします。

3. ネットワーク−ネットワーク間の IPsec コネクションを設定開始するために進むをクリックします。

4. コネクションの一意なニックネームを入力します。たとえば、`ipsec0`。必要に応じて、コンピュータを起動するときに自動的にコネクションを有効にするチェックボックスを選択します。続けるために、進むをクリックします。

5. コネクションの種類としてネットワーク−ネットワーク間暗号化 (VPN) を選択します。進むをクリックします。

6. 使用する暗号化の種類を選択します: 手動または自動。

   手動暗号化を選択すると、暗号キーが後のプロセスにおいて提供されなければいけません。自動暗号化を選択すると、`racoon` デーモンが暗号キーを管理します。自動暗号化を使用したいならば、`ipsec-tools` パッケージがインストールされていなければいけません。

   続けるために進むをクリックします。

7. ローカルネットワークページにおいて、以下の情報を入力します:

   - ローカルネットワークアドレス — プライベート・ネットワークに接続された、IPsec ルータにおけるデバイスの IP アドレス。
   - ローカルサブネットマスク — ローカルネットワーク IP アドレスのサブネットマスク。
   - ローカルネットワークゲートウェイ — プライベートサブネットのゲートウェイ。

   続けるために進むをクリックします。

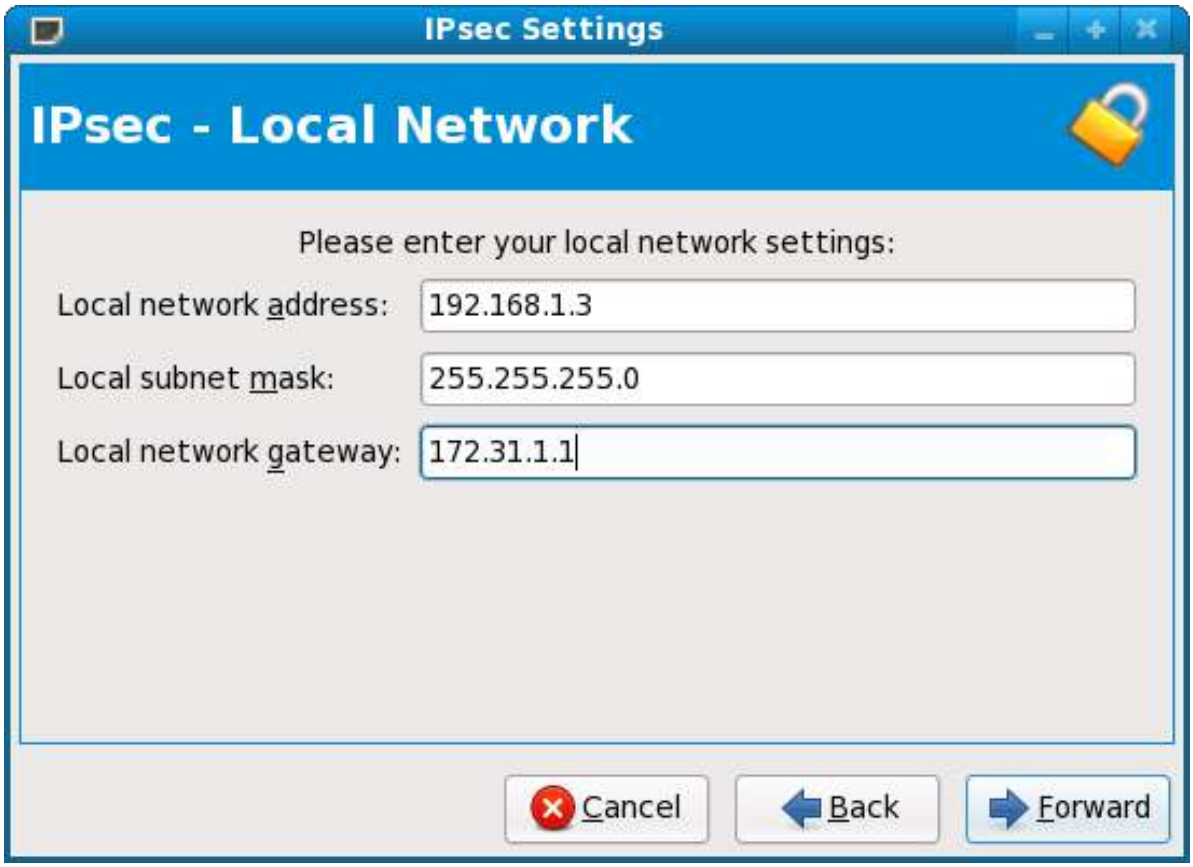

図4.4 ローカル・ネットワーク情報

8. リモートネットワークページにおいて、以下の情報を入力します:

   - リモート IP アドレス — ##プライベート・ネットワークに対してIPsec ルーターのパブリックアドレス可能な IP アドレス。私たちの例では、ipsec0 に対して、ipsec1 のパブリックにアドレス可能な IP アドレスを入力します。逆もまた同様です。
   - リモート・ネットワーク・アドレス — ## IPsec ルーターにバインドされたプライベート・サブネットのネットワーク・アドレス。私たちの例では、ipsec1 を設定しているなら 192.168.1.0 を入力します。ipsec0 を設定しているなら 192.168.2.0 を入力します。
   - リモート・サブネットマスク — リモート IP アドレスのサブネットマスク。
   - リモート・ネットワーク・ゲートウェイ — リモート・ネットワーク・アドレスに対するゲートウェイの IP アドレス。
   - *6*の手順において手動暗号化が選択されると、使用する暗号キーを指定するか、1つ生成するために生成をクリックします。

     認証キーを指定するか、1つ生成するために生成をクリックします。このキーは数字と文字のあらゆる組み合わせが可能です。

   続けるために進むをクリックします。

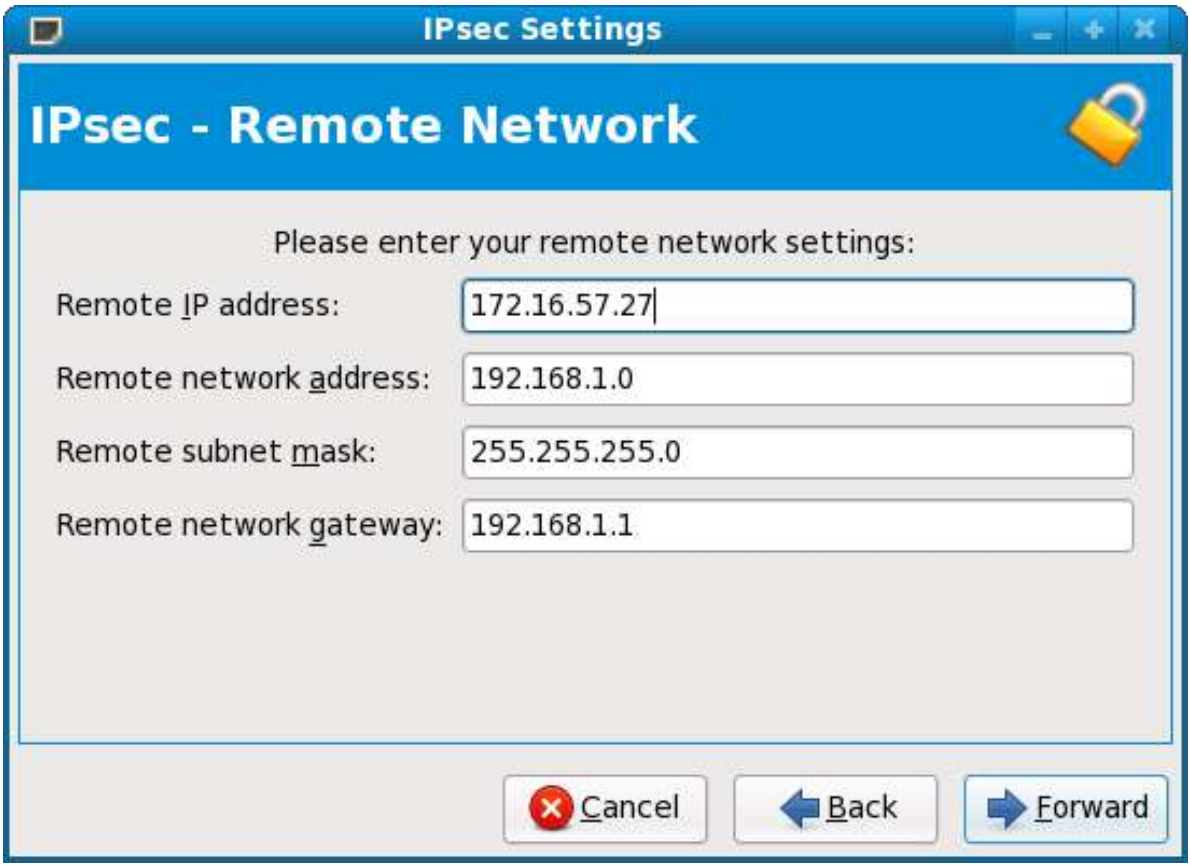

図4.5 リモート・ネットワーク情報

9. IPsec — Summary ページにおいて情報を確認し、Apply をクリックします。

10. 設定を保存するために ファイル => 保存 を選択します。

11. リストから IPsec コネクションを選択し、コネクションを有効にするために有効をクリックします。

12. IP フォワードを有効にします:

    a. `/etc/sysctl.conf` を編集し、`net.ipv4.ip_forward` に 1 をセットします。

    b. 変更を有効にするために以下のコマンドを使用します

```
[root@myServer ~]# /sbin/sysctl -p /etc/sysctl.conf
```

IPsec コネクションを有効化するネットワーク・スクリプトは、必要に応じて IPsec ルーターを通してパケットを送るために、ネットワーク・ルーターを自動的に作成します。

4.2.1.7.2. 手動の IPsec ネットワーク—ネットワーク間の設定

LAN A (lana.example.com) および LAN B (lanb.example.com) がお互いに IPsec トンネルを経由して接続したいと仮定します。LAN A のネットワーク・アドレスは 192.168.1.0/24 範囲にあり、LAN B は 192.168.2.0/24 範囲を使用します。ゲートウェイ IP アドレスは、LAN A に対して 192.168.1.254 、LAN B に対して 192.168.2.254 です。IPsec ルーターは、各 LAN ゲートウェイから分離されており、2つのネットワーク・デバイスを使用します: eth0 はインターネットからアクセスする外部からアクセス可能な静的 IP アドレスを割り当てられています。一方、eth1 は処理するルーティング地点として動作して、あるネットワーク・ノードからリモート・ネットワーク・ノードへと LAN パケットを転送します。

各ネットワークの間の IPsec コネクションは、r3dh4tl1nux の値を持つ事前共有キーを使用します。また、A と B の管理者は、IPsec ルーターの間の認証キーをracoon が自動的に生成および管理することに合意します。LAN A の管理者が IPsec コネクションを ipsec0 と名付けることに決めます。一方、LAN B の管理者が IPsec コネクションを ipsec1 と名付けます。

以下の例は、LAN A に対するネットワーク−ネットワーク間 IPsec コネクションの ifcfg ファイルの内容を示します。この例においてコネクションを識別するための一意な名前は *ipsec0* です。そのため、結果ファイルは /etc/sysconfig/network-scripts/ifcfg-ipsec0 と呼ばれます。

```
TYPE=IPSEC
ONBOOT=yes
IKE_METHOD=PSK
SRCGW=192.168.1.254
DSTGW=192.168.2.254
SRCNET=192.168.1.0/24
DSTNET=192.168.2.0/24
DST=X.X.X.X
```

以下の一覧はこのファイルの内容を説明します:

TYPE=IPSEC
: 接続の種類を指定します。

ONBOOT=yes
: ブート時に接続が初期化されるかを指定します。

IKE_METHOD=PSK
: 接続が使用する認証の事前共有鍵の方式を指定します。

SRCGW=192.168.1.254
: 送信元ゲートウェイの IP アドレス。LAN Aに対しては LAN A のゲートウェイ、LAN B に対しては LAN B ゲートウェイ。

DSTGW=192.168.2.254
: 宛て先ゲートウェイの IP アドレス。LAN A に対しては LAN B のゲートウェイ、LAN B に対しては LAN A のゲートウェイ。

SRCNET=192.168.1.0/24
: IPsec 接続に対する送信元ネットワークを指定します。この例では LAN A のネットワーク範囲です。

DSTNET=192.168.2.0/24
: IPsec 接続に対する宛て先ネットワークを指定します。この例では LAN A のネットワーク範囲です。

DST=X.X.X.X
: LAN B の外部からアクセス可能な IP アドレス。

以下の例は、両方のネットワークがお互いに認証するために使用する、/etc/sysconfig/network-scripts/keys-ipsec***X*** (*X* は、LAN A に対して0、LAN B に対して1です)と呼ばれる事前共有キーファイルの内容です。このファイルの内容は同じであるべきです。また、root ユーザーだけがこのファイルを読み書きできるべきです。

```
IKE_PSK=r3dh4tl1nux
```

重要

root ユーザーだけが keys-ipsec*X* ファイルを読み込みや編集ができるよう、そのファイルを変更するため、ファイル作成後に以下のコマンドを使用します:

```
chmod 600 /etc/sysconfig/network-scripts/keys-ipsec1
```

いつでも認証キーを変更するために、両方の IPsec ルーターにおいて keys-ipsec*X* ファイルを編集します。##################################。

以下の例は、IPsec コネクションに対する /etc/racoon/racoon.conf 設定ファイルの内容です。ファイルの最後にある include 行は、自動的に生成され、IPsec トンネルが実行しているときのみ表れます。

```
# Racoon IKE daemon configuration file.
# See 'man racoon.conf' for a description of the format and entries.
path include "/etc/racoon";
path pre_shared_key "/etc/racoon/psk.txt";
path certificate "/etc/racoon/certs";

sainfo anonymous
{
 pfs_group 2;
 lifetime time 1 hour ;
 encryption_algorithm 3des, blowfish 448, rijndael ;
 authentication_algorithm hmac_sha1, hmac_md5 ;
 compression_algorithm deflate ;
}
include "/etc/racoon/X.X.X.X.conf"
```

以下はリモート・ネットワークへの接続用の具体的な設定ファイルです。このファイルは ***X.X.X.X***.conf という名前です（ここで、*X.X.X.X* は リモート IPsec ルーターの IP アドレスです）。このファイルは、IPsec トンネルが有効化されるときに自動的に生成され、直接編集すべきではありません。

```
remote X.X.X.X{
        exchange_mode aggressive, main;
 my_identifier address;
 proposal {
  encryption_algorithm 3des;
  hash_algorithm sha1;
  authentication_method pre_shared_key;
  dh_group 2 ;
 }
}
```

IPsec 接続を開始するに先立って、IP フォワーディングがカーネルで有効になっていなければいけません。IP フォワーディングを有効にするために:

1. /etc/sysctl.conf を編集し、net.ipv4.ip_forward に 1 をセットします。
2. 変更を有効にするために以下のコマンドを使用します

```
[root@myServer ~] # sysctl -p /etc/sysctl.conf
```

IPsec 接続を開始するために、各ルーターにおいて以下のコマンドを使用します。

```
[root@myServer ~] # /sbin/ifup ipsec0
```

コネクションが有効化され、LAN A と LAN B 両方がお互いにコミュニケートできます。IPsec コネクションにおいて ifup を実行することにより、ルートが初期化スクリプト経由で自動的に作成されます。

```
[root@myServer ~] # /sbin/ip route list
```

IPsec コネクションをテストするために、ホスト間で転送されるネットワーク・パケットを表示するために、外部からルート可能なデバイスにおいて tcpdump ユーティリティを実行して、IPsec 経由で暗号化されていることを確認します。たとえば、LAN A の IPsec コネクションをチェックするために、以下のコマンドを使用します:

```
[root@myServer ~] # tcpdump -n -i eth0 host lana.example.com
```

パケットは AH ヘッダーを含むべきであり、ESP パケットとして示されるべきです。ESP は暗号化されているという意味です。たとえば、(バックスラッシュはある行が続くことを表します):

```
12:24:26.155529 lanb.example.com > lana.example.com: AH(spi=0x021c9834,seq=0x358): ¥
 lanb.example.com > lana.example.com: ESP(spi=0x00c887ad,seq=0x358) (DF) ¥
 (ipip-proto-4)
```

4.2.1.8. IPsec コネクションの開始と停止

IPsec コネクションがブート時に有効化するよう設定されていなければ、コマンドラインから制御できます。

コネクションを開始するために、ホスト間 IPsec に対する各ホスト、またはネットワーク間 IPsec に対する各 IPsec ルータにおいて以下のコマンドを使用します。

```
[root@myServer ~] # /sbin/ifup <nickname>
```

ここで <*nickname*> は、ipsec0 のような、前に設定したニックネームです。

接続を停止するために、以下のコマンドを使用します:

```
[root@myServer ~] # /sbin/ifdown <nickname>
```

4.2.2. Secure Shell

Secure Shell (SSH) はセキュアなチャネル上で他のシステムとコミュニケーション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上の転送は、暗号化され、盗聴から保護されます。暗号のログオンは、伝統的なユーザー名とパスワード上のよりよい認証方法を提供するために活用されます。

SSH は有効にすることが非常に簡単です。単に sshd サービスを開始することで、システムは接続を受け付けるようになり、正しいユーザー名とパスワードが接続プロセスの間に提供されるとき、システムへのアクセスを許可します。SSH サービスに対する標準的な TCP ポートは 22 です。しかしながら、設定ファイル */etc/ssh/sshd_config* を修正して、サービスを再起動することで、これを変更できます。このファイルは SSH に対する他の設定オプションも含みます。

Secure Shell (SSH) は、1つのポートを用いるだけでなく、コンピューター間の暗号化されたトンネルも提供します。*########## SSH ##########*[1] 、トラフィックがそのトンネルを通過するので暗号化されますが、ポートフォワードを使用することが VPN と同じくらい流動的なわけではありません。

[1] http://www.redhatmagazine.com/2007/11/27/advanced-ssh-configuration-and-tunneling-we-dont-need-no-stinking-vpn-software

4.2.2.1. 暗号によるログオン

SSH はコンピューターにログインするために暗号鍵の使用をサポートしています。これはパスワードを用いるよりもはるかに安全です。もし正しくセットアップされれば、複数要素認証を検討できます。

暗号化によるログオンをできる前に設定の変更を行う必要があります。ファイル /etc/ssh/sshd_config において、以下の行を次のようにアンコメントして変更します:

```
PubkeyAuthentication yes
AuthorizedKeysFile .ssh/authorized_keys
```

最初の行により SSH プログラムが公開鍵認証できるように指示します。2 行目は、認可されたキーペアの公開鍵が存在する、システムのホームディレクトリーにあるファイルを示します。

次に行うことは、システムに接続するために使用する、クライアントの SSH キーペアを生成することです。コマンド ssh-keygen はシステムにログインするために設定する RSA 2048-bit キーを生成します。キーは標準状態で ~/.ssh ディレクトリーに保存されます。キーのビット長を変更するにはスイッチ -b を利用できます。2048-bits はおそらく問題ないですが、可能ならば 8192-bit キーまで拡張できます。

~/.ssh ディレクトリーにおいて、作成された二つのキーを確認すべきです。ssh-keygen を実行するとき初期値を使うならば、キーは秘密鍵と公開鍵に id_rsa および id_rsa.pub という名前がつけられます。常に秘密鍵がさらされることから保護するべきです。しかしながら、公開鍵はログインしようとしているシステムに転送する必要があります。一度システムにおくならば、キーを承認リストに追加する最も簡単な方法は、次の方法です:

```
$ cat id_rsa.pub >> ~/.ssh/authorized_keys
```

これは、公開鍵を authorized_key ファイルに追加します。SSH アプリケーションは、ログインを試行するときに、このファイルを確認します。

パスワードや他の認証方式と同じように、定期的に SSH キーを変更すべきです。その際、すべての使用していない鍵を authorized_key ファイルから確実に削除しておきます。

4.2.3. LUKS ディスク暗号化

Linux Unified Key Setup-on-disk-format (or LUKS) は、Linux コンピューターのパーティションを暗号化できるようにします。これはとくに、モバイル・コンピューターやリムーバブル・メディアを使うときに重要です。LUKS は複数のユーザー・キーがパーティションの全体暗号化に対して使用されるマスター・キーを復号できるようにします。

4.2.3.1. Fedora における LUKS 実装

Fedora は、システムシステムの暗号化を実行するために LUKS を利用します。デフォルトで、ファイルシステムを暗号化するオプションはインストール中にチェックされていません。ハードディスクを暗号化するオプションを選択すると、コンピュータを起動するたびにパスフレーズが尋ねられます。このパスフレーズは、パーティションを復号するために用いられる全体暗号鍵を "ロック解除" します。デフォルトのパーティション・テーブルを変更するために選択すると、暗号化したいパーティションを選択できます。これは、パーティション・テーブルの設定にセットされます。

Fedora のデフォルト LUKS 実装は SHA256 ハッシュを持つ AES 128 です。利用可能な暗号は次のとおりです:

- AES - Advanced Encryption Standard - *FIPS PUB 197*[2]

[2] http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf

- Twofish (A 128-bit Block Cipher)
- Serpent
- cast5 - *RFC 2144*[3]
- cast6 - *RFC 2612*[4]

4.2.3.2. ディレクトリの手動暗号化

警告

この手順に従うと、暗号化するパーティションにあるすべてのデータが削除されます。すべての情報を失うでしょう!この手順を始める前にデータを外部ソースへ確実にバックアップしてください!

注記

この手順は、パーティションにある既存のデータを削除して、使用する LUKS 向けの乱数ベースを提供するために、*scrub* を使用します。この乱数ベースは暗号化に対する特定の攻撃を防ぐために重要です。*scrub* は標準ではインストールされていません。使用する前にインストールする必要があります。代わりに、同じことを達成するために、他の乱数生成器を使用することもできます。

手動でパーティションを暗号化したければ、以下の指示が役に立ちます。以下のサンプル・デモは /home パーティションを暗号化しますが、どんなパーティションでも使用できます。

以下の手順は既存のデータをすべて取り去るでしょう。そのため、始める前にテストされたバックアップを確実にします。/home が独立したパーティションである必要があります(ここでは /dev/VG00/LV_home です)。以下はすべて root として実行されなければいけません。これら手順の失敗はすべて、手順が成功するまで進んではいけません。

4.2.3.3. ステップ・バイ・ステップの説明

1. ランレベル 1 に入ります: `telinit 1`
2. パーティションをランダムデータで埋めます: `scrub -p random /home`
3. 既存の /home をアンマウントします: `umount /home`
4. もし失敗したなら、/home を独占しているプロセスを見つけて止めるために `fuser` を使用します: `fuser -mvk /home`
5. /home がもうマウントされていないことを確認します: `cat /proc/mounts | grep home`
6. パーティションを初期化します: `cryptsetup --verbose --verify-passphrase luksFormat /dev/VG00/LV_home`

[3] http://www.ietf.org/rfc/rfc2144.txt
[4] http://www.ietf.org/rfc/rfc2612.txt

7. 新しく暗号化されたデバイスを開きます: `cryptsetup luksOpen /dev/VG00/LV_home home`
8. そこにあ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. マウントします: `mount /dev/mapper/home /home`
11. 見えることを確認します: `df -h | grep home`
12. 以下を /etc/crypttab に追加します: `home /dev/VG00/LV_home none`
13. /etc/fstab を編集して、/home の古いエントリを削除して、`/dev/mapper/home /home ext3 defaults 1 2` を追加します。
14. fstab エントリを確認します: `mount /home`
15. デフォルトの SELinux セキュリティ・コンテキストを復元します: `/sbin/restorecon -v -R /home`
16. 再起動します: `shutdown -r now`
17. /etc/crypttab にあるエントリは、コンピューターがブート時に luks パスフレーズを問い合わせるようにします。
18. root としてログインして、バックアップを復元します。

4.2.3.4. ただ何を達成したでしょうか。

おめでとうございます、これでコンピューターをオフにしている間も安全に保管できるよう、すべてのデータに対する暗号化されたパーティションを持ちました。

4.2.3.5. 興味のリンク

LUKS や暗号化ハードディスクに関する詳細は、以下のリンクを参照してください:

- *LUKS - Linux Unified Key Setup*[5]
- *HOWTO: Creating an encrypted Physical Volume (PV) using a second hard drive, pvmove, and a Fedora LiveCD*[6]

4.2.4. 7-Zip 暗号化アーカイブ

7-Zip[7] は、アーカイブのコンテンツを保護するために強力な暗号（AES-256）も使用できる、クロスプラットフォームで次世代のファイル圧縮ツールです。異なるオペレーティングシステム（たとえば、自宅の Linux と会社の Windows）を使用する複数のコンピュータ間でデータを移送する必要があり、持ち運び可能な暗号化ソリューションが欲しいとき、これは非常に有用です。

4.2.4.1. 7-Zip インストール

7-Zip は Fedora の基本パッケージではありませんが、ソフトウェアリポジトリで入手可能です。一度インストールすると、特別な注意を必要とせず、パッケージはコンピューターにおいてソフトウェアの残りの部分を更新するでしょう。

[5] https://code.google.com/p/cryptsetup/
[6] https://bugzilla.redhat.com/attachment.cgi?id=161912
[7] http://www.7-zip.org/

4.2.4.2. ステップ・バイ・ステップのインストールの説明

- 端末を開きます: アプリケーション -> システムツール -> 端末をクリックします。または、GNOME 3 においてアクティビティ -> アプリケーション -> 端末をクリックします。
- sudo アクセスで 7-Zip をインストールします: `sudo yum install p7zip`
- 端末を閉じます: `exit`

4.2.4.3. ステップ・バイ・ステップの使い方の説明

以下のこれらの説明により、"ドキュメント" ディレクトリを圧縮したり暗号化したりすることができるでしょう。元の "ディレクトリ" はそのまま残ります。この技術はファイルシステムにおいてアクセスすることができるすべてのディレクトリとファイルに適用できます。

- 端末を開きます:アプリケーション -> システムツール -> 端末 をクリックします
- 圧縮または暗号化します: (プロンプトが出たときにパスワードを入力します) `7za a -mhe=on -ms=on -p Documents.7z Documents/`

これで "ドキュメント" ディレクトリが圧縮され暗号化されます。以下の説明はアーカイブをどこか新しい場所に移動して、それを解凍します。

- 新しいディレクトリを作成します: `mkdir newplace`
- 暗号化ファイルを移動します: `mv Documents.7z newplace`
- 新しいディレクトリへ移動します: `cd newplace`
- ファイルを解凍します: (プロンプトが出たときにパスワードを入力します) `7za x Documents.7z`

これでアーカイブは新しい場所に解凍されます。以下の説明はこれまでのステップをすべてクリーンアップして、その前の状態にコンピュータを復元します。

- ディレクトリを上がります: `cd ..`
- テストのアーカイブとテストの解凍したものを削除します: `rm -r newplace`
- 端末を閉じます: `exit`

4.2.4.4. GUI からセキュアな 7-Zip アーカイブを作成する方法

7-Zip アーカイブは他のいろいろなアーカイブと同じように GUI から展開できます。しかし、セキュアな 7-Zip アーカイブを作成するには、いくつかの手順が必要です。

以下のこれらの説明により、"ドキュメント" ディレクトリを圧縮したり暗号化したりすることができるでしょう。元の "ディレクトリ" はそのまま残ります。この技術はファイルシステムにおいてアクセスすることができるすべてのディレクトリとファイルに適用できます。

- ファイルブラウザーを開きます: アクティビティ -> ファイル をクリックします
- "ドキュメント" フォルダーを右クリックします
- "圧縮" オプションを選択します
- ファイル拡張子として ".7z" を選択します
- "他のオプション" を展開します
- "ファイル一覧も暗号化する" をクリックします

- パスワードの項目にパスワードを入力します
- "作成" ボタンをクリックします

これでホームディレクトリーに "Documents.7z" ファイルができたことを確認できます。ファイルを開きたいならば、アーカイブの内容が表示される前にアーカイブのパスワードを尋ねられます。一度正しいパスワードが提供されると、ファイルが開きます。そうすると、アーカイブは通常通り操作できます。"Documents.7z" ファイルを削除することにより、この練習を終えて、コンピューターを元の状態に戻します。

4.2.4.5. 重要なこと

7-Zip は Microsoft Windows や Mac OS X にデフォルトで同梱されていません。それらのプラットフォームにおいて 7-Zip ファイルを使用したいならば、それらのコンピュータに適切なバージョンの 7-Zip をインストールする必要があります。7-Zip *download page*[8] を参照してください。

4.2.5. GNU Privacy Guard (GnuPG) の使用

GnuPG (GPG) は、あなた自身を識別したり、あなたのコミュニケーション（あなたが知らない人とのものを含みます）を認証したり、するために使われます。GPG は GPG 署名された email を読んだ人がその真正性を検証できるようにします。言い換えると、GPG は、あなたにより署名されたコミュニケーションが実際にあなたからであることを、かなり確かであることを可能にします。第三者がコードを変更したり、会話を横取りしたり、メッセージを変更したりするのを防ぐ助けになるので、GPG は有用です。

GPG は、コンピューターやネットワーク・ドライブに保存されているファイルを署名かつ/または暗号化するために使うこともできます。これにより、ファイルが認可されていない人により改ざんまたは読み込まれるのを防ぐという、さらなる保護を追加できます。

To utilize GPG for authentication or encryption of email you must first generate your public and private keys. After generating the keys you will have to setup your email client to utilize them.

4.2.5.1. GNOME における GPG キーの生成

Seahorse ユーティリティにより GPG 鍵管理が容易になります。コマンド `su -c "yum install seahorse"` またはソフトウェアの追加/削除を使用した GUI において、*Seahorse* をインストールできます。

鍵を作成するには パスワードと暗号鍵 を選択します。これによりアプリケーション Seahorse が起動します。ファイルメニューから新規を選択します。そして、PGP キーを選択し、続けるを選択します。あなたが誰であるかを表す、フルネーム、電子メールアドレス、およびオプションのコメント (例: John C. Smith, jsmith@example.com, 男性) を入力します。作成を選択します。鍵のパスフレーズを問い合わせるダイアログが表示されます。強力なパスフレーズですが、覚えやすいものを選択します。OK をクリックすると、鍵が作成されます。

警告

パスフレーズを忘れると、鍵を使うことができなくなり、その鍵を用いて暗号化されたデータが失われます。

[8] http://www.7-zip.org/download.html

GPG キー ID を見つけるために、新しく生成された鍵の次に "キー ID" 列を見ます。多くの場合、キー ID を要求すると、"0x6789ABCD" のように鍵 ID の前に "0x" がつきます。秘密鍵のバックアップをとり、どこか安全な場所に保管するべきです。

4.2.5.2. KDE における GPG キーの生成

メインメニューから アプリケーション > ユーティリティ > 暗号ツールを選択して KGpg プログラムを起動します。これまで KGpg を使用したことがなければ、プログラムがあなた自身の GPG 鍵ペアを生成するプロセスを詳しく説明します。ダイアログボックスは、新しい鍵ペアを生成するためのプロンプトを表示します。名前、電子メールアドレス、およびオプションのコメントを入力します。鍵の長さ(ビット数)とアルゴリズム同様、鍵が失効するまでの時間も選択できます。次のダイアログはパスフレーズに対するプロンプトが表示されます。このとき、あなたの鍵がメインの 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、鍵を使うことができなくなり、その鍵を用いて暗号化されたデータが失われます。

GPG キー ID を見つけるために、新しく生成された鍵の次に "キー ID" 列を見ます。多くの場合、キー ID を要求すると、"0x6789ABCD" のように鍵 ID の前に "0x" がつきます。秘密鍵のバックアップをとり、どこか安全な場所に保管するべきです。

4.2.5.3. コマンドラインを用いた GPG 鍵の生成

次のシェルコマンドを使用します: `gpg --gen-key`

このコマンドは、公開鍵と秘密鍵で構成される鍵ペアを生成します。他の人々は、あなたのコミュニケーションを認証かつ/または復号するためにあなたの公開鍵を使用します。できる限り、あなたの公開鍵を配布します(とくにメーリングリストのように、あなたから認証されたコミュニケーションを受け取りたいと考える、あなたが知っている人に対して)。

一連のプロンプトがプロセスを通してあなたに指示をします。必要に応じて初期値を割り当てるために Enter キーを押します。1番目のプロンプトは、あなたが必要とする鍵の種類を選択するよう尋ねます。

```
作成したい鍵の種類を選択してください:
      (1) RSA および RSA (初期値)
      (2) DSA および Elgamal
      (3) DSA (署名のみ)
      (4) RSA (署名のみ)
      どれにしますか?
```

ほとんどすべての場合において、初期値が正しい選択です。RSA 鍵は通信を署名するだけではなく、ファイルを暗号化できます。

次に、鍵の大きさを選択します:

```
RSA 鍵は 1024 ～ 4096 ビットの長さにできます。
鍵の大きさをどうしますか? (2048)
```

再び、初期値はほとんどすべてのユーザーにとって十分です。強いレベルのセキュリティを意味します。

次に、鍵がいつ失効するのかを選択します。標準の "none" を使用する代わりに、失効日を選択することは素晴らしい考えです。たとえば、鍵にある電子メールアドレスが無効になると、失効日により他者が公開鍵の使用を止めることに気がつきます。

```
鍵の有効期間を指定してください。
      0 = 失効しません
      d = n 日後に失効します
      w = n 週間後に失効します
      m = n か月後に失効します
      y = n 年後に失効します
      鍵をどの期間だけ有効にしますか? (0)
```

たとえば、1y の値を入力すると、鍵が1年間有効になります。(もし気が変わると、鍵を生成した後でこの失効日を変更できます。)

gpg プログラムは署名情報を尋ねる前に、以下のプロンプトが表れます: `Is this correct (y/n)?` プロセスを終わらせるために、y を入力します。

次に、名前と電子メールアドレスを入力します。このプロセスは実在の個人として認証することに関するものであると覚えてください。このため、実際の名前を含めます。アイデンティティを偽装するかわかりにくくするので、エイリアスやハンドルを使いません。

GPG キーの電子メール実アドレスを入力します。偽の電子メールアドレスを選択すると、他者があなたの公開鍵を見つけることがより難しくなります。これはコミュニケーションを認証することを難しくします。たとえば、メーリングリストにおいて [[DocsProject/SelfIntroduction| self-introduction]] に対してこの GPG キーを使用していると、そのリストにおいて使用する電子メールアドレスを入力します。

コメント・フィールドをエイリアスや他の情報を含めるために使用します。(ある人々は異なる目的に対して異なる鍵を使用します。そして、"オフィス" や "オープンソース・プロジェクト" のようなコメントを用いてそれぞれの鍵を識別します。)

すべてのエントリが正しければ、確認プロンプトにおいて、続けるために文字 O を入力します。もしくは、ある問題を修正するために他のオプションを使用します。最後に、秘密鍵に対するパスフレーズを入力します。gpg プログラムはパスフレーズを2回入力するよう尋ね、入力エラーがないことを確実にします。

最終的に、gpg はできる限り一意な鍵を作るためにランダムなデータを生成します。プロセスをスピードアップするためにこの手順の間、マウスを動かします、ランダムなキーを打ちます、もしくはシステムにおいて他のタスクを実行します。この手順が完了すると、鍵が完成し、使用する準備ができます:

```
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]
```

鍵のフィンガープリントは、あなたの鍵のための短縮形の "署名" です。あなたの実際の公開鍵が改ざんされることなく受け取ったことを、他者へ確認できるようにします。このフィンガープリントを書き留めておく必要はありません。いつでもフィンガープリントを表示するために、このコマンドをあなたの電子メールアドレスに置き換えて使用します: `gpg --fingerprint jqdoe@example.com`

"GPG キー ID" は、公開鍵を識別する16進8文字からなります。上の例において、GPG キー ID は 1B2AFA1C です。多くの場合、キー ID を問い合わせると、"0x1B2AFA1C" にあるように、キー ID の前に "0x" がつきます。

警告

パスフレーズを忘れると、鍵を使うことができなくなり、その鍵を用いて暗号化されたデータが失われます。

4.2.5.4. Alpine での GPG の使用

電子メールクライアント *Alpine* または *Pine* を使用していると、*ez-pine-gpg* もダウンロードしてインストールする必要があります。このソフトウェアは現在 *http://business-php.com/opensource/ez-pine-gpg/* から入手可能です。一度 ez-pine-gpg をインストールすると、`~/.pinerc` ファイルを修正する必要があります。以下が必要となります:

1. /home/username/bin は、指定したインストール・パスで置き換えられるべきです。
2. 2箇所において、_RECIPIENTS_ の後にある gpg-identifier はあなたの GPG 公開鍵の識別子で置き換えられるべきです。ここであなた自身の GPG 識別子を含める理由は、”Alice" へと暗号化されたメッセージを送るならば、メッセージはあなたの公開鍵も用いて暗号化されるからです。もしこれをしなければ、送信済みフォルダにあるメッセージを開けなくなり、あなた自身が書いたことを思い出せなくなります。

このように見えるでしょう:

```
# This variable takes a list of programs that message text is piped into
# after MIME decoding, prior to display.
display-filters=_LEADING(”-----BEGIN PGP”)_ /home/max/bin/ez-pine-gpg-incoming

# This defines a program that message text is piped into before MIME
# encoding, prior to sending
sending-filters=/home/max/bin/ez-pine-gpg-sign _INCLUDEALLHDRS_,
    /home/username/bin/ez-pine-gpg-encrypt _RECIPIENTS_ gpg-identifier,
    /home/username/bin/ez-pine-gpg-sign-and-encrypt _INCLUDEALLHDRS_ _RECIPIENTS_ gpg-identifier
```

4.2.5.5. Evolution での GPG の使用

4.2.5.5.1. Evolution とに使用するための GPG の設定

Evolution において使用するための GPG を設定するために、Evolution メインメニューから選択し、左パネルにある ツール、設定... を選択し、メール・アカウントを選択します。右パネルにおいて、使用する電子メールのアカウントを選択します。そして、編集ボタンを選択します。Evolution アカウント・エディタのダイアログが表示されます。セキュリティ・タブを選択します。

PGP/GPG キー ID フィールドにおいて、このアカウントの電子メールアドレスに対応する GPG キー ID を入力します。キー ID が何かはっきりしなければ、このコマンドを使用します: `gpg --fingerprint EMAIL_ADDRESS`。キー ID はキーのフィンガープリントの後ろ8文字 (4 バイト) と同じです。暗号メールを送信するときは必ず自分自身へと暗号化するオプションをクリックすることは良いアイディアです。このアカウントを使用するとき、出ていくメッセージを常に署名するを選択したいかもしれません。

注意

キーリングにおいて公開鍵を信頼されていると印をつけていないと、暗号化するときにキーリングにある鍵を常に信頼するオプションを選択するまで、それらの所有者への電子メールを暗号化することはできません。代わりに信頼性のチェックは失敗したことを意味するダイアログが表示されます。

4.2.5.5.2. Evolution を用いた電子メールの検証

Evolution は入ってくる GPG 署名されたメッセージの有効性を自動的にチェックします。Evolution が公開鍵が失われた(または、改ざんされた)ためにメッセージを GPG 検証できなければ、赤いバナーで終わります。メッセージが検証されたが、ローカルにもグローバルにもキーを署名してしなければ、バナーは黄色でしょう。

メッセージが検証されて、キーが署名されているならば、バナーは緑色でしょう。シール・アイコンをクリックするとき、Evolution は署名に関するセキュリティ情報をより持つダイアログを表示します。公開鍵をキーリングに追加するために、キーの所有者の電子メールアドレスで検索機能を使用します: `gpg --keyserver pgp.mit.edu --search email address`。正しいキーをインポートするために、Evolution により提供される情報を持つキー ID と一致させる必要があるかもしれません。

4.2.5.5.3. Evolution を用いた電子メールの署名と暗号化

電子メールに署名することにより、電子メールが本当にあなたからのものかどうかを、受信者が

電子メールを編集しているとき、セキュリティを選択し、メッセージを署名するために PGP 署名を選択します。メッセージを暗号化するために、PGP 暗号を選択します。同じように暗号化されたメッセージを署名するかもしれません。それはグッドプラクティスです。Evolution はあなたの GPG キーのパスフレーズを入力するよう促します。(3回失敗すると Evolution はエラーを発生させます。)このセッションのリマインダのためにこのパスワードを記録するオプションを選択すると、Evolution を終了するか再起動するまで、署名や復号するために再びパスワードを使う必要はありません。

4.2.5.6. Thunderbird を用いた GPG の使用

Fedora は thunderbird パッケージにおいて Mozilla Thunderbird を、また、mozilla-mail パッケージが Mozilla Suite email アプリケーションを含みます。Thunderbird は推奨された Mozilla email アプリケーションです。これは、デスクトップに アプリケーション > インターネット > Thunderbird Email として表れます。

Mozilla 製品は、メインのアプリケーションに新しい機能を追加するプラグインである拡張機能をサポートします。Enigmail 拡張は Mozilla の email 製品に GPG サポートを提供します。Enigmail のバージョンは、Mozilla Thunderbird と Mozilla Suite (Seamonkey) 両方に対して存在します。AOL の Netscape ソフトウェアは Mozilla 製品に基づき、この拡張も使用します。

Fedora システムに Enigmail をインストールするために、以下の説明に従います。

Enigmail は、メニュー項目とオプションにおいて OpenPGP という語を使用します。GPG は OpenPGP の実装であり、同じ意味として語を扱えます。

Enigmail のホームページは *http://enigmail.mozdev.org/download.html* です。

このページは Enigmail と GPG のアクションのスクリーンショットを提供します: *http://enigmail.mozdev.org/screenshots.html*.

4.2.5.6.1. Enigmail のインストール

Enigmail は Fedora リポジトリにおいて利用可能です。コマンドラインで `yum install thunderbird-enigmail` と入力することで、インストールできます。システム -> 管理 -> ソフトウェアの追加/削除により、*thunderbird-enigmail* をインストールできます。

4.2.5.7. 公開鍵暗号化について

1. *Wikipedia - Public Key Cryptography*[9]

2. *HowStuffWorks - Encryption*[10]

[9] http://en.wikipedia.org/wiki/Public-key_cryptography
[10] http://computer.howstuffworks.com/encryption.htm

情報セキュリティの一般原則

以下の一般原則は良いセキュリティ慣行の概要を提供します:

- 中間者攻撃や盗聴を防ぐ助けとするため、ネットワークで転送されるすべてのデータを暗号化します。パスワードのような認証情報を暗号化することは重要です。
- インストールされているソフトウェアと実行するサービスの量を最小限にします。
- セキュリティを強化するソフトウェアとツールを使用します。たとえば、強制アクセス制御（MAC）用の Security-Enhanced Linux (SELinux)、パケットフィルタリング（ファイアウォール）用の Netfilter iptables、ファイル暗号化用の GNU Privacy Guard (GnuPG) です。
- 可能ならば、ある危険にさらされたサービスが他のサービスを危険にさらすために使用されるリスクを最小限にするために、分離されたシステムにおいて各ネットワーク・サービスを実行します。
- ユーザーアカウントを維持します: 強いパスワードポリシーを作成して強制します; 使用していないユーザーアカウントを削除します
- システムとアプリケーションのログを定期的にレビューします。デフォルトで、セキュリティ関連のシステムログは `/var/log/secure` と `/var/log/audit/audit.log` に書き込まれます。注記: 専用のログサーバにログを送ることは、攻撃者が検知を避けるためにローカルのログを容易に修正することを防ぐ助けになります。
- 絶対に必要になるまで root としてログインしません。管理者は必要なときに root としてコマンドを実行するために sudo を使用することが推奨されます。sudo を実行できるユーザーは `/etc/sudoers` で指定されています。`/etc/sudoers` ファイルを編集するために `visudo` ユーティリティを使用します。

セキュアなインストール

セキュリティは Fedora をインストールするために CD や DVD をディスクドライブにいれた初めてのときから始まります。初めからシステムをセキュアに設定することにより、後から追加のセキュリティ設定を実装することがより簡単になります。

6.1. ディスク・パーティション

NSA は /boot, /, /home, /tmp, および /var/tmp に対して別々のパーティションを作成することを推奨しています。それぞれの理由は異なりますが、各パーティションに取り組みます。

/boot - このパーティションは、ブート中にシステムにより読み込まれる最初のパーティションです。システムを Fedora へブートするために使わ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すべきではありません。このパーティションが / に含まれていて、そのパーティションが暗号化されているか、もしくは別の理由で利用不能になるならば、システムはブートすることができなくなるでしょう。

/home - ユーザーデータ (/home) が独立したパーティションの代わりに / に保存されているとき、オペレーティングシステムが不安定になる原因となる、パーティションが一杯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 Fedora へアップグレードするとき、/home パーティションにあるデータをそのままにしたい場合、インストール中に上書きされないようすることが非常に簡単になります。root パーティション (/) が壊れると、データは永久に失われる可能性があります。このパーティションを頻繁なバックアップの対象にすることもできます。

/tmp および /var/tmp - /tmp と /var/tmp ディレクトリはどちらも長期間の保存が必要とされないデータを保存するために使われます。しかしながら、これらのディレクトリの1つが多くのデータであふれると、ストレージ空間をすべて消費する可能性があります。これが起こり、これらのディレクトリが / の中に保存されていると、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。このため、これらのディレクトリをそれ自身のパーティションに移動することは良いアイディアです。

6.2. LUKS パーティション暗号化の利用

Linux Unified Key Setup-on-disk-format[1] (LUKS) 暗号化の実装は近年より簡単になってきています。インストール・プロセス中に、パーティションを暗号化するオプションがユーザーへ表示されるでしょう。ユーザーは、パーティションのデータをセキュアにするために使われる大量の暗号鍵を解除するための鍵となるパスフレーズを供給しなければいけません。

[1] http://fedoraproject.org/wiki/Security_Guide/9/LUKSDiskEncryption

ソフトウェアのメンテナンス

ソフトウェアのメンテナンスはセキュアなシステムを維持するために非常に重要です。攻撃者がシステムに侵入するために既知のホールを使用するのを防ぐために、ソフトウェアのパッチが利用可能になり次第できる限り早く適用することは極めて重要です。

7.1. 最小限のソフトウェアのインストール

コンピュータにあるソフトウェアの各部品が脆弱性を含む可能性があるので、使用するパッケージだけをインストールすることがベストプラクティスです。もし DVD からインストールしているならば、インストール中にインストールしたいパッケージを正確に選択する機会があります。他のパッケージが必要であるとわかったときに、後からいつでもシステムへ追加することができます。

7.2. セキュリティ・アップデートの計画と設定

すべてのソフトウェアはバグを含みます。しばしば、これらのバグはシステムを悪意のあるユーザーにさらす可能性がある脆弱性となります。パッチを当てていないシステムはコンピューターの侵入の一般的な原因となります。侵入できないようそれらの脆弱性をふさぐために、タイムリーにセキュリティ・パッチをインストールする計画を持つべきです。

自宅のユーザーにとって、セキュリティ・アップデートはできる限り早くインストールされるべきです。セキュリティ・アップデートの自動インストールの設定は、覚えておかなければいけないのを避けられますが、あるものがシステムにおける設定または他のソフトウェアと競合する原因となる可能性があるというわずかなリスクをもたらします。

ビジネスや自宅の高度なユーザーにとって、セキュリティ・アップデートは、テストされ、インストールをスケジュールするべきです。パッチがリリースされてからシステムにインストールされるまでの間、システムを保護するために追加のコントロールを使う必要があります。これらのコントロールはその脆弱性に依存しますが、追加のファイアウォール・ルール、外部ファイアウォールの使用、およびソフトウェア設定の変更を含められます。

7.3. 自動更新の調整

Fedora は日次スケジュールですべてのアップデートを適用するよう設定されています。システムがどのようにアップデートをインストールするかを変更したい場合、"ソフトウェア・アップデートの設定" により実施しなければいけません。利用可能なアップデートを適用または通知するために、スケジュールとアップデートの種類を変更できます。

Gnome では、システム -> 設定 -> ソフトウェアの更新においてアップデートをコントロールできます。KDE では、アプリケーション -> 設定 -> ソフトウェアの更新にあります。

7.4. よく知られたリポジトリからの署名された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を通して公開されます。よく知られたリポジトリはすべてパッケージ署名をサポートしています。パッケージ署名は、リポジトリにより公開されているパッケージが、署名を適用されてから変更されていないことを証明するために、公開鍵の技術を使用します。これにより、パッケージが作成された後ユーザーがダウンロードする前に、悪意を持って変更されているかもしれないソフトウェアに対する保護が提供されます。

多く過ぎるリポジトリ、信頼できないリポジトリ、または署名のないパッケージを持つリポジトリを使用することは、システムに悪意または脆弱性のあるコードを取り込むリスクをより高くします。yum やソフトウェアの更新にリポジトリを追加するときに注意してください。

共通脆弱性識別子 CVE

共通脆弱性識別子または CVE システムは、一般に知られる情報セキュリティの脆弱性と暴露に対する参照方法を提供します。MITRE 社が、アメリカ国土安全保障省のサイバー・セキュリティ部門の資金提供を受け、システムを維持しています。

MITRE 社はすべての脆弱性と暴露に対して CVE 識別子を割り当てます。1つの CVE が複数のソフトウェアパッケージや複数のベンダーに影響する可能性があるため、ソフトウェアの異なる部分を通して脆弱性を追跡するために使われます。

8.1. YUM プラグイン

yum-plugin-security パッケージは Fedora の機能の一つです。インストールされていると、このパッケージにより提供される yum モジュールにより、セキュリティ関連の更新のみ取得するよう yum を制限できます。Red Hat アドバイザリーに関する情報を提供するために使用されます。これは、Red Hat の Bugzilla データベースにあるバグ、または MITRE の Common Vulnerabilities and Exposures ディレクトリーの CVE 番号が、パッケージ更新により示されます。

これらの機能を有効にすることは、`yum install yum-plugin-security` コマンドを実行するくらい簡単です。

SELinux

9.1. SELinux 概要

Security-Enhanced Linux (SELinux) は Linux カーネルにおける ######## 機構の実装です。これは、標準的な########が確認された後で、許可された操作であることを確認します。これは National Security Agency により作成されました。Linux システムにおいてファイルとプロセスに対して、また、定義されたポリシーに基づいてアクションに対してルールを強制できます。

SELinux を使用するとき、ディレクトリやデバイスを含む、ファイルをオブジェクトとして参照します。ユーザーが実行するコマンドや Mozilla® Firefox® アプリケーションのようなプロセスは、サブジェクトとして参照されます。多くのオペレーティングシステムは、サブジェクトがどのようにオブジェクトとやりとりするか、およびサブジェクトがどのようにお互いにやりとりするかを制御する、任意アクセス制御 (DAC: Discretionary Access Control) システムを使用します。DAC を使用するオペレーティングシステムにおいて、ユーザーが所有するファイル (オブジェクト) のパーミッションを制御します。たとえば、Linux® オペレーティングシステムにおいて、ユーザーが自身のホームディレクトリを全体読み込み可能にできます。ユーザーやプロセス (サブジェクト) が潜在的に機微な情報にアクセスする可能性があります。この望まないアクションにさらなら保護がありません。

DAC 機構のみに依存することは、強固なシステムセキュリティに基本的に適していません。DAC アクセス判定は、ユーザーと所有者のみに基づいており、ユーザーの役割、プログラムの関数および信頼性、およびデータの機密性や完全性のような、他のセキュリティ関連の情報を無視します。それぞれのユーザーが自身のファイルに関する完全な決定権を持ち、システム全体のセキュリティポリシーを強制することが不可能です。さらに、すべてのプログラムが、ユーザーに権限委譲されたパーミッションをすべて引き継ぐことにより実行されます。そして、これはユーザーのファイルに対するアクセス権を自由に変更します。そのため、悪意のあるソフトウェアに対して何も保護されません。多くのシステムサービスと特権プログラムが、その要件をはるかに超える荒い権限で実行する必要があります。そのため、これらのプログラムのどれかにおける欠陥がシステムの完全アクセス権を手に入れる危険にさらされる可能性があります。[1]

以下は、Security-Enhanced Linux (SELinux) を実行していない Linux オペレーティングシステムにおいて使用されるパーミッションの例です。これらの例におけるパーミッションと出力はお使いのシステムと異なるかもしれません。ファイルのパーミッションを表示するには `ls -l` コマンドを使用します:

```
$ ls -l file1
-rw-rw-r--. 1 user1 group1 0 May 11 10:46 file1
```

最初 3 つのパーミッションビット `rwx` が `file1` に対して Linux `user1` ユーザー (この場合、所有者) の持つアクセス権を制御します。次の 3 つのパーミッションビット `rw-` が、`file1` に対して Linux `group1` グループの持つアクセス権を制御します。最後の 3 つのパーミッションビット `r--` が、`file1` に対してすべてのユーザーが持つアクセス権を制御します。これは、すべてのユーザーとプロセスが含まれます。

DAC 機構は基本的に強固なシステムセキュリティに適していません。DAC アクセス判定は、ユーザーと所有者のみに基づいており、ユーザーの役割、プログラムの関数および信頼性、およびデータの機密性や完全性のような、他のセキュリティ関連の情報を無視します。それぞれのユーザーが自身のファイルに関する完全な決定権を持ち、システム全体のセキュリティポリシーを強制することが不可能です。さらに、すべてのプログラムが、ユーザーに権限委譲されたパーミッションをすべて引き継ぐことにより実行されます。そして、これはユーザーのファイルに対するアクセス権を自由に変更します。そのため、悪意のあるソフトウェアに対して何も保護されません。多くのシステムサービスと特権プログラムが、その要件をはるかに超える荒い権限で実行する必要があります。

[1] "Integrating Flexible Support for Security Policies into the Linux Operating System", by Peter Loscocco and Stephen Smalley. この論文は元々 National Security Agency のために準備され、公的分野において引き継がれています。初期リリースに関する詳細とドキュメントは#### [http://www.nsa.gov/research/_files/selinux/papers/freenix01/index.shtml]を参照してください。すべての編集と変更が Murray McAllister により実行されました。

そのため、これらのプログラムのどれかにおける欠陥がシステムの完全アクセス権を手に入れる危険にさらされる可能性があります。[2]

以下は、SELinux を実行する Linux オペレーティングシステムにおいて、プロセス、Linux ユーザー、およびファイルに使用されるセキュリティ関連情報を含むラベルの例です。この情報は SELinux ###### と呼ばれ、ls -Z コマンドを使用して表示されます:

```
$ ls -Z file1
-rw-rw-r--. user1 group1 unconfined_u:object_r:user_home_t:s0 file1
```

この例では、SELinux が (unconfined_u)、ロール (object_r)、タイプ (user_home_t) およびレベル (s0) を提供します。この情報はアクセス制御の判定のために使用されます。DAC を用いると、アクセス権が Linux ユーザーとグループ ID のみに基づいて制御されます。SELinux ポリシールールは DAC ルールの#に確認されることを覚えておくことが重要です。まず DAC ルールがアクセスを拒否すれば、SELinux ポリシールールが使用されません。

Linux と SELinux のユーザー

SELinux を実行する Linux オペレーティングシステムにおいて、Linux ユーザーと同じように SELinux ユーザーがあります。SELinux ユーザーは SELinux ポリシーの一部です。Linux ユーザーが SELinux ユーザーに対応づけられます。混乱を避けるため、このドキュメントは両者を区別するために、"Linux ユーザー" と "SELinux ユーザー" を使用します。

9.1.1. SELinux を実行する利点

- すべてのプロセスとファイルはタイプでラベルが付けられます。タイプがプロセス向けのドメインとファイル向けのタイプを定義します。SELinux ポリシールールが、どのようにプロセスがファイルとやりとりするか、およびどのようにプロセスが相互にやりとりするかについて定義します。それを明示的に許可する SELinux ポリシーが存在するときのみ、アクセスが許可されます。

- かなり精細なアクセス制御。ユーザーが任意に制御し、Linux ユーザーとグループ ID に基づく、伝統的な UNIX® パーミッションをはるかに超えて、SELinux は SELinux ユーザー、ロール、タイプおよびオプションのレベルのように、すべての利用可能な情報に基づいてアクセス権を判断します。

- SELinux ポリシーは、管理者により定義され、システム全体で強制され、ユーザーにより任意に設定されません。

- 権限を上昇される攻撃の脆弱性を減らします。ある例: プロセスがドメインで実行され、そのためお互いに分離されます、また、SELinux ポリシールールがどのようにプロセスがファイルや他のプロセスにアクセスするのかを定義します。そのため、プロセスが侵入されても、そのプロセスの通常の機能、およびプロセスがアクセスするよう設定されているファイルに対してのみ、攻撃者がアクセスできます。たとえば、Apache HTTP Server が侵入されても、SELinux ポリシールールがアクセスを許可するよう追加または設定されていなければ、攻撃者がユーザーのホームディレクトリにあるファイルを読み込むために、そのプロセスを使用できません。

- 制限されたサービス。サービスおよびデーモンがより予測可能になり、通常の操作に必要となるアクセスのみを許可されるように、SELinux がそれらを制限する機能を含みます。

- SELinux がデータの機密性および完全性を強制するために使用されます。また、プロセスを信頼されない入力から保護します。

[2] "Integrating Flexible Support for Security Policies into the Linux Operating System", by Peter Loscocco and Stephen Smalley. この論文は元々 National Security Agency のために準備され、公的分野において引き継がれています。初期リリースに関する詳細とドキュメントは#### [http://www.nsa.gov/research/_files/selinux/papers/freenix01/index.shtml]を参照してください。すべての編集と変更が Murray McAllister により実行されました。

SELinux は次のものではありません:

- ウイルス対策ソフトウェア。
- パスワード、ファイアウォール、または他のセキュリティシステムの代替。
- 総合的なセキュリティソリューション。

SELinux は既存のセキュリティソリューションを改善するために設計されています。それらを置き換えるものではありません。SELinux を実行しているときでも、ソフトウェアを最新に保つこと、推測しにくいパスワードを使うこと、ファイアウォールなどの良いセキュリティ慣行に従い続けることが重要です。

9.1.2. 例

以下の例は SELinux がどのようにセキュリティを向上させるかを説明します:

- 標準のアクションは拒否です。プロセスがファイルを開くなどのアクセスを許可する特定の SELinux ポリシールールが存在しなければ、ファイルアクセスが拒否されます。
- SELinux が Linux ユーザーを制限できます。数多くの制限された SELinux ユーザーが SELinux ポリシーに存在します。セキュリティルールとそれらに適用される機構の利点を得るために、Linux ユーザーが制限された SELinux ユーザーに対応づけられます。たとえば、Linux ユーザーを SELinux user_u ユーザーに対応づけることにより、Linux ユーザーが (他に設定されていない限り) `sudo` や `su` のような set user ID (setuid) アプリケーションを実行できなくなります。また、ホームディレクトリにあるファイルやアプリケーションを実行できなくなります - 設定されていると、ユーザーがホームディレクトリにある悪意のあるファイルを実行することを防ぎます。
- プロセス分離が使用されます。プロセスが自身のドメインで実行されます。これは、プロセスが他のプロセスにより使用されるファイルにアクセスすることを防ぎます。また、プロセスが他のプロセスにアクセスすることを防ぎます。たとえば、SELinux を実行しているとき、他に設定されていなければ、攻撃者が Samba サーバーに侵入できません。また、MySQL® により使用されるデータベースのような、他のプロセスにより使用されるファイルを読み書きするために、Samba サーバーに侵入することはできません。
- SELinux が設定ミスにより被害を制限する役に立ちます。*Domain Name System (DNS)*[3] サーバーがしばしば、お互いにゾーン転送として知られていることで情報を複製します。攻撃者が DNS サーバーを偽の情報で更新するためにゾーン転送を使用できます。Fedora で DNS サーバーとして *Berkeley Internet Name Daemon (BIND)*[4] を実行しているとき、管理者がゾーン転送を実行できるサーバーを制限し忘れたときさえ、標準の SELinux ポリシーがゾーンファイル [5] を BIND `named` 自身により、または他のプロセスにより、ゾーン転送経由で更新されることを防ぎます。
- *Red Hat® Magazine*[6] の記事 *Risk report: Three years of Red Hat Enterprise Linux 4*[78]、Red Hat® Enterprise Linux® 4 における標準の SELinux ターゲットポリシーのため、エクスプロイトが制限されました。

[3] http://en.wikipedia.org/wiki/Domain_Name_System
[4] https://www.isc.org/software/bind
[5] ホスト名と IP アドレスのマッピングのような情報を含み、DNS サーバーにより使用されるテキストファイル。
[6] http://www.redhatmagazine.com/
[7] http://www.redhatmagazine.com/2008/02/26/risk-report-three-years-of-red-hat-enterprise-linux-4/
[8] Cox, Mark. "Risk report: Three years of Red Hat Enterprise Linux 4". 2008年2月26日発行。2009年8月27日アクセス確認: *http://www.redhatmagazine.com/2008/02/26/risk-report-three-years-of-red-hat-enterprise-linux-4/*.

- SELinux に関する背景情報および、SELinux により防がれるさまざまなエクスプロイトに関する情報は、次の *LinuxWorld.com*[9] の記事を参照してください。*A seatbelt for server software: SELinux blocks real-world exploits*[10][11]

- Red Hat Enterprise Linux 4 および 5 に同梱された SELinux により低減された *OpenPegasus*[12] のエクスプロイトに関する詳細は、James Morris の *SELinux mitigates remote root vulnerability in OpenPegasus*[13] ブログ投稿を参照してください。

Tresys Technology[14] ウェブサイト (の右側) に *SELinux Mitigation News*[15] セクションがあります。これは SELinux により低減または防止された最近のエクスプロイトを一覧化しています。

9.1.3. SELinux アーキテクチャー

SELinux は Linux カーネルの中に組み込まれた Linux セキュリティモジュールです。SELinux は読み込み可能なポリシールールにより動作します。プロセスがファイルを開くときのように、セキュリティ関連のアクセスが実行されるとき、操作がカーネルにおいて SELinux により割り込まれます。SELinux ポリシールールが操作を許可すると、継続されます。そうでなければ、操作がブロックされ、プロセスがエラーを受け取ります。

アクセスの許可や拒否のような SELinux の判断はキャッシュされます。このキャッシュはアクセスベクターキャッシュ (AVC: Access Vector Cache) として知られています。判断をキャッシュすることにより、SELinux ポリシールールが確認される必要性を減らし、パフォーマンスを改善します。まず DAC によりアクセスを拒否されると、SELinux ポリシールールが効果を持たないことを覚えておいてください。

9.1.4. 他のオペレーティングシステムにおける SELinux

オペレーティングシステムにおける SELinux の実行に関する詳細は以下の情報を参照ください:

- Gentoo SELinux Handbook¥n: *http://www.gentoo.org/proj/en/hardened/selinux/selinux-handbook.xml*.

- Debian: *http://wiki.debian.org/SELinux*.

- Ubuntu: *https://wiki.ubuntu.com/SELinux* および *https://help.ubuntu.com/community/SELinux*.

- Red Hat Enterprise Linux: *Red Hat Enterprise Linux Deployment Guide*[16] および *Red Hat Enterprise Linux 4 SELinux Guide*[17].

- Fedora: *http://fedoraproject.org/wiki/SELinux* および *Fedora Core 5 SELinux FAQ*[18]。

9.2. SELinux コンテキスト

プロセスとファイルは、SELinux ユーザー、役割、タイプ、およびオプションとしてレベルなどの追加情報を含む、SELinux コンテキストでラベル付けされています。SELinux を実行しているとき、この情報のすべてがアク

[9] http://www.linuxworld.com
[10] http://www.linuxworld.com/news/2008/022408-selinux.html?page=1
[11] Marti, Don. "A seatbelt for server software: SELinux blocks real-world exploits". 2008年2月24日発行。2009年8月27日アクセス確認: *http://www.linuxworld.com/news/2008/022408-selinux.html?page=1*.
[12] http://www.openpegasus.org/
[13] http://james-morris.livejournal.com/25421.html
[14] http://www.tresys.com/
[15] http://www.tresys.com/innovation.php
[16] http://www.redhat.com/docs/en-US/Red_Hat_Enterprise_Linux/5.2/html/Deployment_Guide/selg-overview.html
[17] http://www.redhat.com/docs/manuals/enterprise/RHEL-4-Manual/selinux-guide/
[18] http://docs.fedoraproject.org/selinux-faq-fc5/

セス制御の判断のために使用されます。Fedora では、SELinux が役割ベースアクセス制御 (RBAC: Role-Based Access Control)、タイプエンフォースメント (TE: Type Enforcement®)、およびオプションとしてマルチレベルセキュリティ (MLS: Multi-Level Security) を提供します。

以下は SELinux コンテキストを表示する例です。SELinux コンテキストは、SELinux を実行する Linux オペレーティングシステムにおいて、プロセス、Linux ユーザー、およびファイルにおいて使用されます。ファイルとディレクトリの SELinux コンテキストを表示するには `ls -Z` コマンドを使用します:

```
$ ls -Z file1
-rw-rw-r--. user1 group1 unconfined_u:object_r:user_home_t:s0 file1
```

SELinux コンテキストは *SELinux user:role:type:level* 構文に従います:

SELinux ####

SELinux ユーザー識別子は、特定の役割のセット、または特定の MLS 範囲に対して認可されるポリシーに知られる識別子です。それぞれの Linux ユーザーは SELinux ポリシー経由で SELinux ユーザーに対応づけられます。これにより、Linux ユーザーが SELinux ユーザーにおける制限を継承できます。対応づけられた SELinux ユーザー識別子は、そのユーザーが入れる役割とレベルを定義するために、そのセッションにおけるプロセスのための SELinux コンテキストで使用されます。SELinux と Linux ユーザーアカウントの対応づけの一覧を表示するには、Linux root ユーザーとして `semanage login -l` コマンドを実行します:

```
# /usr/sbin/semanage login -l

Login Name                SELinux User              MLS/MCS Range

__default__               unconfined_u              s0-s0:c0.c1023
root                      unconfined_u              s0-s0:c0.c1023
system_u                  system_u                  s0-s0:c0.c1023
```

出力内容はシステムにより異なるかもしれません。`Login Name` 項目は Linux ユーザーの一覧です。また、`SELinux User` 項目は Linux ユーザーが対応づけられている SELinux ユーザーです。プロセスに対しては、SELinux ユーザーがアクセス可能な役割とレベルを制限します。最後の項目 `MLS/MCS Range` はマルチレベルセキュリティ (Multi-Level Security:MLS) とマルチカテゴリーセキュリティ (MCS:Multi-Category Security) により使用されるレベルです。レベルについて後から簡単に議論します。

###

SELinux の一部分は役割ベースアクセス制御 (RBAC: Role-Based Access Control) セキュリティモデルです。役割は RBAC の属性です。SELinux ユーザーが役割に対して認可され、役割がドメインに対して認可されます。役割がドメインと SELinux ユーザーの間の仲介者として処理します。入ることができる役割が、入ることができるドメインを決定します - 最終的に、これがアクセスできるオブジェクトタイプを制御します。これにより、権限上昇攻撃の脆弱性を減らす手助けになります。

###

タイプはタイプエンフォースメントの属性です。タイプがプロセス向けのドメインとファイル向けのタイプを定義します。SELinux ポリシールールが、タイプがお互いにどのようにアクセスするか、ドメインがタイプにアクセスできるかどうか、ドメインが他のドメインにアクセスできるかどうかを定義します。特定の SELinux ポリシールールがそれを許可する場合のみ、アクセスが許可されます。

###

レベルは MLS およびマルチカテゴリセキュリティ（MCS）の属性です。MLS 範囲はレベルの組です。レベルが異なれば *####-####* として、レベルが同じならば *####* として書かれます (`s0-s0` は `s0` と同じです)。各レベルは機密度とカテゴリ（オプション）の組です。カテゴリがあれば、レベルが *###:####-#* として書かれます。カテゴリがなければ、*###* として書かれます。

カテゴリの組が連続した数字であれば、省略できます。たとえば、c0.c3 は c0,c1,c2,c3 と同じです。/etc/selinux/targeted/setrans.conf ファイルにより、レベル (s0:c0) を人間の読みやすい形式 (例 CompanyConfidential) に対応づけます。テキストエディターで setrans.conf を編集しないでください: 変更するには semanage を使用します。詳細は semanage(8) マニュアルページを参照してください。Fedora では、ターゲットポリシーが MCS を強制します。そして、一つの機密度 s0 のみが MCS にあります。Fedora の MCS は 1024 個のカテゴリをサポートします: c0 から c1023 まで。s0-s0:c0.c1023 は気密度 s0 で、すべてのカテゴリに対して認可されます。

MLS が *Bell-La Padula Mandatory Access Model*[19] を強制し、Labeled Security Protection Profile (LSPP) 環境で使用されます。MLS 制限を使用するために、*selinux-policy-mls* パッケージをインストールし、MLS を標準の SELinux ポリシーになるよう /etc/selinux/config ファイルで設定します。Fedora に同梱されている MLS ポリシーは、評価された設定の一部分ではない、多くのプログラムドメインを省略します。そのため、デスクトップ環境における MLS は意味がありません (X Window System をサポートしません)。しかしながら、*upstream SELinux Reference Policy*[20] にある MLS ポリシーは、すべてのプログラムドメインを含むよう構築されています。

9.2.1. ドメイン遷移

あるドメインにあるプロセスが、新しいドメイン向けの entrypoint タイプを持つアプリケーションを実行することにより、他のドメインに遷移します。entrypoint パーミッションが SELinux ポリシーにおいて使用され、ドメインに入るために使用できるアプリケーションを制御します。以下の例がドメイン遷移を説明します:

1. ユーザーがパスワードを変更したいです。これをするために、passwd アプリケーションを実行します。/usr/bin/passwd の実行バイナリが passwd_exec_t タイプのラベルを付けられています:

```
$ ls -Z /usr/bin/passwd
-rwsr-xr-x  root root system_u:object_r:passwd_exec_t:s0 /usr/bin/passwd
```

   passwd アプリケーションが /etc/shadow にアクセスします。これは shadow_t タイプのラベルが付けられています:

```
$ ls -Z /etc/shadow
----------. root root system_u:object_r:shadow_t:s0    /etc/shadow
```

2. SELinux ポリシールールにより、passwd_t ドメインで実行中のプロセスが shadow_t タイプのラベルを持つファイルを読み書きできるよう宣言されます。shadow_t タイプはパスワードを変更するために必要となるファイルのみに適用されます。これに /etc/gshadow, /etc/shadow, およびそのバックアップファイルが含まれます。

3. SELinux ポリシールールにより、passwd_t ドメインが passwd_exec_t タイプに対する entrypoint パーミッションを持つことを宣言されます。

4. ユーザーが /usr/bin/passwd アプリケーションを実行するとき、ユーザーのシェルプログラムが passwd_t ドメインに遷移します。SELinux を用いると、標準のアクションが拒否され、passwd_t ドメインにおいて実行中のアプリケーションが shadow_t タイプのラベルを持つファイルにアクセスできるルールが存在するので、passwd アプリケーションが /etc/shadow にアクセスすることを許可され、ユーザーのパスワードを更新します。

この例は網羅的でなく、ドメイン遷移を説明するための基本的な例として使用されます。passwd_t ドメインで実行中のサブジェクトが shadow_t タイプのラベルを持つオブジェクトにアクセスできるという、実際のルールがあ

[19] http://en.wikipedia.org/wiki/Bell-LaPadula_model
[20] http://oss.tresys.com/projects/refpolicy

るにもかかわらず、サブジェクトが新しいドメインに遷移できる前に、他の SELinux ポリシールールが適合する必要があります。この例では、タイプエンフォースメントが保証します:

- passwd_exec_t タイプのラベルを持つアプリケーションを実行することにより、passwd_t ドメインのみが入れます。lib_t タイプのような認可された共有ライブラリからのみ実行でき、他のすべてのアプリケーションを実行できません。
- passwd_t のような認可されたドメインのみが、shadow_t タイプのラベルを持つファイルにアクセスできます。他のプロセスがスーパーユーザー権限で実行されていても、passwd_t ドメインで実行されていないので、それらのプロセスが shadow_t タイプのラベルを持つファイルにアクセスできません。
- 認可されたドメインのみが passwd_t ドメインに遷移できます。たとえば、sendmail_t ドメインで実行している sendmail プロセスが、passwd を実行する正当な理由がありません。そのため、passwd_t ドメインに移行できません。
- passwd_t ドメインで実行されているプロセスは、etc_t や shadow_t タイプのラベルを持つファイルのように、認可されたタイプのみに読み書きできます。これにより、passwd アプリケーションが任意のファイルをだまして読み書きすることを防ぎます。

9.2.2. プロセスの SELinux コンテキスト

プロセスの SELinux コンテキストを表示するには、ps -eZ コマンドを使用します。たとえば:

1. アプリケーション → システムツール → 端末 などとし、端末を開きます。
2. /usr/bin/passwd コマンドを実行します。新しいパスワードを入力しません。
3. 新しいタブ、または他の端末を開きます。そして ps -eZ | grep passwd コマンドを実行します。出力は以下のようなものです:

```
unconfined_u:unconfined_r:passwd_t:s0-s0:c0.c1023 13212 pts/1 00:00:00 passwd
```

4. 最初のタブ・端末において、passwd アプリケーションを取り消すために Ctrl+C を入力します。

この例では、/usr/bin/passwd アプリケーション (passwd_exec_t タイプのラベルを持つ) が実行されるとき、ユーザーのシェルプロセスが passwd_t ドメインに遷移します。注: タイプが、プロセスに対してドメインを、ファイルに対してタイプを定義します。

実行中のプロセスの SELinux コンテキストを表示するには、ps -eZ コマンドを使用します。以下が出力の一例です。お使いのシステムと異なるかもしれません:

```
system_u:system_r:dhcpc_t:s0     1869 ?        00:00:00 dhclient
system_u:system_r:sshd_t:s0-s0:c0.c1023 1882 ? 00:00:00 sshd
system_u:system_r:gpm_t:s0       1964 ?        00:00:00 gpm
system_u:system_r:crond_t:s0-s0:c0.c1023 1973 ? 00:00:00 crond
system_u:system_r:kerneloops_t:s0 1983 ?       00:00:05 kerneloops
system_u:system_r:crond_t:s0-s0:c0.c1023 1991 ? 00:00:00 atd
```

system_r ロールは、デーモンのようなシステムプロセスのために使用されます。タイプエンフォースメントが各ドメインを分離します。

9.2.3. ユーザー向け SELinux コンテキスト

あなたの Linux ユーザーに関連づけられている SELinux コンテキストを表示するには、id -Z コマンドを使用します:

```
unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
```

Fedora では標準で、Linux ユーザーが制限されずに実行されます。この SELinux コンテキストが、Linux ユーザーが SELinux unconfined_u ユーザーに対応付けられることを示します。これは unconfined_r ロールとして実行され、unconfined_t ドメインにおいて実行されます。s0-s0 は MLS 範囲です。この場合、単に s0 と同じです。.ユーザーがアクセス権を持つカテゴリは c0.c1023 により定義されます。これは、すべてのカテゴリ (c0 から c1023) です。

9.3. Targeted Policy

Targeted policy is the default SELinux policy used in Fedora. When using targeted policy, processes that are targeted run in a confined domain, and processes that are not targeted run in an unconfined domain. For example, by default, logged in users run in the unconfined_t domain, and system processes started by init run in the initrc_t domain - both of these domains are unconfined.

Unconfined domains (as well as confined domains) are subject to executable and writeable memory checks. By default, subjects running in an unconfined domain can not allocate writeable memory and execute it. This reduces vulnerability to *buffer overflow attacks*[21]. These memory checks are disabled by setting Booleans, which allow the SELinux policy to be modified at runtime. Boolean configuration is discussed later.

9.3.1. Confined Processes

Almost every service that listens on a network is confined in Fedora. Also, most processes that run as the Linux root user and perform tasks for users, such as the passwd application, are confined. When a process is confined, it runs in its own domain, such as the httpd process running in the httpd_t domain. If a confined process is compromised by an attacker, depending on SELinux policy configuration, an attacker's access to resources and the possible damage they can do is limited.

The following example demonstrates how SELinux prevents the Apache HTTP Server (httpd) from reading files that are not correctly labeled, such as files intended for use by Samba. This is an example, and should not be used in production. It assumes that the *httpd*, *wget*, *setroubleshoot-server*, *dbus* and *audit* packages are installed, that the SELinux targeted policy is used, and that SELinux is running in enforcing mode:

1. SELinux が有効化され、エンフォースモードで実行され、ターゲットポリシーが使用されていることを確認するために、sestatus コマンドを実行します:

   ```
   $ /usr/sbin/sestatus
   SELinux status:                 enabled
   SELinuxfs mount:                /selinux
   Current mode:                   enforcing
   Mode from config file:          enforcing
   Policy version:                 24
   Policy from config file:        targeted
   ```

   SELinux が有効化されているとき、SELinux status: enabled が返されます。SELinux がエンフォースモードで実行されているとき、Current mode: enforcing が返されます。SELinux ターゲットポリシーが使用されているとき、Policy from config file: targeted が返されます。

[21] http://en.wikipedia.org/wiki/Buffer_overflow

2. As the Linux root user, run the touch /var/www/html/testfile command to create a file.

3. SELinux コンテキストを表示するには ls -Z /var/www/html/testfile コマンドを実行します:

```
-rw-r--r--  root root unconfined_u:object_r:httpd_sys_content_t:s0 /var/www/html/testfile
```

   By default, Linux users run unconfined in Fedora, which is why the testfile file is labeled with the SELinux unconfined_u user. RBAC is used for processes, not files. Roles do not have a meaning for files - the object_r role is a generic role used for files (on persistent storage and network file systems). Under the /proc/ directory, files related to processes may use the system_r role.[22] The httpd_sys_content_t type allows the httpd process to access this file.

4. As the Linux root user, run the service httpd start command to start the httpd process. The output is as follows if httpd starts successfully:

```
# /sbin/service httpd start
Starting httpd:                                            [  OK  ]
```

5. Linux ユーザーが書き込みアクセス権を持つディレクトリに変更し、wget http://localhost/testfile コマンドを実行します。標準の設定から変更されていなければ、このコマンドが成功します:

```
$ wget http://localhost/testfile

--2010-05-11 13:19:07--  http://localhost/testfile
Resolving localhost... ::1, 127.0.0.1
Connecting to localhost|::1|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 0 [text/plain]
Saving to: “testfile”

    [ <=>            ] 0           --.-K/s   in 0s

2010-05-11 13:19:07 (0.00 B/s) - “testfile” saved [0/0]
```

6. The chcon command relabels files; however, such label changes do not survive when the file system is relabeled. For permanent changes that survive a file system relabel, use the semanage command, which is discussed later. As the Linux root user, run the following command to change the type to a type used by Samba:

   chcon -t samba_share_t /var/www/html/testfile

   変更内容を表示するには ls -Z /var/www/html/testfile を実行します:

```
-rw-r--r--  root root unconfined_u:object_r:samba_share_t:s0 /var/www/html/testfile
```

7. 注: 現在の DAC パーミッションにより、httpd プロセスが testfile にアクセスできます。Linux ユーザーが書き込みアクセス権を持つディレクトリに変更し、wget http://localhost/testfile コマンドを実行します。標準の設定から変更されていなければ、このコマンドが成功します:

```
$ wget http://localhost/testfile
```

[22] When using other policies, such as MLS, other roles may be used, for example, secadm_r.

```
--2010-05-11 13:23:49--  http://localhost/testfile
Resolving localhost... ::1, 127.0.0.1
Connecting to localhost|::1|:80... connected.
HTTP request sent, awaiting response... 403 Forbidden
2010-05-11 13:23:49 ERROR 403: Forbidden.
```

8. As the Linux root user, run the `rm -i /var/www/html/testfile` command to remove `testfile`.

9. If you do not require `httpd` to be running, as the Linux root user, run the `service httpd stop` command to stop `httpd`:

```
# /sbin/service httpd stop
Stopping httpd:                                            [  OK  ]
```

This example demonstrates the additional security added by SELinux. Although DAC rules allowed the `httpd` process access to `testfile` in step 7, because the file was labeled with a type that the `httpd` process does not have access to, SELinux denied access. After step 7, an error similar to the following is logged to `/var/log/messages`:

```
May 11 13:23:51 localhost setroubleshoot: SELinux is preventing /usr/sbin/httpd "getattr" access to /var/www/
html/testfile. For complete SELinux messages. run sealert -l ca2ab0df-fcb9-46d1-8283-037450d1efcc
```

以前のログファイルが `/var/log/messages.`***YYYYMMDD*** 形式を使用するかもしれません。syslog-ng を実行しているとき、以前のログファイルが `/var/log/messages.`***X*** 形式を使用するかもしれません。`setroubleshootd` および `auditd` プロセスが実行中ならば、以下のようなエラーが `/var/log/audit/audit.log` に記録されます:

```
type=AVC msg=audit(1220706212.937:70): avc:  denied  { getattr } for  pid=1904 comm="httpd"
 path="/var/www/html/testfile" dev=sda5 ino=247576 scontext=unconfined_u:system_r:httpd_t:s0
 tcontext=unconfined_u:object_r:samba_share_t:s0  tclass=file

type=SYSCALL msg=audit(1220706212.937:70): arch=40000003 syscall=196 success=no exit=-13 a0=b9e21da0
 a1=bf9581dc a2=555ff4 a3=2008171 items=0 ppid=1902 pid=1904 auid=500 uid=48 gid=48 euid=48
 suid=48 fsuid=48 egid=48 sgid=48 fsgid=48 tty=(none) ses=1 comm="httpd" exe="/usr/sbin/httpd"
 subj=unconfined_u:system_r:httpd_t:s0 key=(null)
```

また、以下のようなエラーが `/var/log/httpd/error_log` に記録されます:

```
[Tue May 11 13:23:49 2010] [error] [client ::1] (13)Permission denied: access to /testfile denied
```

9.3.2. Unconfined Processes

Unconfined processes run in unconfined domains, for example, init programs run in the unconfined `initrc_t` domain, unconfined kernel processes run in the `kernel_t` domain, and unconfined Linux users run in the `unconfined_t` domain. For unconfined processes, SELinux policy rules are applied, but policy rules exist that allow processes running in unconfined domains almost all access. Processes running in unconfined domains fall back to using DAC rules exclusively. If an unconfined process is compromised, SELinux does not prevent an attacker from gaining access to system resources and data, but of course, DAC rules are still used. SELinux is a security enhancement on top of DAC rules - it does not replace them.

The following example demonstrates how the Apache HTTP Server (`httpd`) can access data intended for use by Samba, when running unconfined. Note: in Fedora, the `httpd` process runs in the confined `httpd_t` domain by default. This is an example, and should not be used in

production. It assumes that the *httpd*, *wget*, *setroubleshoot-server*, *dbus* and *audit* packages are installed, that the SELinux targeted policy is used, and that SELinux is running in enforcing mode:

1. SELinux が有効化され、エンフォースモードで実行され、ターゲットポリシーが使用されていることを確認するために、sestatus コマンドを実行します:

```
$ /usr/sbin/sestatus
SELinux status:                 enabled
SELinuxfs mount:                /selinux
Current mode:                   enforcing
Mode from config file:          enforcing
Policy version:                 24
Policy from config file:        targeted
```

   SELinux が有効化されているとき、SELinux status: enabled が返されます。SELinux がエンフォースモードで実行されているとき、Current mode: enforcing が返されます。SELinux ターゲットポリシーが使用されているとき、Policy from config file: targeted が返されます。

2. As the Linux root user, run the touch /var/www/html/test2file command to create a file.

3. SELinux コンテキストを表示するには ls -Z /var/www/html/test2file コマンドを実行します:

```
-rw-r--r--  root root unconfined_u:object_r:httpd_sys_content_t:s0 /var/www/html/test2file
```

   By default, Linux users run unconfined in Fedora, which is why the test2file file is labeled with the SELinux unconfined_u user. RBAC is used for processes, not files. Roles do not have a meaning for files - the object_r role is a generic role used for files (on persistent storage and network file systems). Under the /proc/ directory, files related to processes may use the system_r role.[23] The httpd_sys_content_t type allows the httpd process to access this file.

4. The chcon command relabels files; however, such label changes do not survive when the file system is relabeled. For permanent changes that survive a file system relabel, use the semanage command, which is discussed later. As the Linux root user, run the following command to change the type to a type used by Samba:

   chcon -t samba_share_t /var/www/html/test2file

   変更を表示するには ls -Z /var/www/html/test2file コマンドを実行します:

```
-rw-r--r--  root root unconfined_u:object_r:samba_share_t:s0 /var/www/html/test2file
```

5. httpd プロセスが実行中であることを確認するには、service httpd status コマンドを実行します:

```
$ /sbin/service httpd status
httpd is stopped
```

   If the output differs, run the service httpd stop command as the Linux root user to stop the httpd process:

```
# /sbin/service httpd stop
Stopping httpd:                                            [  OK  ]
```

[23] When using other policies, such as MLS, other roles may also be used, for example, secadm_r.

6. To make the httpd process run unconfined, run the following command as the Linux root user to change the type of /usr/sbin/httpd, to a type that does not transition to a confined domain:

   ```
   chcon -t unconfined_exec_t /usr/sbin/httpd
   ```

7. /usr/sbin/httpd が unconfined_exec_t タイプのラベルがつけれらていることを確認するには ls -Z /usr/sbin/httpd コマンドを実行します:

   ```
   -rwxr-xr-x  root root system_u:object_r:unconfined_exec_t /usr/sbin/httpd
   ```

8. As the Linux root user, run the service httpd start command to start the httpd process. The output is as follows if httpd starts successfully:

   ```
   # /sbin/service httpd start
   Starting httpd:                                            [  OK  ]
   ```

9. Run the ps -eZ | grep httpd command to view the httpd running in the unconfined_t domain:

   ```
   $ ps -eZ | grep httpd
   unconfined_u:system_r:unconfined_t 7721 ?      00:00:00 httpd
   unconfined_u:system_r:unconfined_t 7723 ?      00:00:00 httpd
   unconfined_u:system_r:unconfined_t 7724 ?      00:00:00 httpd
   unconfined_u:system_r:unconfined_t 7725 ?      00:00:00 httpd
   unconfined_u:system_r:unconfined_t 7726 ?      00:00:00 httpd
   unconfined_u:system_r:unconfined_t 7727 ?      00:00:00 httpd
   unconfined_u:system_r:unconfined_t 7728 ?      00:00:00 httpd
   unconfined_u:system_r:unconfined_t 7729 ?      00:00:00 httpd
   unconfined_u:system_r:unconfined_t 7730 ?      00:00:00 httpd
   ```

10. Linux ユーザーが書き込みアクセス権を持つディレクトリに変更し、wget http://localhost/testfile コマンドを実行します。標準の設定から変更されていなければ、このコマンドが成功します:

    ```
    --2009-05-07 01:41:10--  http://localhost/test2file
    Resolving localhost... 127.0.0.1
    Connecting to localhost|127.0.0.1|:80... connected.
    HTTP request sent, awaiting response... 200 OK
    Length: 0 [text/plain]
    Saving to: `test2file.1'

    [ <=>                                ]--.-K/s   in 0s

    2009-05-07 01:41:10 (0.00 B/s) - `test2file.1' saved [0/0]
    ```

    httpd プロセスが samba_share_t タイプのラベルを持つファイルにアクセス権を持たないにもかかわらず、httpd が制限されない unconfined_t ドメインで実行されます。そして、DAC ルールを使用するように戻ります。それにより wget コマンドなどが成功します。httpd が制限された httpd_t ドメインで実行されていると、wget コマンドが失敗します。

11. The restorecon command restores the default SELinux context for files. As the Linux root user, run the restorecon -v /usr/sbin/httpd command to restore the default SELinux context for /usr/sbin/httpd:

    ```
    # /sbin/restorecon -v /usr/sbin/httpd
    ```

```
restorecon reset /usr/sbin/httpd context system_u:object_r:unconfined_notrans_exec_t:s0-
>system_u:object_r:httpd_exec_t:s0
```

/usr/sbin/httpd が httpd_exec_t タイプのラベルを持つことを確認するために ls -Z /usr/sbin/httpd コマンドを実行します:

```
$ ls -Z /usr/sbin/httpd
-rwxr-xr-x  root root system_u:object_r:httpd_exec_t   /usr/sbin/httpd
```

12. As the Linux root user, run the /sbin/service httpd restart command to restart httpd. After restarting, run the ps -eZ | grep httpd to confirm that httpd is running in the confined httpd_t domain:

```
# /sbin/service httpd restart
Stopping httpd:                                            [  OK  ]
Starting httpd:                                            [  OK  ]
# ps -eZ | grep httpd
unconfined_u:system_r:httpd_t    8880 ?        00:00:00 httpd
unconfined_u:system_r:httpd_t    8882 ?        00:00:00 httpd
unconfined_u:system_r:httpd_t    8883 ?        00:00:00 httpd
unconfined_u:system_r:httpd_t    8884 ?        00:00:00 httpd
unconfined_u:system_r:httpd_t    8885 ?        00:00:00 httpd
unconfined_u:system_r:httpd_t    8886 ?        00:00:00 httpd
unconfined_u:system_r:httpd_t    8887 ?        00:00:00 httpd
unconfined_u:system_r:httpd_t    8888 ?        00:00:00 httpd
unconfined_u:system_r:httpd_t    8889 ?        00:00:00 httpd
```

13. As the Linux root user, run the rm -i /var/www/html/test2file command to remove test2file.

14. If you do not require httpd to be running, as the Linux root user, run the service httpd stop command to stop httpd:

```
# /sbin/service httpd stop
Stopping httpd:                                            [  OK  ]
```

The examples in these sections demonstrate how data can be protected from a compromised confined-process (protected by SELinux), as well as how data is more accessible to an attacker from a compromised unconfined-process (not protected by SELinux).

9.3.3. Confined and Unconfined Users

Each Linux user is mapped to an SELinux user via SELinux policy. This allows Linux users to inherit the restrictions on SELinux users. This Linux user mapping is seen by running the semanage login -l command as the Linux root user:

```
# /usr/sbin/semanage login -l

Login Name                SELinux User              MLS/MCS Range

__default__               unconfined_u              s0-s0:c0.c1023
root                      unconfined_u              s0-s0:c0.c1023
system_u                  system_u                  s0-s0:c0.c1023
```

In Fedora 19, Linux users are mapped to the SELinux __default__ login by default (which is mapped to the SELinux unconfined_u user). The following defines the default-mapping:

```
__default__             unconfined_u            s0-s0:c0.c1023
```

The following example demonstrates adding a new Linux user, and that Linux user being mapped to the SELinux unconfined_u user. It assumes that the Linux root user is running unconfined, as it does by default in Fedora 19:

1. As the Linux root user, run the /usr/sbin/useradd newuser command to create a new Linux user named newuser.

2. As the Linux root user, run the passwd newuser command to assign a password to the Linux newuser user:

```
# passwd newuser
Changing password for user newuser.
New UNIX password: Enter a password
Retype new UNIX password: Enter the same password again
passwd: all authentication tokens updated successfully.
```

3. Log out of your current session, and log in as the Linux newuser user. When you log in, pam_selinux maps the Linux user to an SELinux user (in this case, unconfined_u), and sets up the resulting SELinux context. The Linux user's shell is then launched with this context. Run the id -Z command to view the context of a Linux user:

```
[newuser@localhost ~]$ id -Z
unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
```

4. Log out of the Linux newuser's session, and log in with your account. If you do not want the Linux newuser user, run the /usr/sbin/userdel -r newuser command as the Linux root user to remove it, along with the Linux newuser's home directory.

Confined and unconfined Linux users are subject to executable and writeable memory checks, and are also restricted by MCS (and MLS, if the MLS policy is used). If unconfined Linux users execute an application that SELinux policy defines can transition from the unconfined_t domain to its own confined domain, unconfined Linux users are still subject to the restrictions of that confined domain. The security benefit of this is that, even though a Linux user is running unconfined, the application remains confined, and therefore, the exploitation of a flaw in the application can be limited by policy. Note: this does not protect the system from the user. Instead, the user and the system are being protected from possible damage caused by a flaw in the application.

The following confined SELinux users are available in Fedora 19:

表9.1 SELinux User Capabilities

User	Domain	X Window System	su and sudo	Execute in home directory and /tmp/	Networking
guest_u	guest_t	no	no	optional	no
xguest_u	xguest_t	yes	no	optional	only Firefox
user_u	user_t	yes	no	optional	yes
staff_u	staff_t	yes	only sudo	optional	yes

- Linux users in the guest_t, xguest_t, and user_t domains can only run set user ID (setuid) applications if SELinux policy permits it (such as passwd). They can not run the su and /usr/bin/

sudo setuid applications, and therefore, can not use these applications to become the Linux root user.

- Linux users in the guest_t domain have no network access, and can only log in via a terminal (including ssh; they can log in via ssh, but can not use ssh to connect to another system).
- The only network access Linux users in the xguest_t domain have is Firefox connecting to web pages.
- Linux users in the xguest_t, user_t and staff_t domains can log in via the X Window System and a terminal.
- By default, Linux users in the staff_t domain do not have permissions to execute applications with /usr/bin/sudo. These permissions must be configured by an administrator.

By default, Linux users in the guest_t and xguest_t domains can not execute applications in their home directories or /tmp/, preventing them from executing applications (which inherit users' permissions) in directories they have write access to. This helps prevent flawed or malicious applications from modifying files users' own.

By default, Linux users in the user_t and staff_t domains can execute applications in their home directories and /tmp/. Refer to ########################## for information about allowing and preventing users from executing applications in their home directories and /tmp/.

9.4. SELinux での動作

以下のセクションは、Fedora におけるおもな SELinux パッケージ、パッケージのインストールおよび更新、使用されるログファイル、おもな SELinux 設定ファイル、SELinux の有効化と無効化、SELinux のモード、ブーリアンの設定、一時的および永続的なファイルおよびディレクトリのラベルの変更、mount コマンドを用いたファイルシステムラベルの上書き、NFS ファイルシステムのマウント、ファイルとディレクトリをコピーおよびアーカイブするときに SELinux コンテキストを保存する方法に関する概略について説明します。

9.4.1. SELinux パッケージ

Fedora では、SELinux をインストール中に手動で除外しない限り、完全インストールで標準でインストールされます。テキストモードで最小インストールを実行していると、*policycoreutils-python* パッケージが標準でインストールされません。また標準で、SELinux ターゲットポリシーが使用され、SELinux をエンフォースモードで実行します。以下はおもな SELinux パッケージの概説です:

policycoreutils-python: SELinux を操作および管理するために semanage, audit2allow, audit2why および chcat のようなユーティリティを提供します。

policycoreutils: SELinux を操作および管理するために restorecon, secon, setfiles, semodule, load_policy, および setsebool のようなユーティリティを提供します。

policycoreutils-gui: SELinux を管理するためにグラフィカルツール system-config-selinux を提供します。

selinux-policy: SELinux リファレンスポリシーを提供します。SELinux リファレンスポリシーは完全な SELinux ポリシーです。SELinux ターゲットポリシーのような、他のポリシーの基礎として使用されます。さらなる情報は Tresys Technology *SELinux Reference Policy*[24] のページを参照してください。*selinux-policy-devel* パッケージが、/usr/share/selinux/devel/policygentool や /usr/share/selinux/devel/policyhelp のような開発ツール、ポリシーファイル例を提供します。このパッケージは *selinux-policy* パッケージにマージされました。

[24] http://oss.tresys.com/projects/refpolicy

selinux-policy-policy: SELinux ポリシーを提供します。ターゲットポリシー向けは *selinux-policy-targeted* をインストールします。MLS 向けは *selinux-policy-mls* をインストールします。

setroubleshoot-server: アクセスが SELinux により拒否されたときに生成された拒否メッセージを、`sealert` (このパッケージにより提供されます) を用いて表示される詳細な説明に変換します。

setools, *setools-gui*, および *setools-console*: これらのパッケージは *Tresys Technology SETools distribution*[25] を提供します。これらは、ポリシーの分析や問い合わせ、監査ログのモニタリングおよびレポート、ファイルコンテキストの管理[26]のために使用される数多くのツールおよびライブラリです。*setools* パッケージは SETools 向けのメタパッケージです。*setools-gui* パッケージが `apol`, `seaudit`, および `sediffx` ツールを提供します。*setools-console* パッケージが `seaudit-report`, `sechecker`, `sediff`, `seinfo`, `sesearch`, `findcon`, `replcon`, および `indexcon` コマンドラインツールを提供します。これらのツールに関する詳細は *Tresys Technology SETools*[27] ページを参照してください。

libselinux-utils: `avcstat`, `getenforce`, `getsebool`, `matchpathcon`, `selinuxconlist`, `selinuxdefcon`, `selinuxenabled`, `setenforce`, `togglesebool` ツールを提供します。

mcstrans: `s0-s0:c0.c1023` のようなレベルを読みやすくするために `SystemLow-SystemHigh` のような形式に変換します。このパッケージは標準でインストールされません。

Fedora においてパッケージをインストールするには、Linux root ユーザーとして `yum install` ***package-name*** コマンドを実行します。たとえば、*mcstrans* パッケージをインストールするには、`yum install mcstrans` コマンドを実行します。Fedora においてすべてのインストール済みパッケージを更新するには、`yum update` コマンドを実行します。

パッケージを管理するために `yum` を使用する方法の詳細は *Managing Software with yum*[28][29] を参照してください。

注記

前のバージョンの Fedora において、`audit2allow -M` を用いてローカルポリシーモジュールを作成するときに、*selinux-policy-devel* パッケージが必要となります。

9.4.2. 使用するログファイル

Fedora 19 では、パッケージが標準のパッケージ選択から削除されていなければ、*dbus*, *setroubleshoot-server* および *audit* パッケージがインストールされます。

以下のような SELinux 拒否メッセージが標準で `/var/log/audit/audit.log` に書き込まれます:

```
type=AVC msg=audit(1223024155.684:49): avc:  denied  { getattr } for  pid=2000 comm="httpd"
 path="/var/www/html/file1" dev=dm-0 ino=399185 scontext=unconfined_u:system_r:httpd_t:s0
 tcontext=system_u:object_r:samba_share_t:s0 tclass=file
```

[25] http://oss.tresys.com/projects/setools
[26] Brindle, Joshua. "Re: blurb for fedora setools packages" Murray McAllister 宛ての電子メール。2008年11月1日。このバージョンにおけるすべての編集や変更は Murray McAllister により実行されました。
[27] http://oss.tresys.com/projects/setools
[28] http://docs.fedoraproject.org/yum/en/
[29] Managing Software with yum、Stuart Ellis 著、Paul W. Frields, Rodrigo Menezes, Hugo Cisneiros 編。

また、setroubleshootd が動作していると、/var/log/audit/audit.log からの拒否メッセージが読みやすい形式に変換され、/var/log/messages に送られます:

```
May  7 18:55:56 localhost setroubleshoot: SELinux is preventing httpd (httpd_t) "getattr" to /var/www/html/
file1 (samba_share_t). For complete SELinux messages. run sealert -l de7e30d6-5488-466d-a606-92c9f40d316d
```

Fedora 19 では、もはや setroubleshootd がサービスとして継続的に実行されることがありません。しかしながら、AVC メッセージを分析するためにまだ使用されます。新しい 2 つのプログラムが、必要なときに setroubleshoot を開始するの方法として動作します: sedispatch および seapplet。sedispatch が監査サブシステムの一部として動作します。また、AVC 拒否が発生したときにメッセージを dbus 経由で送信します。これは、setroubleshootd が動作しているならば、そのままそこに送られます。または、setroubleshootd が動作していなければ、それを開始します。seapplet は、システムのツールバーで動作するツールです。setroubleshootd で dbus メッセージを待ちます。そして、ユーザーが拒否をレビューするために、通知バブルが起動します。

拒否メッセージが別の場所にも送信されます。どのデーモンを実行しているかによります:

デーモン	ログの位置
auditd on	/var/log/audit/audit.log
auditd off; rsyslogd on	/var/log/messages
rsyslogd on と auditd on	/var/log/audit/audit.log。読みやすい形式の拒否メッセージが /var/log/messages にも送信されます

デーモンの自動起動

auditd, rsyslogd, および setroubleshootd デーモンをブート時に自動的に開始するよう設定するには、Linux root ユーザーとして以下のコマンドを実行します:

```
/sbin/chkconfig --levels 2345 auditd on
```

```
/sbin/chkconfig --levels 2345 rsyslog on
```

これらのサービスが実行中であることを確認するには、service ***service-name*** status コマンドを使用します。例

```
$ /sbin/service auditd status
auditd (pid  1318) is running...
```

上のサービスが実行されていなければ (***service-name*** is stopped)、それらを開始するために Linux root ユーザーとして service ***service-name*** start コマンドを使用します。例:

```
# /sbin/service auditd start
Starting auditd:                                [  OK  ]
```

9.4.3. メインの設定ファイル

/etc/selinux/config ファイルがメインの SELinux 設定ファイルです。使用する SELinux モードと SELinux ポリシーを制御します:

```
# このファイルがシステムにおける SELinux の状態を制御します。
# SELINUX= これら 3 つの値のどれかを使用できます:
```

```
#       enforcing - SELinux セキュリティポリシーが強制されます。
#       permissive - SELinux が強制される代わりに警告を表示します。
#       disabled - SELinux ポリシーが読み込まれません。
SELINUX=enforcing
# SELINUXTYPE= これら 2 つの値のどれかを使用できます:
#       targeted - ターゲットポリシーが保護されます。
#       mls - マルチレベルセキュリティ保護。
SELINUXTYPE=targeted
```

SELINUX=enforcing
: SELINUX オプションが SELinux の動作モードを設定します。SELinux は 3 つのモードがあります: エンフォース、パーミッシブ、および無効です。エンフォースモードを使用しているとき、SELinux ポリシーが強制され、SELinux が SELinux ポリシールール基づいてアクセスを拒否し、拒否メッセージが記録されます。パーミッシブモードを使用しているとき、SELinux ポリシーが強制されません。SELinux がアクセスを拒否しませんが、SELinux がエンフォースモードで動作していれば、拒否されたはずのアクションを拒否したことが記録されます。無効モードで動作していると、SELinux が無効化されます (SELinux モジュールが Linux カーネルに登録されません)。DAC ルールのみが使用されます。

SELINUXTYPE=targeted
: SELINUXTYPE オプションが使用する SELinux ポリシーを設定します。ターゲットポリシーが標準のポリシーです。MLS ポリシーを使用したい場合のみ、このオプションを変更します。MLS ポリシーを使用するには、*selinux-policy-mls* パッケージをインストールし、/etc/selinux/config において SELINUXTYPE=mls を設定し、システムを再起動します。

重要

システムが パーミッシブモードまたは無効モードで動作していると、ユーザーが不適切なラベルのファイルにパーミッションを持ちます。また、SELinux が無効になっているときに作成されたファイルはラベルがつけられています。これは、エンフォースモードに変更するときに問題を引き起こします。不適切にラベルづけされたファイルやラベルづけされていないファイルが問題を引き起こすことを防ぐには、無効モードからパーミッシブモードやエンフォースモードに変更するとき、ファイルシステムが自動的に再ラベルづけします。

9.4.4. SELinux の有効化および無効化

SELinux の状態を確認するには /usr/sbin/getenforce または /usr/sbin/sestatus コマンドを使用します。getenforce コマンドが Enforcing, Permissive, または Disabled を返します。SELinux が有効なとき (SELinux ポリシールールがエンフォースであるとき)、getenforce コマンドが Enforcing を返します:

```
$ /usr/sbin/getenforce
Enforcing
```

SELinux が有効になっているとき、getenforce コマンドが Permissive を返します。しかし、SELinux ポリシールールがエンフォースではありません。DAC ルールのみが使用されます。SELinux が無効になっていると、getenforce コマンドが Disabled を返します。

sestatus コマンドが SELinux の状態と使用される SELinux ポリシーを返します:

```
$ /usr/sbin/sestatus
SELinux status:                 enabled
SELinuxfs mount:                /selinux
Current mode:                   enforcing
Mode from config file:          enforcing
Policy version:                 23
```

```
Policy from config file:        targeted
```

SELinux が有効化されているとき、SELinux status: enabled が返されます。SELinux がエンフォースモードで実行されているとき、Current mode: enforcing が返されます。SELinux ターゲットポリシーが使用されているとき、Policy from config file: targeted が返されます。

9.4.4.1. SELinux の有効化

SELinux を無効化されたシステムにおいて、SELINUX=disabled オプションが /etc/selinux/config に設定されています:

```
# このファイルがシステムにおける SELinux の状態を制御します。
# SELINUX= これら 3 つの値のどれかを使用できます:
#       enforcing - SELinux セキュリティポリシーが強制されます。
#       permissive - SELinux が強制される代わりに警告を表示します。
#       disabled - SELinux ポリシーが読み込まれません。
SELINUX=disabled
# SELINUXTYPE= これら 2 つの値のどれかを使用できます:
#       targeted - ターゲットポリシーが保護されます。
#       mls - マルチレベルセキュリティ保護。
SELINUXTYPE=targeted
```

また、getenforce コマンドが Disabled を返します:

```
$ /usr/sbin/getenforce
Disabled
```

SELinux を有効化するには:

1. SELinux パッケージがインストールされていることを確認するには、rpm -qa | grep selinux, rpm -q policycoreutils, および rpm -qa | grep setroubleshoot コマンドを使用します。このガイドは以下のパッケージがインストールされていることを仮定します: *selinux-policy-targeted*, *selinux-policy*, *libselinux*, *libselinux-python*, *libselinux-utils*, *policycoreutils*, *setroubleshoot*, *setroubleshoot-server*, *setroubleshoot-plugins*。これらのパッケージがインストールされていなければ、Linux root ユーザーとして yum install ***package-name*** 経由でインストールします。以下のパッケージはオプションです: *policycoreutils-gui*, *setroubleshoot*, *selinux-policy-devel*, および *mcstrans*。

2. SELinux が有効になる前に、ファイルシステムにある各ファイルが SELinux コンテキストでラベル付けされている必要があります。これをする前は、制限されたドメインがアクセスを拒否される可能性があります。これにより、システムの正常なブートを妨げられます。これを防ぐには、/etc/selinux/config に SELINUX=permissive を設定します:

   ```
   # This file controls the state of SELinux on the system.
   # SELINUX= can take one of these three values:
   #       enforcing - SELinux security policy is enforced.
   #       permissive - SELinux prints warnings instead of enforcing.
   #       disabled - No SELinux policy is loaded.
   SELINUX=permissive
   # SELINUXTYPE= can take one of these two values:
   #       targeted - Targeted processes are protected,
   #       mls - Multi Level Security protection.
   SELINUXTYPE=targeted
   ```

3. システムを再起動するには、Linux root ユーザーとして reboot コマンドを実行します。次回起動時に、ファイルシステムがラベル付けされます。ラベルプロセスがすべてのファイルを SELinux コンテキストでラベル付けします:

   ```
   *** Warning -- SELinux targeted policy relabel is required.
   ```

```
*** Relabeling could take a very long time, depending on file
*** system size and speed of hard drives.
****
```

最終行にある各 * 文字は、ラベル付けされた 1000 ファイルを表します。上の例では、4 つの * 文字が 4000 ファイルがラベル付けされていることを表します。すべてのファイルにラベルを付けるために必要な時間は、システムにあるファイル数とハードディスクの速度に依存します。最近のシステムでは、この処理が 10 分ほどしかかかりません。

4. パーミッシブモードでは、SELinux ポリシーがエンフォースされません。しかし、エンフォースモードで動作していると、拒否されるはずのアクションに対して、拒否メッセージが記録されます。エンフォースモードに変更する前に、SELinux が前回の起動時にアクションを拒否していないことを確認するために、Linux root ユーザーとして grep "SELinux is preventing" /var/log/messages コマンドを実行します。SELinux が前回の起動時にアクションを拒否していなければ、このコマンドが何も出力しません。SELinux が起動中にアクセスを拒否していれば、情報をトラブルシューティングするために##############を参照してください。

5. /var/log/messages に拒否メッセージがなければ、/etc/selinux/config において SELINUX=enforcing を設定します:

```
# このファイルがシステムにおける SELinux の状態を制御します。
# SELINUX= これら 3 つの値のどれかを使用できます:
#       enforcing - SELinux セキュリティポリシーが強制されます。
#       permissive - SELinux が強制される代わりに警告を表示します。
#       disabled - SELinux ポリシーが読み込まれません。
SELINUX=enforcing
# SELINUXTYPE= これら 2 つの値のどれかを使用できます:
#       targeted - ターゲットポリシーが保護されます。
#       mls - マルチレベルセキュリティ保護。
SELINUXTYPE=targeted
```

6. システムを再起動します。再起動後、getenforce コマンドが Enforcing が返すことを確認します:

```
$ /usr/sbin/getenforce
Enforcing
```

7. SELinux と Linux ユーザーの対応付けを表示するには、Linux root ユーザーとして /usr/sbin/semanage login -l コマンドを実行します。出力が次のようになります:

```
Login Name           SELinux User         MLS/MCS Range

__default__          unconfined_u         s0-s0:c0.c1023
root                 unconfined_u         s0-s0:c0.c1023
system_u             system_u             s0-s0:c0.c1023
```

こうなっていなければ、ユーザーの対応づけを修正するために、Linux root ユーザーとして以下のコマンドを実行します。SELinux-user ***username*** is already defined が発生しても、それらの警告を無視しても安全です。ここで、*username* は unconfined_u, guest_u, または xguest_u です:

1. ```
   /usr/sbin/semanage user -a -S targeted -P user -R "unconfined_r system_r" -r s0-s0:c0.c1023 unconfined_u
   ```

2. ```
   /usr/sbin/semanage login -m -S targeted -s "unconfined_u" -r s0-s0:c0.c1023 __default__
   ```

3. ```
   /usr/sbin/semanage login -m -S targeted -s "unconfined_u" -r s0-s0:c0.c1023 root
   ```
   ```

4.
```
/usr/sbin/semanage user -a -S targeted -P user -R guest_r guest_u
```

5.
```
/usr/sbin/semanage user -a -S targeted  -P user -R xguest_r xguest_u
```

重要

システムが パーミッシブモードまたは無効モードで動作していると、ユーザーが不適切なラベルのファイルにパーミッションを持ちます。また、SELinux が無効になっているときに作成されたファイルはラベルがつけられています。これは、エンフォースモードに変更するときに問題を引き起こします。不適切にラベルづけされたファイルやラベルづけされていないファイルが問題を引き起こすことを防ぐには、無効モードからパーミッシブモードやエンフォースモードに変更するとき、ファイルシステムが自動的に再ラベルづけします。

9.4.4.2. SELinux の無効化

SELinux を無効化するには、/etc/selinux/config において SELINUX=disabled を設定します:

```
# このファイルがシステムにおける SELinux の状態を制御します。
# SELINUX= これら 3 つの値のどれかを使用できます:
#       enforcing - SELinux セキュリティポリシーが強制されます。
#       permissive - SELinux が強制される代わりに警告を表示します。
#       disabled - SELinux ポリシーが読み込まれません。
SELINUX=disabled
# SELINUXTYPE= これら 2 つの値のどれかを使用できます:
#       targeted - ターゲットポリシーが保護されます。
#       mls - マルチレベルセキュリティ保護。
SELINUXTYPE=targeted
```

システムを再起動します。再起動後、getenforce コマンドが Disabled を返すことを確認します:

```
$ /usr/sbin/getenforce
Disabled
```

9.4.5. SELinux モード

SELinux は 3 つのモードがあります:

- Enforcing: SELinux ポリシーが強制されます。SELinux が SELinux ポリシーに基づいてアクセスを拒否されます。
- Permissive: SELinux ポリシーが強制されません。SELinux がアクセスを拒否しません。しかし、エンフォースモードで実行していると、拒否されるアクションに対して、拒否したことがログに記録されます。
- Disabled: SELinux が無効化されます。DAC ルールのみが使用されます。

エンフォースモードとパーミッシブモードの間で変更するには、/usr/sbin/setenforce コマンドを使用します。/usr/sbin/setenforce で行った変更は、再起動後は反映されません。エンフォースモードに変更するには、Linux root ユーザーとして /usr/sbin/setenforce 1 コマンドを実行します。パーミッシブモードに変更するには、/usr/sbin/setenforce 0 コマンドを実行します。現在の SELinux モードを表示するには /usr/sbin/getenforce コマンドを使用します。

永続的なモードの変更は*#SELinux ############*において取り扱われています。

9.4.6. ブーリアン

ブーリアンにより、SELinux ポリシー作成の知識なしで、SELinux ポリシーの一部を実行時に変更できます。これは SELinux ポリシーの再読み込みまたは再コンパイルをすることなく、サービスが NFS ファイルシステムにアクセスできるようにするなどの、変更をできるようになります。

9.4.6.1. ブーリアンの一覧表示

ブーリアン、その説明、およびオンかオフかの一覧は、Linux root ユーザーとして `semanage boolean -l` を実行します。以下の例はすべてのブーリアンを一覧表示しません:

```
# /usr/sbin/semanage boolean -l
SELinux boolean                          Description

ftp_home_dir                -> off   Allow ftp to read and write files in the user home directories
xen_use_nfs                 -> off   Allow xen to manage nfs files
xguest_connect_network      -> on    Allow xguest to configure Network Manager
```

SELinux boolean 列がブーリアン名を一覧表示します。Description 列が、ブーリアンがオンかオフか、およびそれが何であるかを一覧表示します。

以下の例において、ftp_home_dir ブーリアンがオフで、SELinux により FTP デーモン (vsftpd) がホームディレクトリを読み込みおよび書き込みすることを拒否されます:

```
ftp_home_dir                -> off   Allow ftp to read and write files in the user home directories
```

`getsebool -a` コマンドが、ブーリアン、それらがオンかオフかについて一覧表示します。それぞれの説明は与えられません。以下の例はすべてのブーリアンを一覧表示しません:

```
$ /usr/sbin/getsebool -a
allow_console_login --> off
allow_cvs_read_shadow --> off
allow_daemons_dump_core --> on
```

`boolean-name` ブーリアンの状態のみを一覧表示するには `getsebool` ***`boolean-name`*** コマンドを実行します:

```
$ /usr/sbin/getsebool allow_console_login
allow_console_login --> off
```

複数のブーリアンを一覧化するには、空白区切りの一覧を使用します:

```
$ getsebool allow_console_login allow_cvs_read_shadow allow_daemons_dump_core
allow_console_login --> off
allow_cvs_read_shadow --> off
allow_daemons_dump_core --> on
```

9.4.6.2. ブーリアンの設定

`setsebool` ***`boolean-name x`*** コマンドがブーリアンをオンまたはオフにします。ここで *`boolean-name`* がブーリアンの名前です。*`x`* は、オンにするには `on` に、オフにするには `off` にします。

以下の例は `httpd_can_network_connect_db` ブーリアンを設定することを説明します:

1. `httpd_can_network_connect_db` ブーリアンが標準でオフです。これにより、Apache HTTP Server のスクリプトやモジュールがデータベースに接続できません:

```
$ /usr/sbin/getsebool httpd_can_network_connect_db
httpd_can_network_connect_db --> off
```

2. データベースファイルに接続するには Apache HTTP Server スクリプトとモジュールを一時的に有効化するために、Linux root ユーザーとして `setsebool httpd_can_network_connect_db on`コマンドを実行します。

3. ブーリアンがオンであることを確認するには、`getsebool httpd_can_network_connect_db` コマンドを使用します:

```
$ /usr/sbin/getsebool httpd_can_network_connect_db
httpd_can_network_connect_db --> on
```

これにより、Apache HTTP Server のスクリプトやモジュールがデータベースサーバーに接続できます。

4. この変更は再起動後に有効になりません。変更を再起動後も有効にするには、Linux root ユーザーとして `setsebool -P` ***`boolean-name`*** `on` コマンドを実行します:

```
# /usr/sbin/setsebool -P httpd_can_network_connect_db on
```

5. 一時的に標準の動作に戻すには、Linux root ユーザーとして `setsebool httpd_can_network_connect_db off` コマンドを実行します。変更を再起動後に有効にするには、`setsebool -P httpd_can_network_connect_db off` コマンドを実行します。

9.4.6.3. NFS および CIFS 向けのブーリアン

標準で、クライアント側にマウントされた NFS ファイルシステムが NFS ファイルシステム向けのポリシーにより定義された標準のコンテキストでラベルづけされます。一般的なポリシーにおいて、この標準のコンテキストが `nfs_t` タイプを使用します。また、クライアント側にマウントされた Samba 共有は、ポリシーにより定義された標準のコンテキストでラベルづけされます。一般的なポリシーにおいて、この標準のコンテキストは `cifs_t` タイプを使用します。

ポリシー設定に依存して、サービスが `nfs_t` や `cifs_t` タイプのラベルを持つファイルを読み込みできないかもしれません。これにより、これらのタイプを持つファイルシステムがマウントされ、他のサービスにより読み込みまたはエクスポートされることを防げます。`nfs_t` および `cifs_t` タイプにアクセスすることを許可するサービスを制御するために、ブーリアンをオンまたはオフにできます。

`setsebool` コマンドと `semanage` コマンドは Linux root ユーザーとして実行する必要があります。`setsebool -P` コマンドが変更を永続化させます。再起動後に変更を反映したくなければ、`-P` オプションを使用しないでください:

Apache HTTP Server

NFS ファイルシステム (`nfs_t` タイプのラベルを持つファイル) にアクセスを許可するには:

```
/usr/sbin/setsebool -P httpd_use_nfs on
```

Samba ファイルシステム (`cifs_t` タイプのラベルを持つファイル) にアクセスを許可するには:

```
/usr/sbin/setsebool -P httpd_use_cifs on
```

Samba

NFS ファイルシステムをエクスポートするには:

```
/usr/sbin/setsebool -P samba_share_nfs on
```

FTP (vsftpd)

NFS ファイルシステムにアクセスを許可するには:

```
/usr/sbin/setsebool -P allow_ftpd_use_nfs on
```

Samba ファイルシステムにアクセスを許可するには:

```
/usr/sbin/setsebool -P allow_ftpd_use_cifs on
```

他のサービス

他のサービスに対する NFS 関連のブーリアンの一覧は:

```
/usr/sbin/semanage boolean -l | grep nfs
```

他のサービスに対する Samba 関連のブーリアンの一覧は:

```
/usr/sbin/semanage boolean -l | grep cifs
```

注記

これらのブーリアンが Fedora 19 に同梱される SELinux ポリシーにあります。これらは他のバージョン Fedora または他のオペレーティングシステムに同梱されるポリシーに存在しない可能性があります。

SELinux ブーリアンに関する詳細は *http://docs.fedoraproject.org* にある SELinux Managing Confined Services Guide を参照してください。

9.4.7. SELinux コンテキスト - ファイルのラベルづけ

SELinux を実行しているシステムにおいて、すべてのプロセスとファイルがセキュリティ関連情報を示す方法でラベル付けされています。この情報は SELinux コンテキストと呼ばれます。これはファイルに対して `ls -Z` コマンドを使用して表示されます:

```
$ ls -Z file1
-rw-rw-r--  user1 group1 unconfined_u:object_r:user_home_t:s0 file1
```

この例では、SELinux がユーザー (unconfined_u)、ロール (object_r)、タイプ (user_home_t) およびレベル (s0) を提供します。この情報はアクセス制御の判定のために使用されます。DAC システムにおいて、アクセス権が Linux ユーザーとグループ ID のみに基づいて制御されます。SELinux ポリシールールは DAC ルールの後に確認されます。まず DAC ルールがアクセスを拒否すれば、SELinux ポリシールールが使用されません。

ファイルの SELinux コンテキストを管理するために、chcon, `semanage fcontext`, および `restorecon` のような、複数のコマンドがあります。

9.4.7.1. 一時的な変更: chcon

chcon コマンドがファイル向けの SELinux コンテキストを変更します。しかしながら、chcon で行った変更は、ファイルシステムの再ラベルや `/sbin/restorecon` コマンドにより消されます。SELinux ポリシーにより、ユーザーがすべての任意のファイルに対して SELinux コンテキストを変更できるかどうかを制御されます。chcon を使用するとき、ユーザーが変更する SELinux コンテキストのすべてまたは一部を提供します。不適切なファイルタイプは、SELinux がアクセスを拒否する一般的な原因となります。

クイックリファレンス

- ファイルのタイプを変更するには chcon -t ***type file-name*** コマンドを実行します。ここで *type* は httpd_sys_content_t のようなタイプで、*file-name* はファイルやディレクトリの名前です。
- ディレクトリとその中のタイプを変更するには chcon -R -t ***type directory-name*** コマンドを実行します。ここで *type* は httpd_sys_content_t のようなタイプで、*directory-name* はディレクトリ名です。

ファイルまたはディレクトリのタイプの変更

以下の例はタイプの変更について説明します。SELinux コンテキストの他の属性は変更しません:

1. ホームディレクトリの中に変更するには cd コマンドを引数なしで実行します。
2. 新しいファイルを作成するには touch file1 コマンドを実行します。file1 の SELinux コンテキストを表示するには ls -Z file1 コマンドを使用します:

   ```
   $ ls -Z file1
   -rw-rw-r--  user1 group1 unconfined_u:object_r:user_home_t:s0 file1
   ```

   この例では、file1 向けの SELinux コンテキストが、SELinux unconfined_u ユーザー、object_r ロール、user_home_t タイプ、および s0 レベルを含みます。SELinux コンテキストのそれぞれの部分に関する説明は*#SELinux #######*を参照してください。
3. タイプを samba_share_t に変更するには chcon -t samba_share_t file1 コマンドを実行します。-t オプションはタイプのみを変更します。ls -Z file1 で変更を表示します:

   ```
   $ ls -Z file1
   -rw-rw-r--  user1 group1 unconfined_u:object_r:samba_share_t:s0 file1
   ```
4. file1 ファイルに関する SELinux コンテキストを復元するには /sbin/restorecon -v file1 コマンドを使用します。変更点を表示するには -v オプションを使用します:

   ```
   $ /sbin/restorecon -v file1
   restorecon reset file1 context unconfined_u:object_r:samba_share_t:s0->system_u:object_r:user_home_t:s0
   ```

   この例では、前のタイプ samba_share_t が正しいタイプ user_home_t に復元されます。ターゲットポリシー (Fedora 19 の標準 SELinux ポリシー) を使用するとき、ファイルが持つべき SELinux コンテキストを確認するために、/sbin/restorecon コマンドが /etc/selinux/targeted/contexts/files/ ディレクトリにあるファイルを読み込みます。

このセクションにある例は、たとえば、file1 がディレクトリならば、ディレクトリに対して同じように動作します。

ディレクトリおよびそのコンテキストタイプの変更

以下の例は、新しいディレクトリを作成し、(内容をそのままで) ディレクトリのファイルタイプを Apache HTTP Server により使用されるタイプに変更することについて説明します。Apache HTTP Server が (/var/www/html/ の代わりに) 違うドキュメントルートを使用したいならば、この例にある設定が使用されます:

1. 新しいディレクトリを作成するために、Linux root ユーザーとして mkdir /web コマンドを実行します。そして、空のファイル 3 個 (file1, file2, および file3) を作成するために、touch /web/file{1,2,3} コマンドを実行します。そこにある /web/ ディレクトリとファイルは default_t タイプのラベルを付けられます:

   ```
   # ls -dZ /web
   drwxr-xr-x  root root unconfined_u:object_r:default_t:s0 /web
   # ls -lZ /web
   -rw-r--r--  root root unconfined_u:object_r:default_t:s0 file1
   ```

```
-rw-r--r--  root root unconfined_u:object_r:default_t:s0 file2
-rw-r--r--  root root unconfined_u:object_r:default_t:s0 file3
```

2. /web/ ディレクトリ (とその中身) のタイプを httpd_sys_content_t に変更するには、Linux root ユーザーとして chcon -R -t httpd_sys_content_t /web/ コマンドを実行します:

```
# chcon -R -t httpd_sys_content_t /web/
# ls -dZ /web/
drwxr-xr-x  root root unconfined_u:object_r:httpd_sys_content_t:s0 /web/
# ls -lZ /web/
-rw-r--r--  root root unconfined_u:object_r:httpd_sys_content_t:s0 file1
-rw-r--r--  root root unconfined_u:object_r:httpd_sys_content_t:s0 file2
-rw-r--r--  root root unconfined_u:object_r:httpd_sys_content_t:s0 file3
```

3. 標準の SELinux コンテキストを復元するには、Linux root ユーザーとして /sbin/restorecon -R -v /web/ コマンドを実行します。

```
# /sbin/restorecon -R -v /web/
restorecon reset /web context unconfined_u:object_r:httpd_sys_content_t:s0->system_u:object_r:default_t:s0
restorecon reset /web/file2 context unconfined_u:object_r:httpd_sys_content_t:s0-
>system_u:object_r:default_t:s0
restorecon reset /web/file3 context unconfined_u:object_r:httpd_sys_content_t:s0-
>system_u:object_r:default_t:s0
restorecon reset /web/file1 context unconfined_u:object_r:httpd_sys_content_t:s0-
>system_u:object_r:default_t:s0
```

chcon に関する詳細は chcon(1) マニュアルページを参照してください。

注記

タイプエンフォースメントが SELinux ターゲットポリシーで使用されるメインのパーミッション制御です。ほとんどの場合、SELinux ユーザーとロールが無視できます。

9.4.7.2. 永続的な変更: semanage fcontext

/usr/sbin/semanage fcontext コマンドがファイルの SELinux コンテキストを変更します。ターゲットポリシーを使用するとき、変更が file_contexts に存在するファイルに対するものであれば、または、/web/ ディレクトリを作成するなど、新しいファイルやディレクトリに対して file_contexts.local に追加されるならば、このコマンドで行われた変更が /etc/selinux/targeted/contexts/files/file_contexts ファイルに追加されます。これは、ファイルシステムが再ラベル付けされるときに使用される、setfiles がこれらのファイルを読み込みます。また、標準の SELinux コンテキストを復元する、¥n/sbin/restorecon がこれらのファイルを読み込みます。これが、ファイルシステムが再ラベル付けされたときでも、/usr/sbin/semanage fcontext により行われた変更が継続される理由です。ユーザーが任意のファイルの SELinux コンテキストを変更できるかどうかを、SELinux ポリシーが制御します。

クイックリファレンス

SELinux コンテキストをファイルシステムの再ラベルづけ後に継続する変更させるには:

1. ファイルやディレクトリの完全パスを使用することを思い出し、/usr/sbin/semanage fcontext -a ***options file-name|directory-name*** コマンドを実行します。

2. コンテキストの変更を適用するには /sbin/restorecon -v ***file-name|directory-name*** コマンドを実行します。

ファイルのタイプの変更

以下の例はファイルのタイプの変更について説明します。SELinux コンテキストの他の属性は変更しません:

1. 新しいファイルを作成するには、Linux root ユーザーとして touch /etc/file1 コマンドを実行します。/etc/ ディレクトリに新しく作成されたファイルは、標準で etc_t タイプのラベルを付けられます:

```
# ls -Z /etc/file1
-rw-r--r--  root root unconfined_u:object_r:etc_t:s0       /etc/file1
```

2. file1 のタイプを samba_share_t に変更するには、Linux root ユーザーとして /usr/sbin/semanage fcontext -a -t samba_share_t /etc/file1 コマンドを実行します。-a オプションが新しいレコードを追加します。-t オプションがタイプ (samba_share_t) を定義します。注: このコマンドを実行しても、タイプを直接変更しません - file1 はまだ etc_t タイプのラベルを付けられています:

```
# /usr/sbin/semanage fcontext -a -t samba_share_t /etc/file1
# ls -Z /etc/file1
-rw-r--r--  root root unconfined_u:object_r:etc_t:s0       /etc/file1
```

   /usr/sbin/semanage fcontext -a -t samba_share_t /etc/file1 コマンドが以下の項目を /etc/selinux/targeted/contexts/files/file_contexts.local に追加します:

```
/etc/file1    unconfined_u:object_r:samba_share_t:s0
```

3. タイプを変更するには Linux root ユーザーとして /sbin/restorecon -v /etc/file1 コマンドを実行します。semanage コマンドが /etc/file1 向けのエントリーを file.contexts.local に追加するので、/sbin/restorecon コマンドがタイプを samba_share_t に変更します:

```
# /sbin/restorecon -v /etc/file1
restorecon reset /etc/file1 context unconfined_u:object_r:etc_t:s0->system_u:object_r:samba_share_t:s0
```

4. Linux root ユーザーとして、file1 を削除するには rm -i /etc/file1 コマンドを実行します。

5. /etc/file1 向けに追加されたコンテキストを削除するには、Linux root ユーザーとして /usr/sbin/semanage fcontext -d /etc/file1 コマンドを実行します。コンテキストが削除されるとき、restorecon を実行することにより、タイプを samba_share_t から etc_t に変更します。

ディレクトリのタイプの変更

以下の例は、新しいディレクトリを作成し、そのディレクトリのファイルタイプを Apache HTTP Server により使用されるタイプに変更することを説明します:

1. 新しいディレクトリを作成するには、Linux root ユーザーとして mkdir /web コマンドを実行します。このディレクトリは default_t タイプのラベルを付けられています:

```
# ls -dZ /web
drwxr-xr-x  root root unconfined_u:object_r:default_t:s0 /web
```

   ls -d オプションにより、ディレクトリの内容ではなく、そのディレクトリに関する情報を ls 一覧表示します。-Z オプションにより SELinux コンテキスト (この例では unconfined_u:object_r:default_t:s0) を ls 表示します。

2. /web/ のタイプを httpd_sys_content_t に変更するには、Linux root ユーザーとして /usr/sbin/semanage fcontext -a -t httpd_sys_content_t /web コマンドを実行します。-a オプションが新しいレ

コードを追加します。-t オプションがタイプ (httpd_sys_content_t) を定義します。注: このコマンドを実行しても、タイプを直接変更しません - /web/ はまだ default_t タイプのラベルを付けられています:

```
# /usr/sbin/semanage fcontext -a -t httpd_sys_content_t /web
# ls -dZ /web
drwxr-xr-x  root root unconfined_u:object_r:default_t:s0   /web
```

/usr/sbin/semanage fcontext -a -t httpd_sys_content_t /web コマンドが以下の項目を /etc/selinux/targeted/contexts/files/file_contexts.local に追加します:

```
/web    unconfined_u:object_r:httpd_sys_content_t:s0
```

3. タイプを変更するには、Linux root ユーザーとして /sbin/restorecon -v /web コマンドを実行します。semanage コマンドが /web 向けのエントリーを file.contexts.local に追加するので、/sbin/restorecon コマンドがタイプを httpd_sys_content_t に変更します:

```
# /sbin/restorecon -v /web
restorecon reset /web context unconfined_u:object_r:default_t:s0->system_u:object_r:httpd_sys_content_t:s0
```

新しく作成されたファイルとディレクトリは標準で、親ディレクトリの SELinux タイプを継承します。この例を使用するとき、および、/web/ 向けに追加された SELinux コンテキストを削除する前に、/web/ディレクトリに作成されたファイルとディレクトリが httpd_sys_content_t タイプのラベルを付けられます。

4. /web/ に追加されたコンテキストを削除するには、Linux root ユーザーとして /usr/sbin/semanage fcontext -d /web コマンドを実行します。

5. 標準の SELinux コンテキストを復元するには、Linux root ユーザーとして /sbin/restorecon -v /web コマンドを実行します。

ディレクトリおよびそのコンテキストタイプの変更

以下の例は、新しいディレクトリを作成し、(内容をそのままで) ディレクトリのファイルタイプを Apache HTTP Server により使用されるタイプに変更することについて説明します。Apache HTTP Server が (/var/www/html/ の代わりに) 違うドキュメントルートを使用したいならば、この例にある設定が使用されます:

1. 新しいディレクトリを作成するために、Linux root ユーザーとして mkdir /web コマンドを実行します。そして、空のファイル 3 個 (file1, file2, および file3) を作成するために、touch /web/file{1,2,3} コマンドを実行します。そこにある /web/ ディレクトリとファイルは default_t タイプのラベルを付けられます:

```
# ls -dZ /web
drwxr-xr-x  root root unconfined_u:object_r:default_t:s0 /web
# ls -lZ /web
-rw-r--r--  root root unconfined_u:object_r:default_t:s0 file1
-rw-r--r--  root root unconfined_u:object_r:default_t:s0 file2
-rw-r--r--  root root unconfined_u:object_r:default_t:s0 file3
```

2. /web/ ディレクトリおよびその中にあるファイルのタイプを httpd_sys_content_t に変更するには、Linux root ユーザーとして /usr/sbin/semanage fcontext -a -t httpd_sys_content_t "/web(/.*)?" コマンドを実行します。-a オプションが新しいレコードを追加します。-t オプションがタイプ (httpd_sys_content_t) を定義します。"/web(/.*)?" 正規表現により、semanage コマンドが変更を /web/ ディレクトリおよびその中にあるファイルに適用されます。注: このコマンドを実行しても、タイプを直接変更しません - /web/ およびその中にあるファイルは、まだ default_t タイプのラベルを付けられています:

```
# ls -dZ /web
drwxr-xr-x  root root unconfined_u:object_r:default_t:s0 /web
```

```
# ls -lZ /web
-rw-r--r--  root root unconfined_u:object_r:default_t:s0 file1
-rw-r--r--  root root unconfined_u:object_r:default_t:s0 file2
-rw-r--r--  root root unconfined_u:object_r:default_t:s0 file3
```

/usr/sbin/semanage fcontext -a -t httpd_sys_content_t "/web(/.*)?" コマンドが以下の項目を /etc/selinux/targeted/contexts/files/file_contexts.local に追加します:

```
/web(/.*)?    system_u:object_r:httpd_sys_content_t:s0
```

3. /web/ ディレクトリおよびその中にあるファイルのタイプを変更するには、Linux root ユーザーとして /sbin/restorecon -R -v /web コマンドを実行します。-R オプションが再帰的、つまり /web/ およびその下にあるすべてのファイルとディレクトリが httpd_sys_content_t タイプのラベルを付けられることを意味します。semanage コマンドがエントリーを /web(/.*)? 向けの file.contexts.local に追加するので、/sbin/restorecon コマンドがタイプを httpd_sys_content_t に変更します:

```
# /sbin/restorecon -R -v /web
restorecon reset /web context unconfined_u:object_r:default_t:s0->system_u:object_r:httpd_sys_content_t:s0
restorecon reset /web/file2 context unconfined_u:object_r:default_t:s0-
>system_u:object_r:httpd_sys_content_t:s0
restorecon reset /web/file3 context unconfined_u:object_r:default_t:s0-
>system_u:object_r:httpd_sys_content_t:s0
restorecon reset /web/file1 context unconfined_u:object_r:default_t:s0-
>system_u:object_r:httpd_sys_content_t:s0
```

新しく作成されたファイルとディレクトリは標準で、親の SELinux タイプを継承します。この例では、/web/ ディレクトリに作成されたファイルとディレクトリが httpd_sys_content_t タイプのラベルを付けられます。

4. "/web(/.*)?" に追加されたコンテキストを削除するには、Linux root ユーザーとして /usr/sbin/semanage fcontext -d "/web(/.*)?" コマンドを実行します。

5. 標準の SELinux コンテキストを復元するには、Linux root ユーザーとして /sbin/restorecon -R -v /web コマンドを実行します。

追加されたコンテキストの削除

以下の例は SELinux コンテキストの追加および削除について説明しています:

1. Linux root ユーザーとして /usr/sbin/semanage fcontext -a -t httpd_sys_content_t /test コマンドを実行します。/test/ ディレクトリが存在する必要がありません。このコマンドが以下のコンテキストを /etc/selinux/targeted/contexts/files/file_contexts.local に追加します:

```
/test    system_u:object_r:httpd_sys_content_t:s0
```

2. コンテキストを削除するには、Linux root ユーザーとして /usr/sbin/semanage fcontext -d ***file-name|directory-name*** コマンドを実行します。ここで、*file-name|directory-name* は file_contexts.local にある最初の部分です。以下は file_contexts.local にあるコンテキストの例です:

```
/test    system_u:object_r:httpd_sys_content_t:s0
```

最初の部分が /test になっていると、/sbin/restorecon を実行した後、またはファイルシステムが再ラベルづけをされた後、/test/ ディレクトリが httpd_sys_content_t のラベルを付けられることを防ぐには、コンテキストを file_contexts.local から削除するために、Linux root ユーザーとして以下のコマンドを実行します:

```
/usr/sbin/semanage fcontext -d /test
```

コンテキストが表記表現の一部分であれば、たとえば /web(/.*)? ならば、正規表現の前後に引用符を使用します:

```
/usr/sbin/semanage fcontext -d "/web(/.*)?"
```

/usr/sbin/semanage の詳細は semanage(8) マニュアルページを参照してください。

重要

SELinux コンテキストを /usr/sbin/semanage fcontext -a で変更するとき、ファイルシステムの再ラベル後、または /sbin/restorecon コマンドの実行後に、ファイルに誤ったラベルを付けられることを避けるために、ファイルやディレクトリの完全パスを使用します。

9.4.8. file_t および default_t のタイプ

拡張属性をサポートするファイルシステムに対して、ディスクに SELinux コンテキストのないファイルにアクセスするとき、SELinux ポリシーにより定義された標準のコンテキストがあるかのように取り扱われます。一般的なポリシーでは、この標準のコンテキストは file_t タイプを使用します。ディスクにあるコンテキストのないファイルがポリシーで区別でき、一般的に制限されたドメインにアクセスできないよう、これはこのタイプのみに使用されます。SELinux を実行しているシステムにあるすべてのファイルが SELinux コンテキストを持つべきであり、file_t タイプがファイルとコンテキストの設定 [30] において使用されていないので、file_t タイプが正しくラベルづけされたファイルシステムに存在すべきではありません。

default_t タイプが、ファイルとコンテキストの設定において、他のすべてのパターンと一致しないファイルに使用されます。そのようなファイルがディスクにコンテキストのないファイルと区別でき、一般的に制限されたドメインにアクセスできないようにするためです。/mydirectory/ のような新しいトップディレクトリを作成すれば、このディレクトリが default_t タイプのラベルを付けられます。サービスがそのようなディレクトリにアクセスする必要がなければ、この位置に対するファイルとコンテキストの設定を更新します。ファイルとコンテキストの設定にコンテキストを追加することに関する詳細は*#######: semanage fcontext#*を参照してください。

9.4.9. ファイルシステムのマウント

拡張属性をサポートするファイルシステムをマウントするとき、各ファイルに対するセキュリティコンテキストが、標準でファイルの *security.selinux* 拡張属性から取得されます。拡張属性をサポートしないファイルシステムにあるファイルは、ファイルシステムの種類に基づいて、ポリシー設定から単一かつ標準のセキュリティコンテキストをポリシーから割り当てられます。

既存の拡張属性を上書きするため、または拡張属性をサポートしないファイルシステムに対する標準と異なるコンテキストを指定するために、mount -o context コマンドを使用します。ファイルシステムが正しい属性を提供することを信頼できなければ、これは有用です。たとえば、複数のシステムにおいて使用されるリムーバブルメディアです。File Allocation Table (FAT) や NFS ファイルシステムのような、拡張属性をサポートしないファイルシステムに対して、mount -o context コマンドがラベル付けをサポートするために使用できます。context で指定されたコンテキストはディスクに書き込まれません。元のコンテキストが保持されます。(ファイルシステムがそもそも拡張属性を持っていれば、) context オプションなしでマウントしたときに確認できます。

[30] /etc/selinux/targeted/contexts/files/ にあるファイルが、ファイルとディレクトリのためのコンテキストを定義します。このディレクトリにあるファイルが、ファイルとディレクトリの標準コンテキストを復元するために restorecon および setfiles により読み込まれます。

ファイルシステムのラベル付けに関する詳細は、James Morris の "Filesystem Labeling in SELinux" 記事を参照してください: *http://www.linuxjournal.com/article/7426*。

9.4.9.1. コンテキストのマウント

既存のコンテキストが存在すれば、それを上書きして、指定されたコンテキストを持つファイルシステムをマウントするために、または拡張属性をサポートしないファイルシステムに対して標準と異なるコンテキストを指定するために、希望するファイルシステムをマウントするときに、Linux root ユーザーとして `mount -o context=`***`SELinux_user:role:type:level`*** コマンドを使用します。コンテキストの変更がディスクに書き込まれません。クライアント側における NFS マウントが標準で、NFS ファイルシステムに対するポリシーにより定義された標準のコンテキストでラベル付けされます。一般的なポリシーにおいて、この標準のコンテキストが `nfs_t` タイプを使用します。追加のマウントオプションがなければ、Apache HTTP Server のような、他のサービス経由で NFS ファイルシステムを共有することを防がれます。Apache HTTP Server 経由で共有できるようにするために、以下の例は NFS ファイルシステムをマウントします:

```
# mount server:/export /local/mount/point -o¥
context="system_u:object_r:httpd_sys_content_t:s0"
```

このファイルシステムに新しく作成されたファイルおよびディレクトリが、`-o context` で指定された SELinux コンテキストを持ちます。しかしながら、コンテキストの変更がこれらの状況に対してディスクに書かれないので、`context` オプションが次回マウント時に使用された場合、または同じコンテキストが指定された場合のみ、`context` オプションで指定されたコンテキストが保持されます。

タイプエンフォースメントが SELinux ターゲットポリシーにおいて使用されるメインのパーミッション制御です。多くの部分に対して、SELinux ユーザーとロールが無視されます。そのため、`-o context` で SELinux コンテキストを上書きするとき、SELinux `system_u` ユーザーおよび `object_r` ロールを使用し、タイプに集中します。MLS ポリシーやマルチカテゴリーセキュリティを使用しなければ、`s0` レベルを使用します。

注記

ファイルシステムが `context` オプションとともにマウントされるとき、(ユーザーやプロセスによる) コンテキストの変更が禁止されます。たとえば、`context` オプションとともにマウントされたファイルシステムにおいて `chcon` を実行することにより、`Operation not supported` エラーが発生します。

9.4.9.2. 標準コンテキストの変更

*#file_t ### default_t #####*に説明されているとおり、拡張属性をサポートするファイルシステムにおいて、ディスクに SELinux コンテキストのないファイルにアクセスするとき、SELinux ポリシーにより定義された標準のコンテキストを持っているように取り扱われます。一般的なポリシーにおいて、この標準コンテキストは `file_t` タイプを使用します。他の標準コンテキストを使用したければ、ファイルシステムを `defcontext` オプションとともにマウントします。

以下の例は新しく作成されたファイルシステム (/dev/sda2 上) を新しく作成された `/test/` ディレクトリ にマウントします。`/test/` ディレクトリ向けのコンテキストを定義する `/etc/selinux/targeted/contexts/files/` ルールがないと仮定します:

```
# mount /dev/sda2 /test/ -o defcontext="system_u:object_r:samba_share_t:s0"
```

この例では:

- defcontext オプションが、system_u:object_r:samba_share_t:s0 が "ラベルづけされていないファイルに対する標準のセキュリティコンテキスト"[31] であることを定義します。
- マウントしたとき、ファイルシステムのルートディレクトリ (/test/) が、defcontext により指定されたコンテキストをラベルづけされているかのように取り扱われます (このラベルはディスクに保存されません)。これは /test/ の下に作成したファイルに対するラベルづけに影響があります: 新しいファイルが samba_share_t タイプを継承し、これらのラベルはディスクに保存されます。
- ファイルシステムが defcontext オプションでマウントされている間、/test/ の下にあるファイルがそれらのラベルを保持します。

9.4.9.3. NFS ファイルシステムのマウント

標準で、クライアント側にマウントされた NFS ファイルシステムが NFS ファイルシステム向けのポリシーにより定義された標準のコンテキストでラベルづけされます。一般的なポリシーにおいて、この標準のコンテキストが nfs_t タイプを使用します。ポリシー設定に依存して、Apache HTTP Server や MySQL のようなサービスが、nfs_t タイプのラベルを持つファイルを読み込めません。これにより、このタイプのラベルを持つファイルシステムがマウントされ、他のサービスにより読み込みまたはエクスポートされることを防ぎます。

NFS ファイルシステムをマウントしたく、他のサービスを用いてそのファイルシステムを読み込みまたはエクスポートしていれば、nfs_t タイプを上書きするために、マウントするときに context オプションを使用します。NFS ファイルシステムを Apache HTTP Server 経由で共有できるよう、それらをマウントするために以下の context オプションを使用します:

```
mount server:/export /local/mount/point -o¥
context="system_u:object_r:httpd_sys_content_t:s0"
```

コンテキストの変更がこれらの状況に対してディスクに書かれないので、context オプションが次回マウント時に使用された場合、または同じコンテキストが指定された場合のみ、context オプションで指定されたコンテキストが保持されます。

context オプションでファイルシステムをマウントする代替として、サービスが nfs_t タイプのラベルを持つファイルシステムにアクセスすることを許可するために、ブーリアンをオンにできます。nfs_t タイプのラベルを持つファイルシステムにアクセスすることを許可する、ブーリアンを設定することの説明は*#NFS ### CIFS ########*##を参照してください。

9.4.9.4. 複数の NFS マウント

同じ NFS エクスポートから複数のマウントをするとき、それぞれのマウントの SELinux コンテキストを異なるコンテキストで上書きしようとすると、後続の mount コマンドが失敗します。以下の例では、NFS サーバーが 2 つのサブディレクトリ web/ と database/ を持つ、1 つのエクスポート /export を持ちます。以下のコマンドが、1 つの NFS エクスポートから 2 つのマウントを試行し、それぞれのコンテキストを上書きしようとします:

```
# mount server:/export/web /local/web -o¥
context="system_u:object_r:httpd_sys_content_t:s0"

# mount server:/export/database /local/database -o¥
context="system_u:object_r:mysqld_db_t:s0"
```

2 回目の mount コマンドが失敗しました。以下が /var/log/messages に記録されます:

[31] Morris, James. "Filesystem Labeling in SELinux". 2004年9月1日発行、2008年9月14日アクセス確認: *http://www.linuxjournal.com/article/7426*.

```
kernel: SELinux: mount invalid.  Same superblock, different security settings for (dev 0:15, type nfs)
```

それぞれ異なるコンテキストを用いて、1 つの NFS エクスポートから複数のマウントをするには、-o nosharecache,context オプションを使用します。以下の例は、それぞれ異なるコンテキストを用いて、(1 つのサービスがそれぞれにアクセスすることを許可して、) 1 つの NFS エクスポートから複数のマウントをします。

```
# mount server:/export/web /local/web -o¥
nosharecache,context="system_u:object_r:httpd_sys_content_t:s0"

# mount server:/export/database /local/database -o¥
nosharecache,context="system_u:object_r:mysqld_db_t:s0"
```

この例では、server:/export/web が /local/web/ にローカルにマウントされます。すべてのファイルが httpd_sys_content_t タイプのラベルを持ち、Apache HTTP Server がアクセスを許可されます。server:/export/database が /local/database にローカルにマウントされます。すべてのファイルが mysqld_db_t タイプを持ち、MySQL がアクセスを許可されます。これらのタイプはディスクに書き込まれません。

重要

nosharecache オプションにより、異なるコンテキストを用いて、1 つのエクスポートの同じサブディレクトリを複数回マウントできるようになります (たとえば、/export/web を複数回マウントします)。ファイルが 2 つの異なるコンテキストの下でアクセス可能である、重なりあったマウントが作成されるので、1 つのエクスポートから同じサブディレクトリを異なるコンテキストでマウントしないでください。

9.4.9.5. コンテキストのマウントの永続化

コンテキストのマウントを再マウントや再起動後に有効にするために、/etc/fstab またはオートマウントマップにファイルシステムのエントリーを追加します。また、マウントオプションとして期待するコンテキストを使用します。以下の例は、NFS コンテキストのマウントのために /etc/fstab にエントリーを追加します:

```
server:/export /local/mount/ nfs context="system_u:object_r:httpd_sys_content_t:s0" 0 0
```

NFS ファイルシステムのマウントに関する詳細は *Red Hat Enterprise Linux 5 Deployment Guide, ##### 19.2. "NFS Client Configuration"*[32] を参照してください。

9.4.10. SELinux ラベルのメンテナンス

これらのセクションは、ファイルやディレクトリをコピー、移動、アーカイブするときに、SELinux コンテキストに発生することを説明します。また、コピーおよびアーカイブするときに、コンテキストを保護する方法について説明します。

9.4.10.1. ファイルとディレクトリのコピー

ファイルやディレクトリがコピーされるとき、新しいファイルやディレクトリが存在しなければ、作成されます。その新しいファイルやディレクトリのコンテキストは、(オプションが元のコンテキストを保護されるために使用されて

[32] http://www.redhat.com/docs/en-US/Red_Hat_Enterprise_Linux/5.2/html/Deployment_Guide/s1-nfs-client-config.html

いなければ、) 元のファイルやディレクトリのコンテキストではなく、標準のラベルづけルールに基づきます。たとえば、ユーザーのホームディレクトリに作成されたファイルが user_home_t タイプのラベルを付けられます:

```
$ touch file1
$ ls -Z file1
-rw-rw-r--  user1 group1 unconfined_u:object_r:user_home_t:s0 file1
```

そのようなファイルが /etc/ のような他のディレクトリにコピーされると、新しいファイルが /etc/ ディレクトリに対する標準のラベルづけルールに基づいて作成されます。(追加のオプションなしで) ファイルをコピーすることにより、元のコンテキストが保護されない可能性があります:

```
$ ls -Z file1
-rw-rw-r--  user1 group1 unconfined_u:object_r:user_home_t:s0 file1
# cp file1 /etc/
$ ls -Z /etc/file1
-rw-r--r--  root root unconfined_u:object_r:etc_t:s0   /etc/file1
```

file1 が /etc/ にコピーされるとき、/etc/file1 が存在しなければ、/etc/file1 が新しいファイルとして作成されます。上の例に示されたとおり、/etc/file1 が標準のラベルづけルールに基づいて、etc_t タイプのラベルを付けられます。

ファイルが既存のファイルにコピーされるとき、ユーザーが --preserve=context のような元のファイルのコンテキストを保護する cp のオプションを指定しなければ、既存のファイルのコンテキストが保護されます。SELinux ポリシーにより、コンテキストがコピー中に保護されることを防がれます。

SELinux コンテキストを保護しないコピー

cp コマンドを用いてファイルをコピーするとき、オプションが指定されていなければ、タイプがターゲットの親ディレクトリから継承されます:

```
$ touch file1
$ ls -Z file1
-rw-rw-r--  user1 group1 unconfined_u:object_r:user_home_t:s0 file1
$ ls -dZ /var/www/html/
drwxr-xr-x  root root system_u:object_r:httpd_sys_content_t:s0 /var/www/html/
# cp file1 /var/www/html/
$ ls -Z /var/www/html/file1
-rw-r--r--  root root unconfined_u:object_r:httpd_sys_content_t:s0 /var/www/html/file1
```

この例では、file1 がユーザーのホームディレクトリに作成され、user_home_t タイプのラベルを付けられます。/var/www/html/ ディレクトリが、ls -dZ /var/www/html/ コマンドを用いて表示されるように、httpd_sys_content_t タイプのラベルを付けられています。file1 が /var/www/html/ にコピーされるとき、ls -Z /var/www/html/file1 コマンドを用いて表示されるように、httpd_sys_content_t タイプを継承します。

コピー中の SELinux コンテキストの保護

コピーするとき、コンテキストを保護するには、cp --preserve=context コマンドを使用します:

```
$ touch file1
$ ls -Z file1
-rw-rw-r--  user1 group1 unconfined_u:object_r:user_home_t:s0 file1
$ ls -dZ /var/www/html/
```

```
drwxr-xr-x  root root system_u:object_r:httpd_sys_content_t:s0 /var/www/html/
# cp --preserve=context file1 /var/www/html/
$ ls -Z /var/www/html/file1
-rw-r--r--  root root unconfined_u:object_r:user_home_t:s0 /var/www/html/file1
```

この例では、file1 がユーザーのホームディレクトリに作成され、user_home_t タイプのラベルが付けられます。/var/www/html/ ディレクトリが、ls -dZ /var/www/html/ コマンドで示されるように、httpd_sys_content_t タイプのラベルを付けられます。--preserve=context オプションを使用することにより、SELinux コンテキストがコピー操作中に保護されます。ls -Z /var/www/html/file1 を用いて示されるように、ファイルが /var/www/html/ にコピーされたとき、file1 の user_home_t タイプが保護されました。

コンテキストのコピーおよび変更

コピー先のコンテキストを変更するには cp -Z コマンドを使用します。以下の例は、ユーザーのホームディレクトリにおいて実行されました:

```
$ touch file1
$ cp -Z system_u:object_r:samba_share_t:s0 file1 file2
$ ls -Z file1 file2
-rw-rw-r--  user1 group1 unconfined_u:object_r:user_home_t:s0 file1
-rw-rw-r--  user1 group1 system_u:object_r:samba_share_t:s0 file2
$ rm file1 file2
```

この例は、コンテキストが -Z オプションで定義されています。-Z オプションがなければ、file2 が unconfined_u:object_r:user_home_t コンテキストのラベルを付けられます。

ファイルの既存ファイルへの上書きコピー

ファイルが既存のファイルに上書きコピーされるとき、(オプションがコンテキストを保護するために使用されていなければ、) 既存のファイルのコンテキストが保護されます。例:

```
# touch /etc/file1
# ls -Z /etc/file1
-rw-r--r--  root root unconfined_u:object_r:etc_t:s0   /etc/file1
# touch /tmp/file2
# ls -Z /tmp/file2
-rw-r--r--  root root unconfined_u:object_r:user_tmp_t:s0 /tmp/file2
# cp /tmp/file2 /etc/file1
# ls -Z /etc/file1
-rw-r--r--  root root unconfined_u:object_r:etc_t:s0   /etc/file1
```

この例では、2 つのファイルが作成されます: etc_t タイプのラベルを持つ /etc/file1 と user_tmp_t タイプのラベルを持つ /tmp/file2 です。cp /tmp/file2 /etc/file1 コマンドが file1 を file2 で上書きします。コピー後、ls -Z /etc/file1 コマンドにより、file1 が etc_t タイプのラベルを持つことが示されます。/etc/file1 を置き換えた /tmp/file2 の user_tmp_t タイプではありません。

重要

ファイルやディレクトリを移動するのではなく、コピーします。これにより、正しい SELinux コンテキストでラベルづけされることを保証する手助けになります。不適切な SELinux コンテキストにより、プロセスがそのようなファイルやディレクトリにアクセスすることを防がれる可能性があります。

9.4.10.2. ファイルとディレクトリの移動

ファイルやディレクトリが移動されるとき、現在の SELinux コンテキストが保持されます。多くの場合、これは移動先の位置に対して不適切です。以下の例は、ユーザーのホームディレクトリから、Apache HTTP Server により使用される /var/www/html/ に移動することについて説明します。ファイルが移動されるので、正しい SELinux コンテキストを継承しません:

1. ホームディレクトリに変更するには、引数なしで cd コマンドを実行します。ホームディレクトリにおいて、ファイルを作成するために、touch file1 コマンドを実行します。このファイルは user_home_t タイプのラベルを付けられています:

```
$ ls -Z file1
-rw-rw-r--  user1 group1 unconfined_u:object_r:user_home_t:s0 file1
```

2. /var/www/html/ ディレクトリの SELinux コンテキストを表示するには ls -dZ /var/www/html/ コマンドを実行します:

```
$ ls -dZ /var/www/html/
drwxr-xr-x  root root system_u:object_r:httpd_sys_content_t:s0 /var/www/html/
```

   /var/www/html/ ディレクトリが標準で、httpd_sys_content_t タイプのラベルを付けられます。/var/www/html/ ディレクトリの下に作成されたファイルとディレクトリがこのタイプを継承します。また、つまりこのタイプをラベルづけされます。

3. file1 を /var/www/html/ ディレクトリに移動するには、Linux root ユーザーとして mv file1 /var/www/html/ コマンドを実行します。このファイルが移動されるので、その現在の user_home_t タイプが残ります:

```
# mv file1 /var/www/html/
# ls -Z /var/www/html/file1
-rw-rw-r--  user1 group1 unconfined_u:object_r:user_home_t:s0 /var/www/html/file1
```

Apache HTTP Server が標準で、user_home_t タイプのラベルを持つファイルを読み込みできません。ウェブページを構成するすべてのファイルが、user_home_t タイプのラベルを持てば、または Apache HTTP Server が読み込めない他のタイプを持てば、Firefox やテキストウェブブラウザ経由でそれらにアクセスしようとしたとき、パーミッションが拒否されます。

重要

ファイルとディレクトリを mv コマンドを用いて移動することにより、Apache HTTP Server および Samba, のようなプロセスがそのようなファイルやディレクトリにアクセスすることを妨げるような、間違った SELinux コンテキストになる可能性があります。

9.4.10.3. 標準の SELinux コンテキストの確認

ファイルとディレクトリが正しい SELinux コンテキストを持つかどうかを確認するには、/usr/sbin/matchpathcon コマンドを使用します。matchpathcon(8) マニュアルページから: "matchpathcon がシステムポリシーを問い合わせ、ファイルパスと関連付けられた標準の SELinux コンテキストを出力します。"[33]。以

[33] matchpathcon(8) マニュアルページ。Fedora における *libselinux-utils* パッケージに同梱されます。Daniel Walsh 著。このバージョンにおける編集や変更は Murray McAllister により実行されました。

下の例は、/var/www/html/ にあるファイルが正しくラベルづけされていることを確認するために、/usr/sbin/matchpathcon コマンドを使用することを説明しています:

1. 3 つのファイル (file1, file2, および file3) を作成するには、Linux root ユーザーとして touch /var/www/html/file{1,2,3} コマンドを実行します。これらのファイルが /var/www/html/ ディレクトリから httpd_sys_content_t タイプを継承します:

```
# touch /var/www/html/file{1,2,3}
# ls -Z /var/www/html/
-rw-r--r--  root root unconfined_u:object_r:httpd_sys_content_t:s0 file1
-rw-r--r--  root root unconfined_u:object_r:httpd_sys_content_t:s0 file2
-rw-r--r--  root root unconfined_u:object_r:httpd_sys_content_t:s0 file3
```

2. file1 タイプを samba_share_t に変更するには、Linux root ユーザーとして chcon -t samba_share_t /var/www/html/file1 コマンドを実行します。注: Apache HTTP Server は samba_share_t タイプのラベルを持つファイルまたはディレクトリを読み込めません。

3. /usr/sbin/matchpathcon -V オプションが、現在の SELinux コンテキストを SELinux ポリシーにおける正しい標準のコンテキストと比較します。/var/www/html/ ディレクトリにあるすべてのファイルを確認するには、/usr/sbin/matchpathcon -V /var/www/html/* コマンドを実行します:

```
$ /usr/sbin/matchpathcon -V /var/www/html/*
/var/www/html/file1 has context unconfined_u:object_r:samba_share_t:s0, should be
 system_u:object_r:httpd_sys_content_t:s0
/var/www/html/file2 verified.
/var/www/html/file3 verified.
```

以下の /usr/sbin/matchpathcon コマンドの出力は、file1 が samba_share_t タイプのラベルを持つが、httpd_sys_content_t タイプのラベルを持つべきであることを示しています:

```
/var/www/html/file1 has context unconfined_u:object_r:samba_share_t:s0, should be
 system_u:object_r:httpd_sys_content_t:s0
```

ラベル問題を解決し、Apache HTTP Server が file1 にアクセスすることを許可するには、Linux root ユーザーとして /sbin/restorecon -v /var/www/html/file1 コマンドを実行します:

```
# /sbin/restorecon -v /var/www/html/file1
restorecon reset /var/www/html/file1 context unconfined_u:object_r:samba_share_t:s0-
>system_u:object_r:httpd_sys_content_t:s0
```

9.4.10.4. tar を用いたファイルのアーカイブ

tar は標準で拡張属性を保持しません。SELinux が拡張属性に保存されるので、コンテキストがファイルをアーカイブするときに失われます。コンテキストを保持したアーカイブを作成するには tar --selinux を使用します。tar アーカイブが拡張属性を持たないファイルを含めば、または拡張属性をシステム標準と一致させたければ、/sbin/restorecon からアーカイブを実行します:

```
$ tar -xvf archive.tar | /sbin/restorecon -f -
```

注: ディレクトリに依存して、/sbin/restorecon コマンドを実行するために、Linux root ユーザーになる必要があるかもしれません。

以下の例は SELinux コンテキストを保持したまま tar アーカイブを作成することを説明します:

1. 3 つのファイル (file1, file2, および file3) を作成するには、Linux root ユーザーとして touch /var/www/html/file{1,2,3} コマンドを実行します。これらのファイルが /var/www/html/ ディレクトリから httpd_sys_content_t タイプを継承します:

```
# touch /var/www/html/file{1,2,3}
# ls -Z /var/www/html/
-rw-r--r--  root root unconfined_u:object_r:httpd_sys_content_t:s0 file1
-rw-r--r--  root root unconfined_u:object_r:httpd_sys_content_t:s0 file2
-rw-r--r--  root root unconfined_u:object_r:httpd_sys_content_t:s0 file3
```

2. /var/www/html/ ディレクトリの中に変更するために cd /var/www/html/ コマンドを実行します。test.tar という名前の tar アーカイブを作成するには、このディレクトリにおいて一度、Linux root ユーザーとして tar --selinux -cf test.tar file{1,2,3} コマンドを実行します。

3. 新しいディレクトリを作成するには、Linux root ユーザーとして mkdir /test コマンドを実行します。そして、すべてのユーザーに /test/ ディレクトリの完全アクセス権を許可するために、chmod 777 /test/ コマンドを実行します。

4. test.tar ファイルを /test/ ディレクトリにコピーするには cp /var/www/html/test.tar /test/ コマンドを実行します。

5. /test/ ディレクトリの中に変更するには cd /test/ コマンドを実行します。このディレクトリに移動すると、tar アーカイブを展開するには tar -xvf test.tar コマンドを実行します:

6. SELinux コンテキストを表示するために ls -lZ /test/ コマンドを実行します。httpd_sys_content_t タイプが残されます。--selinux を使用しないときに起こる default_t に変更されていません。

```
$ ls -lZ /test/
-rw-r--r--  user1 group1 unconfined_u:object_r:httpd_sys_content_t:s0 file1
-rw-r--r--  user1 group1 unconfined_u:object_r:httpd_sys_content_t:s0 file2
-rw-r--r--  user1 group1 unconfined_u:object_r:httpd_sys_content_t:s0 file3
-rw-r--r--  user1 group1 unconfined_u:object_r:default_t:s0 test.tar
```

7. /test/ ディレクトリがもはや必要でなければ、そのディレクトリとその中にあるファイルを削除するために、Linux root ユーザーとして rm -ri /test/ コマンドを実行します。

すべての拡張属性を保持する --xattrs オプションのように、tar に関する詳細は tar(1) マニュアルページを参照してください。

9.4.10.5. star を用いたファイルのアーカイブ

star は標準で拡張属性を保持しません。SELinux が拡張属性に保存されるので、コンテキストがファイルをアーカイブするときに失われます。コンテキストを保持したアーカイブを作成するには star -xattr -H=exustar を使用します。*star* パッケージは標準でインストールされていません。star をインストールするには、Linux root ユーザーとして yum install star コマンドを実行します。

以下の例は、SELinux コンテキストを保持した Star アーカイブを作成することについて説明します。

1. 3 つのファイル (file1, file2, および file3) を作成するには、Linux root ユーザーとして touch /var/www/html/file{1,2,3} コマンドを実行します。これらのファイルが /var/www/html/ ディレクトリから httpd_sys_content_t タイプを継承します:

```
# touch /var/www/html/file{1,2,3}
# ls -Z /var/www/html/
```

```
-rw-r--r--  root root unconfined_u:object_r:httpd_sys_content_t:s0 file1
-rw-r--r--  root root unconfined_u:object_r:httpd_sys_content_t:s0 file2
-rw-r--r--  root root unconfined_u:object_r:httpd_sys_content_t:s0 file3
```

2. /var/www/html/ ディレクトリの中に変更するために cd /var/www/html/ コマンドを実行します。このディレクトリにおいて、test.star という名前の Star アーカイブを作成するために、Linux root ユーザーとして star -xattr -H=exustar -c -f=test.star file{1,2,3} コマンドを実行します:

```
# star -xattr -H=exustar -c -f=test.star file{1,2,3}
star: 1 blocks + 0 bytes (total of 10240 bytes = 10.00k).
```

3. 新しいディレクトリを作成するには、Linux root ユーザーとして mkdir /test コマンドを実行します。そして、すべてのユーザーに /test/ ディレクトリの完全アクセス権を許可するために、chmod 777 /test/ コマンドを実行します。

4. test.star ファイルを /test/ ディレクトリにコピーするには cp /var/www/html/test.star /test/ コマンドを実行します。

5. /test/ ディレクトリの中に変更するには cd /test/ コマンドを実行します。このディレクトリに移動すると、Star アーカイブを展開するには star -x -f=test.star コマンドを実行します:

```
$ star -x -f=test.star
star: 1 blocks + 0 bytes (total of 10240 bytes = 10.00k).
```

6. SELinux コンテキストを表示するために ls -lZ /test/ コマンドを実行します。httpd_sys_content_t タイプが残されます。--selinux を使用しないときに起こる default_t に変更されていません。

```
$ ls -lZ /test/
-rw-r--r--  user1 group1 unconfined_u:object_r:httpd_sys_content_t:s0 file1
-rw-r--r--  user1 group1 unconfined_u:object_r:httpd_sys_content_t:s0 file2
-rw-r--r--  user1 group1 unconfined_u:object_r:httpd_sys_content_t:s0 file3
-rw-r--r--  user1 group1 unconfined_u:object_r:default_t:s0 test.star
```

7. /test/ ディレクトリがもはや必要でなければ、そのディレクトリとその中にあるファイルを削除するために、Linux root ユーザーとして rm -ri /test/ コマンドを実行します。

8. star がもはや必要でなければ、パッケージを削除するために Linux ユーザーとして yum remove star コマンドを実行します。

star に関する詳細は star(1) マニュアルページを参照してください。

9.5. ユーザーの制限

数多くの制限された SELinux ユーザーが Fedora 19 において利用可能です。各 Linux ユーザーが SELinux ポリシー経由で SELinux ユーザーに対応付けられます。これは Linux ユーザーが SELinux ユーザーに置かれた制限を引き継げます。たとえばユーザーに依存しますが、次のことをできなくします: X Window System を実行する、ネットワークを使用する、(SELinux ポリシーがそれを許可していても) setuid アプリケーションを実行する、または su および sudo コマンドを実行する。制限されたユーザーに関するさらなる情報は*#Confined and Unconfined Users#*を参照してください。

9.5.1. Linux と SELinux のユーザーマッピング

Linux ユーザーと SELinux ユーザーの対応付けを表示するには、Linux root ユーザーとして id -Z コマンドを実行します:

```
# /usr/sbin/semanage login -l

Login Name                SELinux User              MLS/MCS Range

__default__               unconfined_u              s0-s0:c0.c1023
root                      unconfined_u              s0-s0:c0.c1023
system_u                  system_u                  s0-s0:c0.c1023
```

Fedora 19 では、Linux ユーザーが標準で SELinux __default__ に対応づけられます (これは次に SELinux unconfined_u ユーザーに対応づけられます)。Linux ユーザーが useradd コマンドで作成されるとき、オプションが指定されていないと、SELinux unconfined_u ユーザーに対応付けられます。以下は標準の対応づけを定義します:

```

__default__               unconfined_u              s0-s0:c0.c1023
```

9.5.2. 制限された新規 Linux ユーザー: useradd

Linux ユーザーは unconfined_t ドメインにおいて実行される SELinux unconfined_u ユーザーに対応づけられます。これは、unconfined_u に対応づけられた Linux ユーザーとしてログインしている間に、id -Z コマンドを実行することにより表示されます:

```

$ id -Z
unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
```

Linux ユーザーが unconfined_t ドメインにおいて実行しているとき、SELinux ポリシールールが適用されます。しかし、unconfined_t ドメインにおいて実行している Linux ユーザーがほとんどすべてにアクセスできるという、ポリシールールが存在します。制限されていないユーザーが SELinux ポリシー定義が unconfined_t ドメインから自身の制限されたドメインに遷移できるアプリケーションを実行していると、制限されていない Linux ユーザーがまだ制限されていないドメインの制限の対象になります。このセキュリティの利点は、Linux ユーザーが制限されず実行しているにもかかわらず、アプリケーションが制限されたままになることです。またそのため、アプリケーションにおける欠陥の侵入をポリシーにより制限できます。注: これはユーザーからシステムを保護しません。その代わり、ユーザーとシステムがアプリケーションにおける欠陥により引き起こされた損害から保護されます。

useradd を使用して Linux ユーザーを作成するとき、対応づけられる SELinux ユーザーを指定するために -Z オプションを使用します。以下の例は、Linux ユーザー useruuser を作成し、SELinux の user_u ユーザーを対応づけます。SELinux の user_u ユーザーに対応づけられたユーザーは user_t ドメインで実行します。このドメインでは、Linux ユーザーが SELinux ポリシーの許可なく setuid アプリケーション (passwd など) を実行できません。また、su や sudo のコマンドを用いて Linux root ユーザーになることを防ぐために、これらを実行できません。

1. SELinux user_u ユーザーに対応づけられる新規 Linux ユーザー (useruuser) を作成するには、Linux root ユーザーとして /usr/sbin/useradd -Z user_u useruuser を実行します。

2. Linux useruuser ユーザーと user_u の間の対応付けを表示するには、Linux root ユーザーとして semanage login -l コマンドを実行します:

   ```

   # /usr/sbin/semanage login -l
   ```

```
Login Name                SELinux User              MLS/MCS Range

__default__               unconfined_u              s0-s0:c0.c1023
root                      unconfined_u              s0-s0:c0.c1023
system_u                  system_u                  s0-s0:c0.c1023
useruuser                 user_u                    s0
```

3. Linux useruuser ユーザーにパスワードを設定するには、Linux root ユーザーとして passwd useruuser コマンドを実行します:

```
# passwd useruuser
Changing password for user useruuser.
New UNIX password: Enter a password
Retype new UNIX password: Enter the same password again
passwd: all authentication tokens updated successfully.
```

4. 現在のセッションをログアウトし、Linux useruuser ユーザーとしてログインします。ログインしたとき、pam_selinux が Linux ユーザーと SELinux ユーザー (この場合、user_u) を対応づけ、結果となる SELinux コンテキストをセットアップします。そして、Linux ユーザーのシェルがこのコンテキストで起動されます。Linux ユーザーのコンテキストを表示するには、id -Z コマンドを実行します:

```
[useruuser@localhost ~]$ id -Z
user_u:user_r:user_t:s0
```

5. Linux useruuser のセッションをログアウトし、お使いのアカウントで再びログインします。Linux useruuser ユーザーを使いたくなければ、そのホームディレクトリを残して、ユーザーを削除するために、Linux root ユーザーとして /usr/sbin/userdel -r useruuser コマンドを実行します。

9.5.3. 制限された既存の Linux ユーザー: semanage login

Linux ユーザーが SELinux unconfined_u ユーザーに対応づけられ (標準の動作)、対応づけられる SELinux ユーザーを変更したければ、semanage login コマンドを使用します。以下の例は、newuser という名前の新規 Linux ユーザーを作成し、Linux ユーザーを SELinux user_u ユーザーに対応づけます:

1. 新規 Linux ユーザー (newuser) を作成するには、Linux root ユーザーとして /usr/sbin/useradd newuser コマンドを実行します。このユーザーは標準の対応付けを使用するため、/usr/sbin/semanage login -l の出力に表示されません:

```
# /usr/sbin/semanage login -l

Login Name                SELinux User              MLS/MCS Range

__default__               unconfined_u              s0-s0:c0.c1023
root                      unconfined_u              s0-s0:c0.c1023
system_u                  system_u                  s0-s0:c0.c1023
```

2. Linux newuser ユーザーを SELinux user_u ユーザーと対応づけるには、Linux root ユーザーとして以下のコマンドを実行します:

   /usr/sbin/semanage login -a -s user_u newuser

   -a オプションが新規レコードを追加します。また、-s オプションが Linux ユーザーに対応づけるために SELinux ユーザーを指定します。最後の引数 newuser は、指定した SELinux ユーザーに対応づけたい Linux ユーザーです。

3. Linux newuser ユーザーと user_u との間の対応付けを表示するには、Linux root ユーザーとして semanage login -l を実行します:

```
# /usr/sbin/semanage login -l

Login Name                SELinux User              MLS/MCS Range

__default__               unconfined_u              s0-s0:c0.c1023
newuser                   user_u                    s0
root                      unconfined_u              s0-s0:c0.c1023
system_u                  system_u                  s0-s0:c0.c1023
```

4. Linux newuser ユーザーにパスワードを設定するには、Linux root ユーザーとして passwd newuser コマンドを実行します:

```
# passwd newuser
Changing password for user newuser.
New UNIX password: Enter a password
Retype new UNIX password: Enter the same password again
passwd: all authentication tokens updated successfully.
```

5. 現在のセッションをログアウトし、Linux newuser ユーザーとしてログインします。newuser の SELinux コンテキストを表示するには、id -Z コマンドを実行します:

```
[newuser@rlocalhost ~]$ id -Z
user_u:user_r:user_t:s0
```

6. Linux newuser のセッションをログアウトし、お使いのアカウントで再びログインします。Linux newuser ユーザーを使いたくなければ、そのホームディレクトリを残して、ユーザーを削除するために、Linux root ユーザーとして userdel -r newuser コマンドを実行します。また、Linux newuser ユーザーと user_u の対応付けが削除されます:

```
# /usr/sbin/userdel -r newuser
# /usr/sbin/semanage login -l

Login Name                SELinux User              MLS/MCS Range

__default__               unconfined_u              s0-s0:c0.c1023
root                      unconfined_u              s0-s0:c0.c1023
system_u                  system_u                  s0-s0:c0.c1023
```

9.5.4. 標準の対応付けの変更

Fedora 19 では、Linux ユーザーが標準で SELinux __default__ に対応づけられます (これは次に SELinux unconfined_u ユーザーに対応づけられます)。新しい Linux ユーザーを必要とし、Linux ユーザーが標準で制限される SELinux ユーザーにとくに対応づけられていなければ、標準の対応づけを semanage login コマンドで変更します。

たとえば、標準のマッピングを unconfined_u から user_u に変更するには、Linux root ユーザーとして以下のコマンドを実行します:

```
/usr/sbin/semanage login -m -S targeted -s "user_u" -r s0 __default__
```

__default__ ログインが user_u に対応付けられていることを確認するには、Linux root ユーザーとして semanage login -l コマンドを実行します:

```
# /usr/sbin/semanage login -l

Login Name                SELinux User              MLS/MCS Range

__default__               user_u                    s0
root                      unconfined_u              s0-s0:c0.c1023
system_u                  system_u                  s0-s0:c0.c1023
```

新規 Linux ユーザーが作成され、SELinux ユーザーが指定されていなければ、または既存の Linux ユーザーがログインし、semanage login -l の出力にある特定の項目に一致しなければ、そのユーザーは __default__ ログインとして user_u に対応付けられます。

標準の動作に戻すには、__default__ ログインを SELinux unconfined_u ユーザーに対応づけるために、Linux root ユーザーとして以下のコマンドを実行します:

```
/usr/sbin/semanage login -m -S targeted -s "unconfined_u" -r¥
s0-s0:c0.c1023 __default__
```

9.5.5. xguest: キオスクモード

xguest パッケージがキオスクユーザーアカウントを提供します。このアカウントは、図書館、銀行、空港、情報キオスク、およびコーヒーショップにいる人のように、行きかう人々が使用するセキュアなマシンのために使用されます。キオスクユーザーアカウントは非常に制限されます。基本的に、ユーザーがログインして、インターネット閲覧のために Firefox を使用することのみが許可されます。そのアカウントでログインしている間に行った、ファイルの作成や設定の変更などの変更はログアウト時に無くなります。

キオスクアカウントをセットアップするには:

1. *xguest* パッケージをインストールするには、Linux root ユーザーとして yum install xguest コマンドを実行します。必要に応じて依存するものがインストールされます。

2. キオスクアカウントがさまざまな人々により使用できるようにするために、アカウントがパスワードで保護されていません。そしてそのように、アカウントは SELinux がエンフォースモードで実行されている場合のみ保護できます。このアカウントでログインする前に、SELinux がエンフォースモードで実行されていることを確認するために、getenforce コマンドを使用します:

   ```
   $ /usr/sbin/getenforce
   Enforcing
   ```

   このようになっていなければ、エンフォースモードに変更する方法について*#SELinux ####*を参照してください。SELinux がパーミッシブモードまたは無効になっていると、このアカウントでログインできます。

3. このアカウントに GNOME Display Manager (GDM) 経由のみでログインできます。*xguest* パッケージがインストールされると、Guest アカウントが GDM に追加されます。ログインするには、Guest アカウントをクリックします:

9.5.6. ユーザー実行アプリケーション向けブーリアン

Linux ユーザーがホームディレクトリおよび /tmp/ で(ユーザーのパーミッションを継承する)アプリケーション(書き込みアクセス権を持ちます)を実行することを禁止することにより、セキュリティ侵害された、または悪意のあるアプリケーションがユーザーが所有するファイルを改ざんすることを防ぎます。Fedora 19 では標準で、guest_t および xguest_t ドメインのユーザーが、自身のホームディレクトリや /tmp/ にあるアプリケーションを実行できません。しかしながら標準で、user_t および staff_t のユーザーは実行できます。

ブーリアンがこの挙動を変更するために利用でき、setsebool コマンドで設定されます。setsebool コマンドは Linux root ユーザーとして実行する必要があります。setsebool -P コマンドは変更を永続化します。再起動後も変更を有効にしたくなければ、-P オプションを使用しないでください:

guest_t

guest_t ドメインにいる Linux ユーザーがホームディレクトリおよび /tmp/ にあるアプリケーションを実行###ようにするには:

```
/usr/sbin/setsebool -P allow_guest_exec_content on
```

xguest_t

xguest_t ドメインにいる Linux ユーザーがホームディレクトリおよび /tmp/ にあるアプリケーションを実行###ようにするには:

```
/usr/sbin/setsebool -P allow_xguest_exec_content on
```

user_t

user_t ドメインにいる Linux ユーザーがホームディレクトリおよび /tmp/ にあるアプリケーションを実行###ようにするには:

```
/usr/sbin/setsebool -P allow_user_exec_content off
```

staff_t

staff_t ドメインにいる Linux ユーザーがホームディレクトリおよび /tmp/ にあるアプリケーションを実行###ようにするには:

```
/usr/sbin/setsebool -P allow_staff_exec_content off
```

9.6. トラブルシューティング

以下の章は、SELinux がアクセスを拒否したときに何が起こるか、問題の主な 3 つの原因、正しいラベルに関する情報を見つけられるところ、SELinux の拒否の分析、および `audit2allow` を用いた個別ポリシーモジュールの作成について説明します。

9.6.1. アクセスが拒否されたときに何が起こるでしょうか

アクセスの許可または拒否のような、SELinux の判断がキャッシュされます。このキャッシュはアクセスベクターキャッシュ (AVC: Access Vector Cache) として知られています。SELinux がアクセスを拒否されたとき、拒否メッセージが記録されます。これらの拒否は "AVC 拒否" として知られています。実行されているデーモンにより、別の場所に記録されます:

デーモン	ログの位置
auditd on	`/var/log/audit/audit.log`
auditd off; rsyslogd on	`/var/log/messages`
setroubleshootd, rsyslogd, and auditd on	`/var/log/audit/audit.log`。読みやすい形式の拒否メッセージが `/var/log/messages` にも送信されます

X Window System を実行していて、*setroubleshoot* および *setroubleshoot-server* パッケージがインストールされ、かつ、`setroubleshootd` および `auditd` デーモンが実行していると、SELinux により拒否されたとき、警告が表示されます:

'表示' をクリックすることにより、なぜ SELinux がアクセスを拒否したのか詳細な分析、およびアクセスを許可するための解決案を表示します。もし X Window System を実行していなければ、アクセスが SELinux によりアクセスが拒否されたとき、よりわかりにくくなります。たとえば、ウェブサイトを閲覧しているユーザーが以下のようなメッセージを受け取るかもしれません:

```
Forbidden

You don't have permission to access file name on this server
```

これらの状況に対して、DAC ルール (標準的な Linux パーミッション) がアクセスを許可すると、`/var/log/messages` と `/var/log/audit/audit.log` をそれぞれ "SELinux is preventing" と "denied" エラーがないかを確認します。これは Linux root ユーザーとして以下のコマンドを実行することにより実行できます:

```
grep "SELinux is preventing" /var/log/messages

grep "denied" /var/log/audit/audit.log
```

9.6.2. 問題の 3 大原因

以下のセクションは問題の主な 3 つの原因について説明します。ラベルづけの問題、サービス向けのブーリアンとポートの設定、および SELinux ルールの発展です。

9.6.2.1. ラベルづけの問題

SELinux を実行しているシステムにおいて、すべてのプロセスとファイルはセキュリティ関連の情報を含むラベルを付けられています。この情報は SELinux コンテキストと呼ばれます。これらのラベルが間違っていれば、アクセスが拒否されます。アプリケーションが不適切にラベル付けされていると、プロセスが遷移したプロセスが正しいラベルを持たないかもしれません。これにより、SELinux がアクセスを拒否されます。また、プロセスが間違ったラベルを持つファイルを作成できます。

ラベル付け問題の一般的な原因は、非標準的なディレクトリがサービスのために使用されるときです。たとえば、ウェブサイトのために /var/www/html/ を使用する代わりに、管理者が /srv/myweb/ を使用したいときです。Fedora 19 では、/srv/ ディレクトリが var_t タイプのラベルが付けられています。/srv/ に作成されたファイルとディレクトリはこのタイプを継承します。また、新しく作成された (/myserver/ のような) トップディレクトリは default_t タイプのラベルを付けられます。SELinux により、Apache HTTP Server (httpd) がこれらのタイプにアクセスすることを防がれます。アクセスを許可するには、SELinux が /srv/myweb/ にあるファイルが httpd にアクセス可能であることを知らせる必要があります:

```
# /usr/sbin/semanage fcontext -a -t httpd_sys_content_t ¥
"/srv/myweb(/.*)?"
```

この semanage コマンドが、/srv/myweb/ ディレクトリ (および、それ以下のすべてのファイルとディレクトリ) のコンテキストに SELinux のファイルとコンテキストの設定[34] を追加します。semanage コマンドはコンテキストを変更しません。変更を適用するには Linux root ユーザーとして restorecon コマンドを実行します:

```
# /sbin/restorecon -R -v /srv/myweb
```

コンテキストにファイルとコンテキストの設定を追加することに関する詳細は*####### semanage fcontext #*を参照してください。

9.6.2.1.1. 適切なコンテキストとは何か?

matchpathcon コマンドがファイルのコンテキストを確認し、そのパスに対して標準のラベルと比較します。以下の例は、不適切なラベルを付けられたファイルを含むディレクトリに matchpathcon を使用することについて説明します:

```
$ /usr/sbin/matchpathcon -V /var/www/html/*
/var/www/html/index.html has context unconfined_u:object_r:user_home_t:s0, should be
 system_u:object_r:httpd_sys_content_t:s0
/var/www/html/page1.html has context unconfined_u:object_r:user_home_t:s0, should be
 system_u:object_r:httpd_sys_content_t:s0
```

この例では、index.html と page1.html ファイルが user_home_t タイプのラベルを付けられています。このタイプはユーザーのホームディレクトリにあるファイル向けに使用されます。ファイルをホームディレクトリから移動するために mv コマンドを使用することにより、ファイルに user_home_t タイプのラベルが付けられます。このタイプはホームディレクトリの外にあるべきではありません。そのようなファイルに適切なタイプを復元するには restorecon コマンドを使用します:

```
# /sbin/restorecon -v /var/www/html/index.html
```

[34] /etc/selinux/targeted/contexts/files/ にあるファイルがファイルとディレクトリのコンテキストを定義します。このディレクトリにあるファイルが、ファイルとディレクトリを標準のコンテキストに復元するために、restorecon および setfiles により読み込まれます。

```
restorecon reset /var/www/html/index.html context unconfined_u:object_r:user_home_t:s0-
>system_u:object_r:httpd_sys_content_t:s0
```

ディレクトリの下にあるすべてのファイルのコンテキストを復元するには、-R オプションを使用します:

```
# /sbin/restorecon -R -v /var/www/html/
restorecon reset /var/www/html/page1.html context unconfined_u:object_r:samba_share_t:s0-
>system_u:object_r:httpd_sys_content_t:s0
restorecon reset /var/www/html/index.html context unconfined_u:object_r:samba_share_t:s0-
>system_u:object_r:httpd_sys_content_t:s0
```

matchpathcon の詳細な例は#### *SELinux* ##########を参照してください。

9.6.2.2. どのように制限されたサービスが動作するか?

サービスはさまざまな方法で動作できます。これを満たすために、サービスがどのように動作しているのかを SELinux に教える必要があります。これはブーリアン経由で達成できます。これにより、SELinux ポリシー作成の知識なしで、SELinux ポリシーを実行時に変更できます。SELinux ポリシーの再読み込みまたは再コンパイルをすることなく、サービスが NFS ファイルシステムにアクセスを許可されるなどの、変更がこれにより許可されます。また、非標準のポート番号でサービスを実行することは、semanage コマンド経由でポリシー設定を更新する必要があります。

たとえば、Apache HTTP Server が MySQL と通信することを許可するには、httpd_can_network_connect_db ブーリアンをオンにします:

```
# /usr/sbin/setsebool -P httpd_can_network_connect_db on
```

特定のサービスのアクセスが拒否されたならば、アクセスを許可するためのすべてのブーリアンが利用可能かどうかを確認するために、getsebool および grep コマンドを使用します。たとえば、FTP 関連のブーリアンを探すには getsebool -a | grep ftp コマンドを使用します:

```
$ /usr/sbin/getsebool -a | grep ftp
allow_ftpd_anon_write --> off
allow_ftpd_full_access --> off
allow_ftpd_use_cifs --> off
allow_ftpd_use_nfs --> off
ftp_home_dir --> off
httpd_enable_ftp_server --> off
tftp_anon_write --> off
```

ブーリアンおよびそれがオンかオフかの一覧は、/usr/sbin/getsebool -a コマンドを実行します。ブーリアン、その説明およびオンかオフかの一覧は、Linux root ユーザーとして /usr/sbin/semanage boolean -l を実行します。ブーリアンの一覧表示と設定に関する詳細は########を参照してください。

ポート番号

ポリシー設定に依存して、サービスは特定のポート番号のみにおいて実行することが許可されるかもしれません。ポリシーを変更することなくポート番号の変更することにより、サービスが開始できなくなるかもしれません。たとえば、http 関連のポートを一覧表示するには、root ユーザーとして semanage port -l | grep -w http を実行します:

```
# /usr/sbin/semanage port -l | grep http
http_cache_port_t              tcp      3128, 8080, 8118
```

```
http_cache_port_t              udp      3130
http_port_t                    tcp      80, 443, 488, 8008, 8009, 8443
pegasus_http_port_t            tcp      5988
pegasus_https_port_t           tcp      5989
```

http_port_t ポートタイプが、Apache HTTP Server がリッスンできるポートを定義します。この場合、TCP ポート 80, 443, 488, 8008, 8009, および 8443 です。管理者が httpd にポート 9876 をリッスンするよう httpd.conf を設定 (Listen 9876) したが、ポリシーがこれを反映するために更新されていないと、service httpd start コマンドが失敗します:

```
# /sbin/service httpd start
Starting httpd: (13)Permission denied: make_sock: could not bind to address [::]:9876
(13)Permission denied: make_sock: could not bind to address 0.0.0.0:9876
no listening sockets available, shutting down
Unable to open logs
                   [FAILED]
```

以下のような SELinux の拒否が /var/log/audit/audit.log に記録されます:

```
type=AVC msg=audit(1225948455.061:294): avc:  denied  { name_bind } for  pid=4997 comm="httpd" src=9876
 scontext=unconfined_u:system_r:httpd_t:s0 tcontext=system_u:object_r:port_t:s0 tclass=tcp_socket
```

httpd が http_port_t ポートタイプに一覧化されていないポートをリッスンすることを許可するには、ポートをポリシー設定 [35] に追加するために、semanage port コマンドを実行します:

```
# /usr/sbin/semanage port -a -t http_port_t -p tcp 9876
```

-a オプションが新規レコードを追加します。-t オプションがタイプを定義します。また、-p オプションがプロトコルを定義します。最後の引数が追加するポート番号です。

9.6.2.3. ルールの放出および壊れたアプリケーション

SELinux にアクセスを拒否されたことが原因で、アプリケーションが壊れているかもしれません。また、SELinux ルールが放出されます - SELinux が特定の方法で実行されているアプリケーションを認識しないかもしれません。アプリケーションが期待したとおりに動作していても、アクセスを拒否される可能性があります。たとえば、新しいバージョンの PostgreSQL がリリースされていると、現在のポリシーが以前に見られなかったアクションを実行する可能性があります。これにより、アクセスが許可されていても、アクションが拒否されることになります。

これらの状況に対して、アクセスが拒否された後、アクセスを許可する個別ポリシーモジュールを作成するために audit2allow を使用します。audit2allow の使い方に関する詳細は########: audit2allow#を参照してください。

9.6.3. 問題の修復

以下のセクションは、問題をトラブルシューティングする手助けになります。次のことを進めていきます: SELinux ルールの前に確認される Linux パーミッションの確認。SELinux がアクセスを拒否して、メッセージが記録されないことが起こり得る原因。ラベルとブーリアンの情報を含む、サービスのマニュアルページ。システム全体ではなく、あるプロセスがパーミッシブに実行することを許可するパーミッシブなドメイン。拒否メッセージを検索および表示する方法。拒否を分析する方法。audit2allow を用いた個別ポリシーモジュールの作成。

[35] semanage port -a コマンドが項目を /etc/selinux/targeted/modules/active/ports.local ファイルに追加します。注: このファイルは標準で Linux root ユーザーのみにより表示できます。

9.6.3.1. Linux パーミッション

アクセスが拒否されたとき、標準的な Linux パーミッションを確認します。*#SELinux ###*に記載されているように、多くのオペレーティングシステムがアクセスを制御するために任意アクセス制御 (DAC) を使用します。これは、ユーザーが所有するファイルのパーミッションを自身が制御できます。SELinux ポリシールールは DAC ルールの後に確認されます。まず DAC ルールがアクセスを拒否すると、SELinux ポリシールールが使用されません。

アクセスが拒否され、SELinux 拒否が記録されていなければ、標準的な Linux パーミッションを表示するために `ls -l` コマンドを使用します:

```
$ ls -l /var/www/html/index.html
-rw-r----- 1 root root 0 2009-05-07 11:06 index.html
```

この例では、`index.html` が root ユーザーとグループにより所有されています。root ユーザーが読み込み権と書き込み権を持ち (`-rw`)、root グループのメンバーが読み込み権を持ちます (`-r-`)。その他のユーザーはアクセス権を持ちません (`---`)。そのようなパーミッションは標準で、`httpd` がこのファイルを読み込むことを許可されません。この問題を解決するには、所有者とグループを変更するために `chown` コマンドを使用します。このコマンドは Linux root ユーザーとして実行する必要があります:

```
# chown apache:apache /var/www/html/index.html
```

これは標準の設定を仮定しています。つまり、`httpd` が Linux apache ユーザーとして動作しています。`httpd` を違うユーザーで実行していれば、`apache:apache` をそのユーザーに置き換えます。

9.6.3.2. 静かな拒否の起こりうる原因

特定の状況において、SELinux がアクセスを拒否したときに、AVC 拒否が記録されないことがあります。アプリケーションやシステムライブラリが、ときどきそのタスクを実行するために必要となる以上のアクセス権を調査します。アプリケーションによる害のない調査のために、監査ログを AVC 拒否で埋め尽くすことなく、最小権限を維持するために、ポリシーが `dontaudit` ルールを使用することにより、パーミッションを許可することなく AVC 拒否を表示しなくできます。これらのルールは標準的なポリシーにおいて共通です。`dontaudit` の良くない面は、SELinux がアクセスを拒否したにもかかわらず、拒否メッセージが記録されず、トラブルシューティングが難しくなることです。

すべての拒否を記録することを許可する、`dontaudit` ルールを一時的に無効化するには、Linux root ユーザーとして以下のコマンドを実行します:

```
/usr/sbin/semodule -DB
```

`-D` オプションが `dontaudit` ルールを無効化します。`-B` オプションがポリシーを再構築します。`semodule -DB` を実行した後、パーミッション問題に遭遇した問題を実行してみて、SELinux 拒否 — アプリケーションにとって適切なもの — が記録されるかどうかを確認します。いくつかは `dontaudit` ルール経由で無視および処理されるべきであるので、拒否を許可すべきかどうかを判断するには注意してください。疑わしければ、また基準を探していれば、*fedora-selinux-list*[36] のような SELinux 一覧にある、他の SELinux ユーザーや開発者に連絡を取ってみます。

ポリシーを再構築し、`dontaudit` ルールを有効化するには、Linux root ユーザーとして以下のコマンドを実行します:

```
/usr/sbin/semodule -B
```

[36] http://www.redhat.com/mailman/listinfo/fedora-selinux-list

これにより、ポリシーの元の状態に復元されます。dontaudit ルールの完全な一覧は、sesearch --dontaudit コマンドを実行します。-s ***domain*** オプションと grep を使用して、検索を絞っていきます。例:

```
$ sesearch --dontaudit -s smbd_t | grep squid
WARNING: This policy contained disabled aliases; they have been removed.
dontaudit smbd_t squid_port_t : tcp_socket name_bind ;
dontaudit smbd_t squid_port_t : udp_socket name_bind ;
```

拒否について分析することに関する詳細は ########### および *#sealert* ###### を参照してください。

9.6.3.3. サービスのマニュアルページ

サービスのマニュアルページが、与えられた状況に使用するためのファイルタイプ、およびサービスが持つアクセス権 (httpd が NFS ファイルシステムにアクセスするなど) を変更するためのブーリアンのように、価値のある情報を含みます。この情報は標準的なマニュアルページにあるかもしれません。または、selinux が前か後ろに付いたマニュアルページにあるかもしれません。

たとえば、httpd_selinux(8) マニュアルページが、与えられた情報に使用するファイルタイプ、およびスクリプトを許可する、ファイルを共有する、ユーザーのホームディレクトリの中にアクセスするなどのブーリアンに関する情報を持ちます。サービス向けの SELinux 情報を含む、他のマニュアルページは次のものがあります:

- Samba: samba_selinux(8) マニュアルページが、Samba 経由でエクスポートされるファイルとディレクトリが samba_share_t タイプのラベルを付けられている必要があるものを説明しています。また、samba_share_t タイプ以外のラベルを持つファイルを Samba 経由でエクスポートすることを許可するためのブーリアンについて説明しています。

- NFS: nfs_selinux(8) マニュアルページが、ファイルシステムが標準で NFS 経由でエクスポートできないことを説明しています。また、ファイルシステムのエクスポートを許可するために、nfs_export_all_ro や nfs_export_all_rw のようなブーリアンをオンにする必要があることについて説明しています。

- Berkeley Internet Name Domain (BIND): named(8) マニュアルページが、与えられた状況のために使用するファイルのタイプについて説明しています (Red Hat SELinux BIND Security Profile セクション参照)。named_selinux(8) マニュアルページが、named が標準でマスターゾーンファイルに書き込みできないこと、そして、そのようなアクセスを許可するために、named_write_master_zones ブーリアンをオンにする必要があることを説明しています。

マニュアルページにある情報が、SELinux がアクセスを拒否する手助けになる、正しいファイルタイプとブーリアンを設定する役に立ちます。

9.6.3.4. パーミッシブドメイン

SELinux がパーミッシブモードで動作しているとき、SELinux がアクセスを拒否しません。しかし、エンフォースモードで実行していれば、拒否されたアクションに対して拒否メッセージが記録されます。以前は、単一のドメインをパーミッシブにできませんでした (補足: プロセスがドメインで動作します)。特定の状況において、問題をトラブルシュートするために、システム全体をパーミッシブにすることになります。

Fedora 19 はパーミッシブドメインを含みます。これは、管理者が、システム全体をパーミッシブにする代わりに、単一のプロセス (ドメイン) をパーミッシブに動作できます。SELinux がまだパーミッシブドメインのために実行されます。しかしながら、SELinux がアクセスを拒否する状況に対して、カーネルによりアクセスを許可され、AVC 拒否を記録されます。パーミッシブドメインが Fedora において利用可能です。

Red Hat Enterprise Linux 4 および 5 では、アプリケーションが制限されたドメインに遷移することを防ぐために、***domain***_disable_trans ブーリアンが利用可能です。そのため、プロセスが initrc_t のような制限されないドメインで動作します。そのようなブーリアンをオンにすることにより、主な問題が引き起こされます。たとえば、httpd_disable_trans ブーリアンがオンならば:

- httpd が制限されない initrc_t ドメインで動作します。initrc_t ドメインで動作しているプロセスにより作成されたファイルが、httpd_t ドメインで動作しているプロセスにより作成されたファイルに適用されたものと同じラベル付けルールを持たないかもしれません。これにより、プロセスが潜在的に誤ったラベルを付けられたファイルを作成されます。これは後ほどアクセス問題を引き起こします。
- httpd_t ドメインと通信することを許可される、制限されたドメインが initrc_t ドメインと通信できません。これにより、さらなるエラーを引き起こす可能性があります。

代替がありませんでしたが、***domain***_disable_trans ブーリアンが Fedora から削除されました。パーミッシブドメインが上の問題を解決します: 遷移ルールが適用され、ファイルが正しいラベルで作成されます。

パーミッシブドメインは次のために使用できます:

- システム全体をパーミッシブにすることによりシステム全体を危険にさらすよりは、単一プロセス (ドメイン) をパーミッシブに実行することにより、問題をトラブルシューティングします。
- 新しいアプリケーション向けのポリシーを作成します。これまでは、最小ポリシーを作成することが推奨されていました。そして、マシン全体をパーミッシブモードにします。それにより、アプリケーションを実行できますが、SELinux 拒否が記録されます。audit2allow がポリシーの作成に役立てるために使用できます。これはシステム全体を危険にさらします。パーミッシブドメインを用いると、システム全体を危険にさらすことなく、新しいポリシーにあるドメインのみがパーミッシブにできます。

9.6.3.4.1. ドメインのパーミッシブ化

ドメインをパーミッシブにするには、semanage permissive -a ***domain*** コマンドを実行します。*domain* はパーミッシブにしたいドメインです。たとえば、httpd_t ドメイン (Apache HTTP Server が動作するドメイン) をパーミッシブにするには、Linux root ユーザーとして以下のコマンドを実行します:

```
/usr/sbin/semanage permissive -a httpd_t
```

パーミッシブになっているドメインの一覧を表示するには、Linux root ユーザーとして semodule -l | grep permissive コマンドを実行します。たとえば:

```
# /usr/sbin/semodule -l | grep permissive
permissive_httpd_t      1.0
```

もはやドメインをパーミッシブにしたくなければ、Linux root ユーザーとして semanage permissive -d ***domain*** コマンドを実行します。例:

```
/usr/sbin/semanage permissive -d httpd_t
```

9.6.3.4.2. パーミッシブドメインに対する拒否

SYSCALL メッセージはパーミッシブ向けと異なります。以下は Apache HTTP Server からの AVC 拒否 (および、関連するシステムコール) の例です:

```
type=AVC msg=audit(1226882736.442:86): avc:  denied  { getattr } for  pid=2427 comm="httpd"
 path="/var/www/html/file1" dev=dm-0 ino=284133 scontext=unconfined_u:system_r:httpd_t:s0
 tcontext=unconfined_u:object_r:samba_share_t:s0 tclass=file

type=SYSCALL msg=audit(1226882736.442:86): arch=40000003 syscall=196 success=no exit=-13 a0=b9a1e198
 a1=bfc2921c a2=54dff4 a3=2008171 items=0 ppid=2425 pid=2427 auid=502 uid=48 gid=48 euid=48
 suid=48 fsuid=48 egid=48 sgid=48 fsgid=48 tty=(none) ses=4 comm="httpd" exe="/usr/sbin/httpd"
 subj=unconfined_u:system_r:httpd_t:s0 key=(null)
```

httpd_t ドメインは標準でパーミッシブでありません。アクションが拒否されたときと同じように、SYSCALL メッセージが success=no を含みます。以下は、semanage permissive -a httpd_t コマンドが httpd_t ドメインをパーミッシブにするために実行されること以外は、同じような状況に対する AVC 拒否の例です。

```
type=AVC msg=audit(1226882925.714:136): avc:  denied  { read } for  pid=2512 comm="httpd" name="file1"
 dev=dm-0 ino=284133 scontext=unconfined_u:system_r:httpd_t:s0 tcontext=unconfined_u:object_r:samba_share_t:s0
 tclass=file

type=SYSCALL msg=audit(1226882925.714:136): arch=40000003 syscall=5 success=yes exit=11 a0=b962a1e8 a1=8000
 a2=0 a3=8000 items=0 ppid=2511 pid=2512 auid=502 uid=48 gid=48 euid=48 suid=48 fsuid=48 egid=48 sgid=48
 fsgid=48 tty=(none) ses=4 comm="httpd" exe="/usr/sbin/httpd" subj=unconfined_u:system_r:httpd_t:s0 key=(null)
```

この場合、AVC 拒否が記録されましたが、SYSCALL メッセージに success=yes と示されるとおり、アクセスが拒否されませんでした。

パーミッシブドメインに関する詳細は Dan Walsh の*"Permissive Domains"*[37]ブログエントリーを参照してください。

9.6.3.5. 拒否の検索および表示

このセクションは、*setroubleshoot*, *setroubleshoot-server*, *dbus* および *audit* パッケージがインストールされ、auditd, rsyslogd, および setroubleshootd デーモンが実行中であることを仮定しています。これらのデーモンを開始することの詳細は##########を参照してください。ausearch, aureport, および sealert のような、多くのツールが SELinux 拒否を検索および表示するために利用可能です。

ausearch

audit パッケージが ausearch を提供します。ausearch(8) マニュアルページから: "ausearch は異なる検索基準に基づいたイベントに対して、監査デーモンのログを問い合わせできるツールです。"[38]。ausearch ツールが /var/log/audit/audit.log にアクセスします。そのように Linux root ユーザーとして実行する必要があります:

検索対象	コマンド
すべての拒否	/sbin/ausearch -m avc
本日の拒否	/sbin/ausearch -m avc -ts today
最近 10 分の拒否	/sbin/ausearch -m avc -ts recent

特定のサービスに対する SELinux 拒否を検索するには、-c ***comm-name*** オプションを使用します。ここで、*comm-name* " は実行可能バイナリの名前です"[39]。たとえば、Apache HTTP Server 向けの httpd、Samba 向けの smbd です:

```
/sbin/ausearch -m avc -c httpd

/sbin/ausearch -m avc -c smbd
```

さらなる ausearch オプションに関する ausearch(8) マニュアルページを参照してください。

[37] http://danwalsh.livejournal.com/24537.html
[38] ausearch(8) マニュアルページから、Fedora 19 で *audit* パッケージとして同梱されています。
[39] ausearch(8) マニュアルページから、Fedora 19 に *audit* パッケージとして同梱されます。

aureport

audit パッケージが aureport を提供します。aureport(8) マニュアルページから: "aureport は監査システムログの概要レポートを作成するツールです"[40]。aureport ツールが /var/log/audit/audit.log にアクセスします。そのように、Linux root ユーザーとして実行する必要があります。SELinux 拒否の一覧およびそれぞれの発生回数を表示するには、aureport -a コマンドを実行します。以下は 2 つの拒否を含む出力です:

```
# /sbin/aureport -a

AVC Report
============================================================
# date time comm subj syscall class permission obj event
============================================================
1. 05/01/2009 21:41:39 httpd unconfined_u:system_r:httpd_t:s0 195 file getattr
 system_u:object_r:samba_share_t:s0 denied 2
2. 05/03/2009 22:00:25 vsftpd unconfined_u:system_r:ftpd_t:s0 5 file read unconfined_u:object_r:cifs_t:s0
 denied 4
```

さらなる aureport オプションに関する aureport(8) マニュアルページを参照してください。

sealert

setroubleshoot-server パッケージが sealert を提供します。ここで、*setroubleshoot-server* により変換された拒否メッセージを読み込みます。拒否は /var/log/messages で見られるように ID を割り当てられます。以下は messages にある拒否の例です:

```
setroubleshoot: SELinux is preventing httpd (httpd_t) "getattr" to /var/www/html/file1 (samba_share_t). For
 complete SELinux messages. run sealert -l 84e0b04d-d0ad-4347-8317-22e74f6cd020
```

この例では、拒否 ID が 84e0b04d-d0ad-4347-8317-22e74f6cd020 です。-l オプションが引数として ID をとります。sealert -l 84e0b04d-d0ad-4347-8317-22e74f6cd020 コマンドを実行することにより、SELinux がアクセスを拒否した詳細な理由、およびアクセスを許可するために可能性のある解決方法が示されます。

X Window System を実行していて、*setroubleshoot* および *setroubleshoot-server* パッケージがインストールされていて、setroubleshootd, dbus および auditd デーモンが実行中であれば、SELinux によりアクセスが拒否されたときに、警告が表示されます。'表示 (Show)' をクリックすることにより、sealert GUI を起動し、HTML 出力に拒否を表示します:

[40] aureport(8) マニュアルページから、Fedora 19 で *audit* パッケージとして同梱されます。

- sealert GUI を起動するには sealert -b コマンドを実行します。
- すべての拒否の詳細な分析を表示するには sealert -l ¥* コマンドを実行します。
- sealert GUI で表示されるように、HTML バージョンの sealert 分析を作成するには、Linux root ユーザーとして sealert -a /var/log/audit/audit.log -H > audit.html コマンドを実行します。

sealert の詳細はsealert(8)マニュアルページを参照してください。

9.6.3.6. 生の監査メッセージ

生の監査メッセージが /var/log/audit/audit.log に記録されます。以下は、Apache HTTP Server (httpd_t ドメインで実行中) が (samba_share_t タイプのラベルを持つ) /var/www/html/file1 ファイルにアクセスを試行したときに発生する、AVC 拒否 (および、関連するシステムコール) の例です:

```
type=AVC msg=audit(1226874073.147:96): avc:  denied  { getattr } for  pid=2465 comm="httpd"
 path="/var/www/html/file1" dev=dm-0 ino=284133 scontext=unconfined_u:system_r:httpd_t:s0
 tcontext=unconfined_u:object_r:samba_share_t:s0 tclass=file

type=SYSCALL msg=audit(1226874073.147:96): arch=40000003 syscall=196 success=no exit=-13 a0=b98df198
 a1=bfec85dc a2=54dff4 a3=2008171 items=0 ppid=2463 pid=2465 auid=502 uid=48 gid=48 euid=48
 suid=48 fsuid=48 egid=48 sgid=48 fsgid=48 tty=(none) ses=6 comm="httpd" exe="/usr/sbin/httpd"
 subj=unconfined_u:system_r:httpd_t:s0 key=(null)
```

{ getattr }
: 括弧の中にある項目は、拒否されたパーミッションを意味します。getattr は、対象ファイルのステータス情報を読み込もうとするソースプロセスを意味します。これはファイルを読み込む前に発生します。間違ったラベルを持つファイルにアクセスしたため、このアクションが拒否されます。一般的に見られるパーミッションは、getattr, read, および write が含まれます。

comm="*httpd*"
: プロセスを起動する実行可能ファイルです。実行可能ファイルの完全パスがシステムコール (SYSCALL) メッセージの exe= セクションにあります。この場合、exe="/usr/sbin/httpd" です。

path="*/var/www/html/file1*"
プロセスがアクセスを試行したオブジェクト (ターゲット) のパスです。

scontext="*unconfined_u:system_r:httpd_t:s0*"
拒否されたアクションを試行したプロセスの SELinux コンテキストです。この場合、httpd_t ドメインで実行されている Apache HTTP Server の SELinux コンテキストです。

tcontext="*unconfined_u:object_r:samba_share_t:s0*"
プロセスがアクセスを試行したオブジェクト (ターゲット) の SELinux コンテキストです。この場合、file1 の SELinux コンテキストです。注: samba_share_t タイプは httpd_t ドメインで実行中のプロセスにアクセスできません。

特定の状況において、tcontext が scontext と一致するかもしれません。たとえば、プロセスが実行中のプロセスの特徴 (ユーザー ID など) を変更するシステムサービスを実行しようとするときです。また、tcontext が scontext と一致する可能性があります。これは、プロセスが通常の制限で許可された以上に (メモリのような) リソースを使用しようとするとき、そのプロセスがこれらの制限を超えることを許可されるかどうかを確認するためにセキュリティチェックされます。

システムコール (SYSCALL) メッセージから、2 つの項目に注目します:

- success=***no***: 拒否 (AVC) が強制されたかどうかを意味します。success=no はシステムコールが成功しなかったことを意味します (SELinux がアクセスを拒否しました)。success=yes はシステムコールが成功したことを意味します - これは initrc_t や kernel_t のようなパーミッシブドメインに対して見られます。
- exe="***/usr/sbin/httpd***": プロセスを起動した実行可能ファイルの完全パスです。この場合、exe="/usr/sbin/httpd" です。

正しくないファイルタイプが SELinux がアクセスを拒否する一般的な原因です。トラブルシューティングを開始するには、ソースコンテキスト (scontext) とターゲットコンテキスト (tcontext) を比較します。プロセス (scontext) がそのようなオブジェクト (tcontext) にアクセスすべきでしょうか? たとえば、Apache HTTP Server (httpd_t) は、設定を変更されていない限り、httpd_sys_content_t, public_content_t などのような、httpd_selinux(8) において指定されているタイプのみアクセスすべきです。

9.6.3.7. sealert メッセージ

拒否が /var/log/messages に見られる ID を割り当てられます。以下は、Apache HTTP Server (httpd_t ドメインで実行中) が (samba_share_t タイプのラベルを持つ) /var/www/html/file1 ファイルにアクセスを試行したときに発生する、(messages に記録される) AVC 拒否の例です:

```
hostname setroubleshoot: SELinux is preventing httpd (httpd_t) "getattr" to /var/www/html/file1
 (samba_share_t). For complete SELinux messages. run sealert -l 84e0b04d-d0ad-4347-8317-22e74f6cd020
```

推奨されるように、完全なメッセージを表示するには sealert -l 84e0b04d-d0ad-4347-8317-22e74f6cd020 コマンドを実行します。このコマンドはローカルマシンにおいてのみ機能します。sealert GUI と同じ情報が表示されます:

```
$ sealert -l 84e0b04d-d0ad-4347-8317-22e74f6cd020

Summary:

SELinux is preventing httpd (httpd_t) "getattr" to /var/www/html/file1
(samba_share_t).

Detailed Description:
```

```
SELinux denied access to /var/www/html/file1 requested by httpd.
/var/www/html/file1 has a context used for sharing by different program. If you
would like to share /var/www/html/file1 from httpd also, you need to change its
file context to public_content_t. If you did not intend to this access, this
could signal a intrusion attempt.

Allowing Access:

You can alter the file context by executing chcon -t public_content_t
'/var/www/html/file1'

Fix Command:

chcon -t public_content_t '/var/www/html/file1'

Additional Information:

Source Context                unconfined_u:system_r:httpd_t:s0
Target Context                unconfined_u:object_r:samba_share_t:s0
Target Objects                /var/www/html/file1 [ file ]
Source                        httpd
Source Path                   /usr/sbin/httpd
Port                          <Unknown>
Host                          hostname
Source RPM Packages           httpd-2.2.10-2
Target RPM Packages
Policy RPM                    selinux-policy-3.5.13-11.fc12
Selinux Enabled               True
Policy Type                   targeted
MLS Enabled                   True
Enforcing Mode                Enforcing
Plugin Name                   public_content
Host Name                     hostname
Platform                      Linux hostname 2.6.27.4-68.fc12.i686 #1 SMP Thu Oct
30 00:49:42 EDT 2008 i686 i686
Alert Count                   4
First Seen                    Wed Nov  5 18:53:05 2008
Last Seen                     Wed Nov  5 01:22:58 2008
Local ID                      84e0b04d-d0ad-4347-8317-22e74f6cd020
Line Numbers

Raw Audit Messages

node=hostname type=AVC msg=audit(1225812178.788:101): avc:  denied  { getattr } for  pid=2441
 comm="httpd" path="/var/www/html/file1" dev=dm-0 ino=284916 scontext=unconfined_u:system_r:httpd_t:s0
 tcontext=unconfined_u:object_r:samba_share_t:s0 tclass=file

node=hostname type=SYSCALL msg=audit(1225812178.788:101): arch=40000003 syscall=196 success=no
 exit=-13 a0=b8e97188 a1=bf87aaac a2=54dff4 a3=2008171 items=0 ppid=2439 pid=2441 auid=502 uid=48 gid=48
 euid=48 suid=48 fsuid=48 egid=48 sgid=48 fsgid=48 tty=(none) ses=3 comm="httpd" exe="/usr/sbin/httpd"
 subj=unconfined_u:system_r:httpd_t:s0 key=(null)
```

概要

拒否されたアクションの概要です。これは /var/log/messages にある拒否メッセージと同じです。この例では、httpd プロセスが、samba_share_t タイプのラベルが付いている、ファイル (file1) へのアクセスを拒否されました。

詳細な説明

より詳細な説明です。この例では、file1 が samba_share_t タイプのラベルを付けられています。このタイプは Samba 経由でエクスポートしたいファイルとディレクトリのために使用されます。この説明は、Apache HTTP Server や Samba によるアクセスを希望するならば、そのようなアクセスをできるタイプにタイプを変更することを推奨しています。

アクセスの許可
: アクセスを許可する方法の推奨です。これは、ファイルの再ラベル付け、ブーリアンのオン、またはローカルポリシーモジュールの作成が考えられます。この場合、推奨は Apache HTTP Server と Samba の両方にアクセス可能なタイプをファイルに付けることです。

修復コマンド
: アクセスを許可し、拒否を解決するために、推奨されるコマンドです。この例では、Apache HTTP Server と Samba にアクセス可能となる、public_content_t に file1 のタイプを変更するコマンドを与えられます。

追加情報
: ポリシーパッケージの名前とバージョン (selinux-policy-3.5.13-11.fc12) のような、バグ報告において有用な情報です。しかし、拒否が発生した理由を解決するための手助けにならないかもしれません。

生の監査メッセージ
: 拒否と関連付けられた /var/log/audit/audit.log から生の監査メッセージです。AVC 拒否にある各項目に関する詳細は##########を参照してください。

9.6.3.8. アクセスの許可: audit2allow

本番環境でこのセクションにある例を使用しないでください。audit2allow の使い方について説明するためだけに使用します。

audit2allow(1) マニュアルページから: "audit2allow - 拒否された操作のログから SELinux ポリシーの許可ルールを生成します"[41]。*#sealert ######*として拒否を分析した後、ラベルが変更されなければ、またはブーリアンがアクセスを許可しなければ、ローカルポリシーモジュールを作成するために audit2allow を使用します。アクセスが SELinux により拒否された後、audit2allow コマンドを実行することにより、前に拒否されたアクセスを許可するタイプエンフォースメントルールができます。

以下の例は、ポリシーモジュールを作成するために、audit2allow を使用することについて説明します:

1. 拒否および関連するシステムコールが /var/log/audit/audit.log に記録されます:

```
type=AVC msg=audit(1226270358.848:238): avc:  denied  { write } for  pid=13349
 comm="certwatch" name="cache" dev=dm-0 ino=218171 scontext=system_u:system_r:certwatch_t:s0
 tcontext=system_u:object_r:var_t:s0 tclass=dir

type=SYSCALL msg=audit(1226270358.848:238): arch=40000003 syscall=39 success=no exit=-13 a0=39a2bf
 a1=3ff a2=3a0354 a3=94703c8 items=0 ppid=13344 pid=13349 auid=4294967295 uid=0 gid=0 euid=0 suid=0
 fsuid=0 egid=0 sgid=0 fsgid=0 tty=(none) ses=4294967295 comm="certwatch" exe="/usr/bin/certwatch"
 subj=system_u:system_r:certwatch_t:s0 key=(null)
```

   この例では、certwatch (comm="certwatch") が var_t タイプ (tcontext=system_u:object_r:var_t:s0) のラベルを持つディレクトリに書き込みアクセス ({ write }) を拒否されました。*#sealert ######*にあるとおり、拒否を分析します。ラベルが変更されていなく、ブーリアンがアクセスを許可すると、ローカルポリシーモジュールを作成するために audit2allow を使用します。

2. 手順 1 における certwatch 拒否のように、記録された拒否を用いて、拒否された理由を人間が読みやすい説明を作成するために、audit2allow -w -a コマンドを実行します。-a オプションにより、すべての監査ログを読めるようにします。-w オプションにより、人間が読みやすい説明を作成します。audit2allow ツールが /var/log/audit/audit.log にアクセスします。これは Linux root ユーザーとして実行する必要があります:

[41] audit2allow(1) マニュアルページから、Fedora 19 に *policycoreutils* パッケージとして同梱されます。

```
# audit2allow -w -a
type=AVC msg=audit(1226270358.848:238): avc:  denied  { write } for  pid=13349
 comm="certwatch" name="cache" dev=dm-0 ino=218171 scontext=system_u:system_r:certwatch_t:s0
 tcontext=system_u:object_r:var_t:s0 tclass=dir
 Was caused by:
  Missing type enforcement (TE) allow rule.

 You can use audit2allow to generate a loadable module to allow this access.
```

示されたとおり、タイプエンフォースメントルールの欠落により、アクセスが拒否されました。

3. 拒否されたアクセスを許可するタイプエンフォースメントルールを表示するには audit2allow -a コマンドを実行します:

```
# audit2allow -a


#============= certwatch_t ==============
allow certwatch_t var_t:dir write;
```

重要

タイプエンフォースメントルールがないことが、通常 SELinux ポリシーにあるバグにより引き起こされます。*Red Hat Bugzilla*[42]に報告すべきです。Fedora に対して、Fedora 製品のバグを作成し、selinux-policy コンポーネントを選択します。そのバグ報告に audit2allow -w -a と audit2allow -a コマンドの出力を含みます。

4. audit2allow -a により表示されるルールを使用するには、個別モジュールを作成するために Linux root ユーザーとして audit2allow -a -M ***mycertwatch*** コマンドを実行します。-M オプションにより、現在の作業ディレクトリに、-M で指定した名前を持つタイプエンフォースメントファイル (.te) を作成します。

```
# audit2allow -a -M mycertwatch

****************** IMPORTANT ***********************
To make this policy package active, execute:

semodule -i mycertwatch.pp

# ls
mycertwatch.pp  mycertwatch.te
```

また、audit2allow がタイプエンフォースメントルールをポリシーパッケージ (.pp) にコンパイルします。モジュールをインストールするには、Linux root ユーザーとして /usr/sbin/semodule -i ***mycertwatch.pp*** コマンドを実行します。

[42] https://bugzilla.redhat.com/

重要

audit2allow で作成されたモジュールにより、必要とする以上のアクセス権が許可される可能性があります。audit2allow を用いて作成されたポリシーは、レビューのために *fedora-selinux-list*[43] のような SELinux リストに投稿することが推奨されます。ポリシーにバグがあると確信があれば、*Red Hat Bugzilla*[44] にバグを作成してください。

複数のプログラムから複数の拒否が発生するが、一つのプロセスのみに対して個別ポリシーを作成したければ、audit2allow の入力を絞るために grep コマンドを使用します。以下の例は、certwatch に関連した拒否のみを audit2allow に送るために、grep を使用することを説明します:

```
# grep certwatch /var/log/audit/audit.log | audit2allow -M mycertwatch2
******************** IMPORTANT ***********************
To make this policy package active, execute:

# /usr/sbin/semodule -i mycertwatch2.pp
```

ポリシーモジュールを作成するために audit2allow を使用することに関する詳細は Dan Walsh の *"Using audit2allow to build policy modules. Revisited."*[45] を参照してください。

9.7. 詳細情報

9.7.1. 貢献者

- *Geert Warrink*[46] (翻訳 - オランダ語)
- *Domingo Becker*[47] (翻訳 - スペイン語)
- *Daniel Cabrera*[48] (翻訳 - スペイン語)

9.7.2. 他のリソース

アメリカ国家安全保障局 (NSA: The National Security Agency)

NSA *Contributors to SELinux*[49] ページから:

NSA # National Information Assurance Research Laboratory (NIARL) ##########Linux ########## ##########Flask #########

[43] http://www.redhat.com/mailman/listinfo/fedora-selinux-list
[44] https://bugzilla.redhat.com/
[45] http://danwalsh.livejournal.com/24750.html
[46] http://fedoraproject.org/wiki/GeertWarrink
[47] http://fedoraproject.org/wiki/User:Beckerde
[48] http://fedoraproject.org/wiki/User:Logan
[49] http://www.nsa.gov/research/selinux/contrib.shtml

##NSA ##### Linux 2.6 ####### LSM #### SELinux # ###########NSA # X Window System (XACE/XSELinux) ### Xen (XSM/Flask) ################

- メインの SELinux ウェブサイト: *http://www.nsa.gov/research/selinux/index.shtml*。
- SELinux ドキュメント: *http://www.nsa.gov/research/selinux/docs.shtml*。
- SELinux バックグラウンド: *http://www.nsa.gov/research/selinux/background.shtml*。

Tresys Technology

Tresys Technology[50] は次に向けたアップストリームです:

- *SELinux userland libraries and tools*[51]。
- *SELinux Reference Policy*[52]。

SELinux ニュース

- ニュース: *http://selinuxnews.org/wp/*。
- Planet SELinux (ブログ): *http://selinuxnews.org/planet/*。

SELinux Project Wiki

- メインページ: *http://selinuxproject.org/page/Main_Page*。
- ドキュメント、メーリングリスト、ウェブサイトおよびツールへのリンクを含む、ユーザーリソース: *http://selinuxproject.org/page/User_Resources*。

Red Hat Enterprise Linux

- *Red Hat Enterprise Linux Deployment Guide*[53] に SELinux *References*[54] セクションがあります。SELinux チュートリアル、一般的な情報、および SELinux の技術へのリンクがあります。
- *Red Hat Enterprise Linux 4 SELinux Guide*[55]。

Fedora

- メインページ: *http://fedoraproject.org/wiki/SELinux*.
- トラブルシューティング: *http://fedoraproject.org/wiki/SELinux/Troubleshooting*.
- Fedora SELinux FAQ: *http://docs.fedoraproject.org/selinux-faq/*。
- SELinux Managing Confined Services Guide: *http://docs.fedoraproject.org/selinux-managing-confined-services-guide/*

[50] http://www.tresys.com/
[51] http://userspace.selinuxproject.org/trac/
[52] http://oss.tresys.com/projects/refpolicy
[53] http://www.redhat.com/docs/en-US/Red_Hat_Enterprise_Linux/5.2/html/Deployment_Guide/index.html
[54] http://www.redhat.com/docs/en-US/Red_Hat_Enterprise_Linux/5.2/html/Deployment_Guide/selg-chapter-0054.html
[55] http://www.redhat.com/docs/manuals/enterprise/RHEL-4-Manual/selinux-guide/index.html

The UnOfficial SELinux FAQ

http://www.crypt.gen.nz/selinux/faq.html

IRC

Freenode[56] において:

- #selinux
- #fedora-selinux
- #security

[56] http://freenode.net/

制限されたサービスの管理

10.1. Introduction

Security-Enhanced Linux (SELinux) refers to files, such as directories and devices, as objects. Processes, such as a user running a command or the Mozilla® Firefox® application, are referred to as subjects. Most operating systems use a Discretionary Access Control (DAC) system that controls how subjects interact with objects, and how subjects interact with each other. On operating systems using DAC, users control the permissions of files (objects) that they own. For example, on Linux® operating systems, users could make their home directories world-readable, inadvertently giving users and processes (subjects) access to potentially sensitive information.

DAC mechanisms are fundamentally inadequate for strong system security. DAC access decisions are only based on user identity and ownership, ignoring other security-relevant information such as the role of the user, the function and trustworthiness of the program, and the sensitivity and integrity of the data. Each user has complete discretion over their files, making it impossible to enforce a system-wide security policy. Furthermore, every program run by a user inherits all of the permissions granted to the user and is free to change access to the user's files, so no protection is provided against malicious software. Many system services and privileged programs must run with coarse-grained privileges that far exceed their requirements, so that a flaw in any one of these programs can be exploited to obtain complete system access.[1]

The following is an example of permissions used on Linux operating systems that do not run Security-Enhanced Linux (SELinux). The permissions in these examples may differ from your system. Use the `ls -l` command to view file permissions:

```
$ ls -l file1
-rwxrw-r-- 1 user1 group1 0 2010-02-28 07:12 file1
```

The first three permission bits, `rwx`, control the access the Linux `user1` user (in this case, the owner) has to `file1`. The next three permission bits, `rw-`, control the access the Linux `group1` group has to `file1`. The last three permission bits, `r--`, control the access everyone else has to `file1`, which includes all users and processes.

Security-Enhanced Linux (SELinux) adds Mandatory Access Control (MAC) to the Linux kernel, and is enabled by default in Fedora. A general purpose MAC architecture needs the ability to enforce an administratively-set security policy over all processes and files in the system, basing decisions on labels containing a variety of security-relevant information. When properly implemented, it enables a system to adequately defend itself and offers critical support for application security by protecting against the tampering with, and bypassing of, secured applications. MAC provides strong separation of applications that permits the safe execution of untrustworthy applications. Its ability to limit the privileges associated with executing processes limits the scope of potential damage that can result from the exploitation of vulnerabilities in applications and system services. MAC enables information to be protected from legitimate

[1] "Integrating Flexible Support for Security Policies into the Linux Operating System", by Peter Loscocco and Stephen Smalley. This paper was originally prepared for the National Security Agency and is, consequently, in the public domain. Refer to the *original paper* [http://www.nsa.gov/research/_files/selinux/papers/freenix01/index.shtml] for details and the document as it was first released. Any edits and changes were done by Murray McAllister.

users with limited authorization as well as from authorized users who have unwittingly executed malicious applications.[2]

The following is an example of the labels containing security-relevant information that are used on processes, Linux users, and files, on Linux operating systems that run SELinux. This information is called the SELinux context, and is viewed using the `ls -Z` command:

```
$ ls -Z file1
-rwxrw-r--  user1 group1 unconfined_u:object_r:user_home_t:s0      file1
```

In this example, SELinux provides a user (`unconfined_u`), a role (`object_r`), a type (`user_home_t`), and a level (`s0`). This information is used to make access control decisions. This example also displays the DAC rules, which are shown in the SELinux context via the `ls -Z` command. SELinux policy rules are checked after DAC rules. SELinux policy rules are not used if DAC rules deny access first.

10.2. Targeted policy

Targeted policy is the default SELinux policy used in Fedora. When using targeted policy, processes that are targeted run in a confined domain, and processes that are not targeted run in an unconfined domain. For example, by default, logged in users run in the `unconfined_t` domain, and system processes started by init run in the `initrc_t` domain - both of these domains are unconfined.

SELinux is based on the least level of access required for a service to run. Services can be run in a variety of ways; therefore, you must tell SELinux how you are running services. This can be achieved via Booleans that allow parts of SELinux policy to be changed at runtime, without any knowledge of SELinux policy writing. This allows changes, such as allowing services access to NFS file systems, without reloading or recompiling SELinux policy. Boolean configuration is discussed later.

Other changes, such as using non-default directories to store files for services, and changing services to run on non-default port numbers, require policy configuration to be updated via tools such as `semanage`. This is discussed later using detailed configuration examples.

10.2.1. Type Enforcement

Type Enforcement is the main permission control used in SELinux targeted policy. All files and processes are labeled with a type: types define a domain for processes and a type for files. SELinux policy rules define how types access each other, whether it be a domain accessing a type, or a domain accessing another domain. Access is only allowed if a specific SELinux policy rule exists that allows it.

10.2.2. Confined processes

Almost every service that listens on a network is confined in Fedora. Also, most processes that run as the root user and perform tasks for users, such as the passwd application, are confined. When a process is confined, it runs in its own domain, such as the `httpd` process running in the

[2] "Meeting Critical Security Objectives with Security-Enhanced Linux", by Peter Loscocco and Stephen Smalley. This paper was originally prepared for the National Security Agency and is, consequently, in the public domain. Refer to the *original paper* [http://www.nsa.gov/research/_files/selinux/papers/ottawa01/index.shtml] for details and the document as it was first released. Any edits and changes were done by Murray McAllister.

httpd_t domain. If a confined process is compromised by an attacker, depending on SELinux policy configuration, an attacker's access to resources and the possible damage they can do is limited.

The following example demonstrates how SELinux prevents the Apache HTTP Server (httpd) from reading files that are not correctly labeled, such as files intended for use by Samba. This is an example, and should not be used in production. It assumes that the *httpd*, *wget*, *setroubleshoot-server*, and *audit* packages are installed, that the SELinux targeted policy is used, and that SELinux is running in enforcing mode:

1. Run the sestatus command to confirm that SELinux is enabled, is running in enforcing mode, and that targeted policy is being used:

```
$ /usr/sbin/sestatus
SELinux status:                 enabled
SELinuxfs mount:                /selinux
Current mode:                   enforcing
Mode from config file:          enforcing
Policy version:                 24
Policy from config file:        targeted
```

   SELinux status: enabled is returned when SELinux is enabled. Current mode: enforcing is returned when SELinux is running in enforcing mode. Policy from config file: targeted is returned when the SELinux targeted policy is used.

2. As the root user, run the touch /var/www/html/testfile command to create a file.

3. Run the ls -Z /var/www/html/testfile command to view the SELinux context:

```
-rw-r--r--  root root unconfined_u:object_r:httpd_sys_content_t:s0 /var/www/html/testfile
```

   The testfile file is labeled with the SELinux unconfined_u user because a Linux user that is mapped to the unconfined_u SELinux user created the file. Role-Based Access Control (RBAC) is used for processes, not files. Roles do not have a meaning for files - the object_r role is a generic role used for files (on persistent storage and network file systems). Under the /proc/ directory, files related to processes may use the system_r role.[3] The httpd_sys_content_t type allows the httpd process to access this file.

4. As the root user, run the service httpd start command to start the httpd process. The output is as follows if httpd starts successfully:

```
# /sbin/service httpd start
Starting httpd:                                            [  OK  ]
```

5. Change into a directory where your Linux user has write access to, and run the wget http://localhost/testfile command. Unless there are changes to the default configuration, this command succeeds:

```
--2010-02-28 08:44:36--  http://localhost/testfile
Resolving localhost... 127.0.0.1
Connecting to localhost|127.0.0.1|:80... connected.
HTTP request sent, awaiting response... 200 OK
```

[3] When using other policies, such as MLS, other roles may be used, for example, secadm_r.

```
Length: 0 [text/plain]
Saving to: `testfile'

[ <=>                                 ] 0     --.-K/s   in 0s

2010-02-28 08:44:36 (0.00 B/s) - `testfile' saved [0/0]
```

6. The chcon command relabels files; however, such label changes do not survive when the file system is relabeled. For permanent changes that survive a file system relabel, use the semanage command, which is discussed later. As the root user, run the following command to change the type to a type used by Samba:

   chcon -t samba_share_t /var/www/html/testfile

   Run the ls -Z /var/www/html/testfile command to view the changes:

```
-rw-r--r--  root root unconfined_u:object_r:samba_share_t:s0 /var/www/html/testfile
```

7. Note: the current DAC permissions allow the httpd process access to testfile. Change into a directory where your Linux user has write access to, and run the wget http://localhost/testfile command. Unless there are changes to the default configuration, this command fails:

```
--2010-02-28 08:45:07--  http://localhost/testfile
Resolving localhost... 127.0.0.1
Connecting to localhost|127.0.0.1|:80... connected.
HTTP request sent, awaiting response... 403 Forbidden
2010-02-28 08:45:08 ERROR 403: Forbidden.
```

8. As the root user, run the rm -i /var/www/html/testfile command to remove testfile.

9. If you do not require httpd to be running, as the root user, run the service httpd stop command to stop httpd:

```
# /sbin/service httpd stop
Stopping httpd:                                            [  OK  ]
```

This example demonstrates the additional security added by SELinux. DAC rules allowed the httpd process access to testfile in step 7, but because the file was labeled with a type that the httpd process does not have access to, SELinux denied access. After step 7, an error similar to the following is logged to /var/log/messages:

```
Apr  6 23:00:54 localhost setroubleshoot: SELinux is preventing httpd (httpd_t) "getattr"
to /var/www/html/testfile (samba_share_t). For complete SELinux messages.
run sealert -l c05911d3-e680-4e42-8e36-fe2ab9f8e654
```

Previous log files may use a /var/log/messages.***YYYYMMDD*** format. When running syslog-ng, previous log files may use a /var/log/messages.***X*** format. If the setroubleshootd and auditd processes are running, errors similar to the following are logged to /var/log/audit/audit.log:

```
type=AVC msg=audit(1220706212.937:70): avc:  denied  { getattr } for  pid=1904 comm="httpd"
 path="/var/www/html/testfile" dev=sda5 ino=247576 scontext=unconfined_u:system_r:httpd_t:s0
 tcontext=unconfined_u:object_r:samba_share_t:s0  tclass=file

type=SYSCALL msg=audit(1220706212.937:70): arch=40000003 syscall=196 success=no exit=-13 a0=b9e21da0
 a1=bf9581dc a2=555ff4 a3=2008171 items=0 ppid=1902 pid=1904 auid=500 uid=48 gid=48 euid=48
```

```
suid=48 fsuid=48 egid=48 sgid=48 fsgid=48 tty=(none) ses=1 comm="httpd" exe="/usr/sbin/httpd"
subj=unconfined_u:system_r:httpd_t:s0 key=(null)
```

Also, an error similar to the following is logged to /var/log/httpd/error_log:

```
[Sat Apr 06 23:00:54 2009] [error] [client 127.0.0.1] (13)Permission denied: access to /testfile denied
```

10.2.3. Unconfined processes

Unconfined processes run in unconfined domains. For example, init programs run in the unconfined initrc_t domain, unconfined kernel processes run in the kernel_t domain, and unconfined Linux users run in the unconfined_t domain. For unconfined processes, SELinux policy rules are applied, but policy rules exist that allow processes running in unconfined domains almost all access. Processes running in unconfined domains fall back to using DAC rules exclusively. If an unconfined process is compromised, SELinux does not prevent an attacker from gaining access to system resources and data, but of course, DAC rules are still used. SELinux is a security enhancement on top of DAC rules - it does not replace them.

The following example demonstrates how the Apache HTTP Server (httpd) can access data intended for use by Samba, when running unconfined. Note: in Fedora, the httpd process runs in the confined httpd_t domain by default. This is an example, and should not be used in production. It assumes that the *httpd*, *wget*, *setroubleshoot-server*, and *audit* packages are installed, that the SELinux targeted policy is used, and that SELinux is running in enforcing mode:

1. Run the sestatus command to confirm that SELinux is enabled, is running in enforcing mode, and that targeted policy is being used:

   ```
   $ /usr/sbin/sestatus
   SELinux status:                 enabled
   SELinuxfs mount:                /selinux
   Current mode:                   enforcing
   Mode from config file:          enforcing
   Policy version:                 24
   Policy from config file:        targeted
   ```

   SELinux status: enabled is returned when SELinux is enabled. Current mode: enforcing is returned when SELinux is running in enforcing mode. Policy from config file: targeted is returned when the SELinux targeted policy is used.

2. As the root user, run the touch /var/www/html/test2file command to create a file.

3. Run the ls -Z /var/www/html/test2file command to view the SELinux context:

   ```
   -rw-r--r--  root root unconfined_u:object_r:httpd_sys_content_t:s0 /var/www/html/test2file
   ```

   test2file is labeled with the SELinux unconfined_u user because a Linux user that is mapped to the unconfined_u SELinux user created the file. RBAC is used for processes, not files. Roles do not have a meaning for files - the object_r role is a generic role used for files (on persistent storage and network file systems). Under the /proc/ directory, files related to processes may use the system_r role.[4] The httpd_sys_content_t type allows the httpd process to access this file.

[4] When using other policies, such as MLS, other roles may also be used, for example, secadm_r.

4. The chcon command relabels files; however, such label changes do not survive when the file system is relabeled. For permanent changes that survive a file system relabel, use the semanage command, which is discussed later. As the root user, run the following command to change the type to a type used by Samba:

   ```
   chcon -t samba_share_t /var/www/html/test2file
   ```

   Run the ls -Z /var/www/html/test2file command to view the changes:

   ```
   -rw-r--r--  root root unconfined_u:object_r:samba_share_t:s0 /var/www/html/test2file
   ```

5. Run the service httpd status command to confirm that the httpd process is not running:

   ```
   $ /sbin/service httpd status
   httpd is stopped
   ```

   If the output differs, run the service httpd stop command as the root user to stop the httpd process:

   ```
   # /sbin/service httpd stop
   Stopping httpd:                                            [  OK  ]
   ```

6. To make the httpd process run unconfined, run the following command as the root user to change the type of /usr/sbin/httpd, to a type that does not transition to a confined domain:

   ```
   chcon -t unconfined_exec_t /usr/sbin/httpd
   ```

7. Run the ls -Z /usr/sbin/httpd command to confirm that /usr/sbin/httpd is labeled with the unconfined_exec_t type:

   ```
   -rwxr-xr-x  root root system_u:object_r:unconfined_exec_t /usr/sbin/httpd
   ```

8. As the root user, run the service httpd start command to start the httpd process. The output is as follows if httpd starts successfully:

   ```
   # /sbin/service httpd start
   Starting httpd:                                            [  OK  ]
   ```

9. Run the ps -eZ | grep httpd command to view the httpd processes running in the unconfined_t domain:

   ```
   $ ps -eZ | grep httpd
   unconfined_u:system_r:unconfined_t 7721 ?      00:00:00 httpd
   unconfined_u:system_r:unconfined_t 7723 ?      00:00:00 httpd
   unconfined_u:system_r:unconfined_t 7724 ?      00:00:00 httpd
   unconfined_u:system_r:unconfined_t 7725 ?      00:00:00 httpd
   unconfined_u:system_r:unconfined_t 7726 ?      00:00:00 httpd
   unconfined_u:system_r:unconfined_t 7727 ?      00:00:00 httpd
   unconfined_u:system_r:unconfined_t 7728 ?      00:00:00 httpd
   unconfined_u:system_r:unconfined_t 7729 ?      00:00:00 httpd
   unconfined_u:system_r:unconfined_t 7730 ?      00:00:00 httpd
   ```

10. Change into a directory where your Linux user has write access to, and run the `wget http://localhost/test2file` command. Unless there are changes to the default configuration, this command succeeds:

```
--2008-09-07 01:41:10--  http://localhost/test2file
Resolving localhost... 127.0.0.1
Connecting to localhost|127.0.0.1|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 0 [text/plain]
Saving to: `test2file.1'

[ <=>                                 ]--.-K/s   in 0s

2008-09-07 01:41:10 (0.00 B/s) - `test2file.1' saved [0/0]
```

    Although the `httpd` process does not have access to files labeled with the `samba_share_t` type, `httpd` is running in the unconfined `unconfined_t` domain, and falls back to using DAC rules, and as such, the `wget` command succeeds. Had `httpd` been running in the confined `httpd_t` domain, the `wget` command would have failed.

11. The `restorecon` command restores the default SELinux context for files. As the root user, run the `restorecon -v /usr/sbin/httpd` command to restore the default SELinux context for `/usr/sbin/httpd`:

```
# /sbin/restorecon -v /usr/sbin/httpd
restorecon reset /usr/sbin/httpd context system_u:object_r:unconfined_notrans_exec_t:s0-
>system_u:object_r:httpd_exec_t:s0
```

    Run the `ls -Z /usr/sbin/httpd` command to confirm that `/usr/sbin/httpd` is labeled with the `httpd_exec_t` type:

```
$ ls -Z /usr/sbin/httpd
-rwxr-xr-x  root root system_u:object_r:httpd_exec_t   /usr/sbin/httpd
```

12. As the root user, run the `/sbin/service httpd restart` command to restart `httpd`. After restarting, run the `ps -eZ | grep httpd` to confirm that `httpd` is running in the confined `httpd_t` domain:

```
# /sbin/service httpd restart
Stopping httpd:                                            [  OK  ]
Starting httpd:                                            [  OK  ]
# ps -eZ | grep httpd
unconfined_u:system_r:httpd_t    8880 ?        00:00:00 httpd
unconfined_u:system_r:httpd_t    8882 ?        00:00:00 httpd
unconfined_u:system_r:httpd_t    8883 ?        00:00:00 httpd
unconfined_u:system_r:httpd_t    8884 ?        00:00:00 httpd
unconfined_u:system_r:httpd_t    8885 ?        00:00:00 httpd
unconfined_u:system_r:httpd_t    8886 ?        00:00:00 httpd
unconfined_u:system_r:httpd_t    8887 ?        00:00:00 httpd
unconfined_u:system_r:httpd_t    8888 ?        00:00:00 httpd
unconfined_u:system_r:httpd_t    8889 ?        00:00:00 httpd
```

13. As the root user, run the `rm -i /var/www/html/test2file` command to remove `test2file`.

14. If you do not require `httpd` to be running, as the root user, run the `service httpd stop` command to stop `httpd`:

```
# /sbin/service httpd stop
Stopping httpd:                                            [  OK  ]
```

The examples in these sections demonstrate how data can be protected from a compromised confined process (protected by SELinux), as well as how data is more accessible to an attacker from a compromised unconfined process (not protected by SELinux).

10.3. The Apache HTTP Server

From the *Apache HTTP Server Project*[5] page:

"The Apache HTTP Server Project is an effort to develop and maintain an open-source HTTP server for modern operating systems including UNIX and Windows NT. The goal of this project is to provide a secure, efficient and extensible server that provides HTTP services in sync with the current HTTP standards".[6]

In Fedora, the *httpd* package provides the Apache HTTP Server. Run `rpm -q httpd` to see if the *httpd* package is installed. If it is not installed and you want to use the Apache HTTP Server, run the following command as the root user to install it:

```
yum install httpd
```

10.3.1. The Apache HTTP Server and SELinux

When SELinux is enabled, the Apache HTTP Server (`httpd`) runs confined by default. Confined processes run in their own domains, and are separated from other confined processes. If a confined process is compromised by an attacker, depending on SELinux policy configuration, an attacker's access to resources and the possible damage they can do is limited. The following example demonstrates the `httpd` processes running in their own domain. This example assumes the *httpd* package is installed:

1. Run `getenforce` to confirm SELinux is running in enforcing mode:

   ```
   $ getenforce
   Enforcing
   ```

   The `getenforce` command returns `Enforcing` when SELinux is running in enforcing mode.

2. Run `service httpd start` as the root user to start `httpd`:

   ```
   # service httpd start
   Starting httpd:                                            [  OK  ]
   ```

3. Run `ps -eZ | grep httpd` to view the `httpd` processes:

[5] http://httpd.apache.org/

[6] From the "The Number One HTTP Server On The Internet" section of the Apache HTTP Server Project page: *http://httpd.apache.org/*. Copyright © 2010 The Apache Software Foundation. Accessed 1 March 2010.

```
$ ps -eZ | grep httpd
unconfined_u:system_r:httpd_t:s0 2850 ?        00:00:00 httpd
unconfined_u:system_r:httpd_t:s0 2852 ?        00:00:00 httpd
unconfined_u:system_r:httpd_t:s0 2853 ?        00:00:00 httpd
unconfined_u:system_r:httpd_t:s0 2854 ?        00:00:00 httpd
unconfined_u:system_r:httpd_t:s0 2855 ?        00:00:00 httpd
unconfined_u:system_r:httpd_t:s0 2856 ?        00:00:00 httpd
unconfined_u:system_r:httpd_t:s0 2857 ?        00:00:00 httpd
unconfined_u:system_r:httpd_t:s0 2858 ?        00:00:00 httpd
unconfined_u:system_r:httpd_t:s0 2859 ?        00:00:00 httpd
```

The SELinux context associated with the `httpd` processes is `unconfined_u:system_r:httpd_t:s0`. The second last part of the context, `httpd_t`, is the type. A type defines a domain for processes and a type for files. In this case, the `httpd` processes are running in the `httpd_t` domain.

SELinux policy defines how processes running in confined domains, such as `httpd_t`, interact with files, other processes, and the system in general. Files must be labeled correctly to allow `httpd` access to them. For example, `httpd` can read files labeled with the `httpd_sys_content_t` type, but can not write to them, even if Linux permissions allow write access. Booleans must be turned on to allow certain behavior, such as allowing scripts network access, allowing `httpd` access to NFS and CIFS file systems, and `httpd` being allowed to execute Common Gateway Interface (CGI) scripts.

When `/etc/httpd/conf/httpd.conf` is configured so `httpd` listens on a port other than TCP ports 80, 443, 488, 8008, 8009, or 8443, the `semanage port` command must be used to add the new port number to SELinux policy configuration. The following example demonstrates configuring `httpd` to listen on a port that is not defined in SELinux policy configuration for `httpd`, and, as a consequence, `httpd` failing to start. This example also demonstrates how to then configure the SELinux system to allow `httpd` to successfully listen on a non-standard port that is not already defined in the policy. This example assumes the *httpd* package is installed. Run each command in the example as the root user:

1. Run `service httpd status` to confirm `httpd` is not running:

   ```
   # service httpd status
   httpd is stopped
   ```

   If the output differs, run `service httpd stop` to stop the process:

   ```
   # service httpd stop
   Stopping httpd:                                        [  OK  ]
   ```

2. Run `semanage port -l | grep -w http_port_t` to view the ports SELinux allows `httpd` to listen on:

   ```
   # semanage port -l | grep -w http_port_t
   http_port_t                    tcp      80, 443, 488, 8008, 8009, 8443
   ```

3. Edit /etc/httpd/conf/httpd.conf as the root user. Configure the Listen option so it lists a port that is not configured in SELinux policy configuration for httpd. In this example, httpd is configured to listen on port 12345:

```
# Change this to Listen on specific IP addresses as shown below to
# prevent Apache from glomming onto all bound IP addresses (0.0.0.0)
#
#Listen 12.34.56.78:80
Listen 127.0.0.1:12345
```

4. Run service httpd start to start httpd:

```
# service httpd start
Starting httpd: (13)Permission denied: make_sock: could not bind to address 127.0.0.1:12345
no listening sockets available, shutting down
Unable to open logs       [FAILED]
```

   An SELinux denial similar to the following is logged to /var/log/messages:

```
setroubleshoot: SELinux is preventing the httpd (httpd_t) from binding to port 12345. For complete SELinux
 messages. run sealert -l f18bca99-db64-4c16-9719-1db89f0d8c77
```

5. For SELinux to allow httpd to listen on port 12345, as used in this example, the following command is required:

```
# semanage port -a -t http_port_t -p tcp 12345
```

6. Run service httpd start again to start httpd and have it listen on the new port:

```
# service httpd start
Starting httpd:        [  OK  ]
```

7. Now that SELinux has been configured to allow httpd to listen on a non-standard port (TCP 12345 in this example), httpd starts successfully on this port.

8. To prove that httpd is listening and communicating on TCP port 12345, open a telnet connection to the specified port and issue a HTTP GET command, as follows:

```
# telnet localhost 12345
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
GET / HTTP/1.0

HTTP/1.1 200 OK
Date: Tue, 31 Mar 2009 13:12:10 GMT
Server: Apache/2.2.11 (Fedora)
Accept-Ranges: bytes
Content-Length: 3918
Content-Type: text/html; charset=UTF-8
[...continues...]
```

10.3.2. Types

Type Enforcement is the main permission control used in SELinux targeted policy. All files and processes are labeled with a type: types define a domain for processes and a type for files. SELinux policy rules define how types access each other, whether it be a domain accessing a type, or a domain accessing another domain. Access is only allowed if a specific SELinux policy rule exists that allows it.

The following example creates a new file in the `/var/www/html/` directory, and shows the file inheriting the `httpd_sys_content_t` type from its parent directory (`/var/www/html/`):

1. Run `ls -dZ /var/www/html` to view the SELinux context of `/var/www/html/`:

```
$ ls -dZ /var/www/html
drwxr-xr-x  root root system_u:object_r:httpd_sys_content_t:s0 /var/www/html
```

   This shows `/var/www/html/` is labeled with the `httpd_sys_content_t` type.

2. Run `touch /var/www/html/file1` as the root user to create a new file.

3. Run `ls -Z /var/www/html/file1` to view the SELinux context:

```
$ ls -Z /var/www/html/file1
-rw-r--r--  root root unconfined_u:object_r:httpd_sys_content_t:s0 /var/www/html/file1
```

The `ls -Z` command shows `file1` labeled with the `httpd_sys_content_t` type. SELinux allows `httpd` to read files labeled with this type, but not write to them, even if Linux permissions allow write access. SELinux policy defines what types a process running in the `httpd_t` domain (where `httpd` runs) can read and write to. This helps prevent processes from accessing files intended for use by another process.

For example, `httpd` can access files labeled with the `httpd_sys_content_t` type (intended for the Apache HTTP Server), but by default, can not access files labeled with the `samba_share_t` type (intended for Samba). Also, files in user home directories are labeled with the `user_home_t` type: by default, this prevents `httpd` from reading or writing to files in user home directories.

The following types are used with `httpd`. Different types allow you to configure flexible access:

`httpd_sys_content_t`
: Use this type for static web content, such as `.html` files used by a static website. Files labeled with this type are accessible (read only) to `httpd` and scripts executed by `httpd`. By default, files and directories labeled with this type can not be written to or modified by `httpd` or other processes. Note: by default, files created in or copied into `/var/www/html/` are labeled with the `httpd_sys_content_t` type.

`httpd_sys_script_exec_t`
: Use this type for scripts you want `httpd` to execute. This type is commonly used for Common Gateway Interface (CGI) scripts in `/var/www/cgi-bin/`. By default, SELinux policy prevents `httpd` from executing CGI scripts. To allow this, label the scripts with the `httpd_sys_script_exec_t` type and turn the `httpd_enable_cgi` Boolean on. Scripts labeled with `httpd_sys_script_exec_t` run in the `httpd_sys_script_t` domain when executed by `httpd`. The `httpd_sys_script_t` domain has access to other system domains, such as `postgresql_t` and `mysqld_t`.

`httpd_sys_content_rw_t`
: Files labeled with this type can be written to by scripts labeled with the `httpd_sys_script_exec_t` type, but can not be modified by scripts labeled with any other type. You must use the `httpd_sys_content_rw_t` type to label files that will be read from and written to by scripts labeled with the `httpd_sys_script_exec_t` type.

`httpd_sys_content_ra_t`
: Files labeled with this type can be appended to by scripts labeled with the `httpd_sys_script_exec_t` type, but can not be modified by scripts labeled with any other type. You must use the `httpd_sys_content_ra_t` type to label files that will be read from and appended to by scripts labeled with the `httpd_sys_script_exec_t` type.

`httpd_unconfined_script_exec_t`
: Scripts labeled with this type run without SELinux protection. Only use this type for complex scripts, after exhausting all other options. It is better to use this type instead of turning SELinux protection off for `httpd`, or for the entire system.

Changing the SELinux Context

The type for files and directories can be changed with the `chcon` command. Changes made with `chcon` do not survive a file system relabel or the `restorecon` command. SELinux policy controls whether users are able to modify the SELinux context for any given file. The following example demonstrates creating a new directory and an `index.html` file for use by `httpd`, and labeling that file and directory to allow `httpd` access to them:

1. Run `mkdir -p /my/website` as the root user to create a top-level directory structure to store files to be used by `httpd`.

2. Files and directories that do not match a pattern in file-context configuration may be labeled with the `default_t` type. This type is inaccessible to confined services:

```
$ ls -dZ /my
drwxr-xr-x  root root unconfined_u:object_r:default_t:s0 /my
```

3. Run `chcon -R -t httpd_sys_content_t /my/` as the root user to change the type of the `/my/` directory and subdirectories, to a type accessible to `httpd`. Now, files created under `/my/website/` inherit the `httpd_sys_content_t` type, rather than the `default_t` type, and are therefore accessible to httpd:

```
# chcon -R -t httpd_sys_content_t /my/
# touch /my/website/index.html
# ls -Z /my/website/index.html
-rw-r--r--  root root unconfined_u:object_r:httpd_sys_content_t:s0 /my/website/index.html
```

Use the `semanage fcontext` command to make label changes that survive a relabel and the `restorecon` command. This command adds changes to file-context configuration. Then, run the `restorecon` command, which reads file-context configuration, to apply the label change. The following example demonstrates creating a new directory and an `index.html` file for use by `httpd`, and persistently changing the label of that directory and file to allow `httpd` access to them:

1. Run `mkdir -p /my/website` as the root user to create a top-level directory structure to store files to be used by `httpd`.

2. Run the following command as the root user to add the label change to file-context configuration:

```
semanage fcontext -a -t httpd_sys_content_t "/my(/.*)?"
```

The "/my(/.*)?" expression means the label change applies to the /my/ directory and all files and directories under it.

3. Run touch /my/website/index.html as the root user to create a new file.

4. Run restorecon -R -v /my/ as the root user to apply the label changes (restorecon reads file-context configuration, which was modified by the semanage command in step 2):

```
# restorecon -R -v /my/
restorecon reset /my context unconfined_u:object_r:default_t:s0->system_u:object_r:httpd_sys_content_t:s0
restorecon reset /my/website context unconfined_u:object_r:default_t:s0-
>system_u:object_r:httpd_sys_content_t:s0
restorecon reset /my/website/index.html context unconfined_u:object_r:default_t:s0-
>system_u:object_r:httpd_sys_content_t:s0
```

10.3.3. Booleans

SELinux is based on the least level of access required for a service to run. Services can be run in a variety of ways; therefore, you must tell SELinux how you are running services. This can be achieved via Booleans that allow parts of SELinux policy to be changed at runtime, without any knowledge of SELinux policy writing. This allows changes, such as allowing services access to NFS file systems, without reloading or recompiling SELinux policy.

To modify the state of a Boolean, use the setsebool command. For example, to turn the allow_httpd_anon_write Boolean on, run the following command as the root user:

```
# setsebool -P allow_httpd_anon_write on
```

To turn a Boolean off, using the same example, simply change on to off in the command, as shown below:

```
# setsebool -P allow_httpd_anon_write off
```

注記

Do not use the -P option if you do not want setsebool changes to persist across reboots.

Below is a description of common Booleans available that cater for the way httpd is running:

allow_httpd_anon_write
: When disabled, this Boolean allows httpd only read access to files labeled with the public_content_rw_t type. Enabling this Boolean will allow httpd to write to files labeled with the public_content_rw_t type, such as a public directory containing files for a public file transfer service.

`allow_httpd_mod_auth_ntlm_winbind`
Enabling this Boolean allows access to NTLM and Winbind authentication mechanisms via the `mod_auth_ntlm_winbind` module in `httpd`.

`allow_httpd_mod_auth_pam`
Enabling this Boolean allows access to PAM authentication mechanisms via the `mod_auth_pam` module in `httpd`.

`allow_httpd_sys_script_anon_write`
This Boolean defines whether or not HTTP scripts are allowed write access to files labeled with the `public_content_rw_t` type, as used in a public file transfer service.

`httpd_builtin_scripting`
This Boolean defines access to `httpd` scripting. Having this Boolean enabled is often required for PHP content.

`httpd_can_network_connect`
When disabled, this Boolean prevents HTTP scripts and modules from initiating a connection to a network or remote port. Turn this Boolean on to allow this access.

`httpd_can_network_connect_db`
When disabled, this Boolean prevents HTTP scripts and modules from initiating a connection to database servers. Turn this Boolean on to allow this access.

`httpd_can_network_relay`
Turn this Boolean on when `httpd` is being used as a forward or reverse proxy.

`httpd_can_sendmail`
When disabled, this Boolean prevents HTTP modules from sending mail. This can prevent spam attacks should a vulnerability be found in `httpd`. Turn this Boolean on to allow HTTP modules to send mail.

`httpd_dbus_avahi`
When off, this Boolean denies `httpd` access to the `avahi` service via `D-Bus`. Turn this Boolean on to allow this access.

`httpd_enable_cgi`
When disabled, this Boolean prevents `httpd` from executing CGI scripts. Turn this Boolean on to allow `httpd` to execute CGI scripts (CGI scripts must be labeled with the `httpd_sys_script_exec_t` type).

`httpd_enable_ftp_server`
Turning this Boolean on will allow `httpd` to listen on the FTP port and act as an FTP server.

`httpd_enable_homedirs`
When disabled, this Boolean prevents `httpd` from accessing user home directories. Turn this Boolean on to allow `httpd` access to user home directories; for example, content in `/home/*/`.

`httpd_execmem`
When enabled, this Boolean allows `httpd` to execute programs that require memory addresses that are both executable and writeable. Enabling this Boolean is not recommended from a security standpoint as it reduces protection against buffer overflows, however certain modules and applications (such as Java and Mono applications) require this privilege.

`httpd_ssi_exec`
: This Boolean defines whether or not server side include (SSI) elements in a web page can be executed.

`httpd_tmp_exec`
: Enabling this Boolean allows `httpd` to execute files in temporary directories.

`httpd_tty_comm`
: This Boolean defines whether or not `httpd` is allowed access to the controlling terminal. Usually this access is not required, however in cases such as configuring an SSL certificate file, terminal access is required to display and process a password prompt.

`httpd_unified`
: When enabled, this Boolean allows `httpd_t` complete access to all of the `httpd` types (i.e. to execute, read, or write sys_content_t). When disabled, there is separation in place between web content that is read-only, writeable or executable. Disabling this Boolean ensures an extra level of security but adds the administrative overhead of having to individually label scripts and other web content based on the file access that each should have.

`httpd_use_cifs`
: Turn this Boolean on to allow `httpd` access to files on CIFS file systems that are labeled with the `cifs_t` type, such as file systems mounted via Samba.

`httpd_use_gpg`
: Enabling this Boolean allows `httpd` to make use of GPG encryption.

`httpd_use_nfs`
: Turn this Boolean on to allow `httpd` access to files on NFS file systems that are labeled with the `nfs_t` type, such as file systems mounted via NFS.

10.3.4. Configuration examples

The following examples provide real-world demonstrations of how SELinux complements the Apache HTTP Server and how full function of the Apache HTTP Server can be maintained.

10.3.4.1. Running a static site

To create a static website, label the `.html` files for that website with the `httpd_sys_content_t` type. By default, the Apache HTTP Server can not write to files that are labeled with the `httpd_sys_content_t` type. The following example creates a new directory to store files for a read-only website:

1. Run `mkdir /mywebsite` as the root user to create a top-level directory.

2. As the root user, create a `/mywebsite/index.html` file. Copy and paste the following content into `/mywebsite/index.html`:

```
<html>
<h2>index.html from /mywebsite/</h2>
</html>
```

3. To allow the Apache HTTP Server read only access to `/mywebsite/`, as well as files and subdirectories under it, label `/mywebsite/` with the `httpd_sys_content_t` type. Run the following command as the root user to add the label change to file-context configuration:

```
# semanage fcontext -a -t httpd_sys_content_t "/mywebsite(/.*)?"
```

4. Run `restorecon -R -v /mywebsite` as the root user to make the label changes:

```
# restorecon -R -v /mywebsite
restorecon reset /mywebsite context unconfined_u:object_r:default_t:s0-
>system_u:object_r:httpd_sys_content_t:s0
restorecon reset /mywebsite/index.html context unconfined_u:object_r:default_t:s0-
>system_u:object_r:httpd_sys_content_t:s0
```

5. For this example, edit `/etc/httpd/conf/httpd.conf` as the root user. Comment out the existing `DocumentRoot` option. Add a `DocumentRoot "/mywebsite"` option. After editing, these options should look as follows:

```
#DocumentRoot "/var/www/html"
DocumentRoot "/mywebsite"
```

6. Run `service httpd status` as the root user to see the status of the Apache HTTP Server. If the server is stopped, run `service httpd start` as the root user to start it. If the server is running, run `service httpd restart` as the root user to restart the service (this also applies any changes made to `httpd.conf`).

7. Use a web browser to navigate to `http://localhost/index.html`. The following is displayed:

```
index.html from /mywebsite/
```

10.3.4.2. Sharing NFS and CIFS file systems

By default, NFS mounts on the client side are labeled with a default context defined by policy for NFS file systems. In common policies, this default context uses the `nfs_t` type. Also, by default, Samba shares mounted on the client side are labeled with a default context defined by policy. In common policies, this default context uses the `cifs_t` type.

Depending on policy configuration, services may not be able to read files labeled with the `nfs_t` or `cifs_t` types. This may prevent file systems labeled with these types from being mounted and then read or exported by other services. Booleans can be turned on or off to control which services are allowed to access the `nfs_t` and `cifs_t` types.

Turn the `httpd_use_nfs` Boolean on to allow `httpd` to access and share NFS file systems (labeled with the `nfs_t` type. Run the `setsebool` command as the root user to turn the Boolean on:

```
setsebool -P httpd_use_nfs on
```

Turn the `httpd_use_cifs` Boolean on to allow `httpd` to access and share CIFS file systems (labeled with the `cifs_t` type. Run the `setsebool` command as the root user to turn the Boolean on:

```
setsebool -P httpd_use_cifs on
```

注記

Do not use the -P option if you do not want `setsebool` changes to persist across reboots.

10.3.4.3. Sharing files between services

Type Enforcement helps prevent processes from accessing files intended for use by another process. For example, by default, Samba can not read files labeled with the `httpd_sys_content_t` type, which are intended for use by the Apache HTTP Server. Files can be shared between the Apache HTTP Server, FTP, rsync, and Samba, if the desired files are labeled with the `public_content_t` or `public_content_rw_t` type.

The following example creates a directory and files, and allows that directory and files to be shared (read only) through the Apache HTTP Server, FTP, rsync, and Samba:

1. Run `mkdir /shares` as the root user to create a new top-level directory to share files between multiple services.
2. Files and directories that do not match a pattern in file-context configuration may be labeled with the `default_t` type. This type is inaccessible to confined services:

```
$ ls -dZ /shares
drwxr-xr-x  root root unconfined_u:object_r:default_t:s0 /shares
```

3. As the root user, create a `/shares/index.html` file. Copy and paste the following content into `/shares/index.html`:

```
<html>
<body>
<p>Hello</p>
</body>
</html>
```

4. Labeling `/shares/` with the `public_content_t` type allows read-only access by the Apache HTTP Server, FTP, rsync, and Samba. Run the following command as the root user to add the label change to file-context configuration:

```
semanage fcontext -a -t public_content_t "/shares(/.*)?"
```

5. Run `restorecon -R -v /shares/` as the root user to apply the label changes:

```
# restorecon -R -v /shares/
restorecon reset /shares context unconfined_u:object_r:default_t:s0->system_u:object_r:public_content_t:s0
restorecon reset /shares/index.html context unconfined_u:object_r:default_t:s0-
>system_u:object_r:public_content_t:s0
```

To share `/shares/` through Samba:

1. Run `rpm -q samba samba-common samba-client` to confirm the *samba*, *samba-common*, and *samba-client* packages are installed (version numbers may differ):

```
$ rpm -q samba samba-common samba-client
samba-3.5.2-59.fc13.i386
samba-common-3.5.2-59.fc13.i386
samba-client-3.5.2-59.fc13.i386
```

   If any of these packages are not installed, install them by running `yum install` ***package-name*** as the root user.

2. Edit `/etc/samba/smb.conf` as the root user. Add the following entry to the bottom of this file to share the `/shares/` directory through Samba:

```
[shares]
comment = Documents for Apache HTTP Server, FTP, rsync, and Samba
path = /shares
public = yes
writeable = no
```

3. A Samba account is required to mount a Samba file system. Run `smbpasswd -a` ***username*** as the root user to create a Samba account, where *username* is an existing Linux user. For example, `smbpasswd -a testuser` creates a Samba account for the Linux testuser user:

```
# smbpasswd -a testuser
New SMB password: Enter a password
Retype new SMB password: Enter the same password again
Added user testuser.
```

   Running `smbpasswd -a` ***username***, where *username* is the username of a Linux account that does not exist on the system, causes a `Cannot locate Unix account for '`***username***`'!` error.

4. Run `service smb start` as the root user to start the Samba service:

```
service smb start
Starting SMB services:                                     [  OK  ]
```

5. Run `smbclient -U` ***username*** `-L localhost` to list the available shares, where *username* is the Samba account added in step 3. When prompted for a password, enter the password assigned to the Samba account in step 3 (version numbers may differ):

```
$ smbclient -U username -L localhost
Enter username's password:
Domain=[HOSTNAME] OS=[Unix] Server=[Samba 3.5.2-59.fc13]

Sharename       Type      Comment
---------       ----      -------
shares          Disk      Documents for Apache HTTP Server, FTP, rsync, and Samba
IPC$            IPC       IPC Service (Samba Server Version 3.5.2-59)
username        Disk      Home Directories
Domain=[HOSTNAME] OS=[Unix] Server=[Samba 3.5.2-59.fc13]
```

```
Server               Comment
---------            -------

Workgroup            Master
---------            -------
```

6. Run `mkdir /test/` as the root user to create a new directory. This directory will be used to mount the `shares` Samba share.

7. Run the following command as the root user to mount the `shares` Samba share to `/test/`, replacing *`username`* with the username from step 3:

```
mount //localhost/shares /test/ -o user=username
```

   Enter the password for *`username`*, which was configured in step 3.

8. Run `cat /test/index.html` to view the file, which is being shared through Samba:

```
$ cat /test/index.html
<html>
<body>
<p>Hello</p>
</body>
</html>
```

To share `/shares/` through the Apache HTTP Server:

1. Run `rpm -q httpd` to confirm the *`httpd`* package is installed (version number may differ):

```
$ rpm -q httpd
httpd-2.2.11-6.i386
```

   If this package is not installed, run `yum install httpd` as the root user to install it.

2. Change into the `/var/www/html/` directory. Run the following command as the root user to create a link (named `shares`) to the `/shares/` directory:

```
ln -s /shares/ shares
```

3. Run `service httpd start` as the root user to start the Apache HTTP Server:

```
service httpd start
Starting httpd:                                        [  OK  ]
```

4. Use a web browser to navigate to `http://localhost/shares`. The `/shares/index.html` file is displayed.

By default, the Apache HTTP Server reads an `index.html` file if it exists. If `/shares/` did not have `index.html`, and instead had `file1`, `file2`, and `file3`, a directory listing would occur when accessing `http://localhost/shares`:

1. Run `rm -i /shares/index.html` as the root user to remove the `index.html` file.

2. Run `touch /shares/file{1,2,3}` as the root user to create three files in `/shares/`:

```
# touch /shares/file{1,2,3}
# ls -Z /shares/
-rw-r--r--  root root system_u:object_r:public_content_t:s0 file1
-rw-r--r--  root root unconfined_u:object_r:public_content_t:s0 file2
-rw-r--r--  root root unconfined_u:object_r:public_content_t:s0 file3
```

3. Run `service httpd status` as the root user to see the status of the Apache HTTP Server. If the server is stopped, run `service httpd start` as the root user to start it.

4. Use a web browser to navigate to `http://localhost/shares`. A directory listing is displayed:

10.3.4.4. Changing port numbers

Depending on policy configuration, services may only be allowed to run on certain port numbers. Attempting to change the port a service runs on without changing policy may result in the service failing to start. Run `semanage port -l | grep -w "http_port_t"` as the root user to list the ports SELinux allows `httpd` to listen on:

```
# semanage port -l | grep -w http_port_t
http_port_t                    tcp      80, 443, 488, 8008, 8009, 8443
```

By default, SELinux allows `http` to listen on TCP ports 80, 443, 488, 8008, 8009, or 8443. If `/etc/httpd/conf/httpd.conf` is configured so that `httpd` listens on any port not listed for `http_port_t`, `httpd` fails to start.

To configure `httpd` to run on a port other than TCP ports 80, 443, 488, 8008, 8009, or 8443:

1. Edit `/etc/httpd/conf/httpd.conf` as the root user so the `Listen` option lists a port that is not configured in SELinux policy for `httpd`. The following example configures `httpd` to listen on the 10.0.0.1 IP address, and on port 12345:

```
# Change this to Listen on specific IP addresses as shown below to
```

```
# prevent Apache from glomming onto all bound IP addresses (0.0.0.0)
#
#Listen 12.34.56.78:80
Listen 10.0.0.1:12345
```

2. Run `semanage port -a -t http_port_t -p tcp 12345` as the root user to add the port to SELinux policy configuration.

3. Run `semanage port -l | grep -w http_port_t` as the root user to confirm the port is added:

```
# semanage port -l | grep -w http_port_t
http_port_t                    tcp      12345, 80, 443, 488, 8008, 8009, 8443
```

If you no longer run `httpd` on port 12345, run `semanage port -d -t http_port_t -p tcp 12345` as the root user to remove the port from policy configuration.

10.4. Samba

From the *Samba*[7] website:

"Samba is an *Open Source*[8]/*Free Software*[9] suite that has, *since 1992*[10], provided file and print services to all manner of SMB/CIFS clients, including the numerous versions of Microsoft Windows operating systems. Samba is freely available under the *GNU General Public License*[11]."[12]

In Fedora, the *samba* package provides the Samba server. Run `rpm -q samba` to see if the *samba* package is installed. If it is not installed and you want to use Samba, run the following command as the root user to install it:

```
yum install samba
```

10.4.1. Samba and SELinux

When SELinux is enabled, the Samba server (`smbd`) runs confined by default. Confined services run in their own domains, and are separated from other confined services. The following example demonstrates the `smbd` process running in its own domain. This example assumes the *samba* package is installed:

1. Run `getenforce` to confirm SELinux is running in enforcing mode:

```
$ getenforce
Enforcing
```

   The `getenforce` command returns `Enforcing` when SELinux is running in enforcing mode.

2. Run `service smbd start` as the root user to start `smbd`:

[7] http://samba.org/
[8] http://www.opensource.org/
[9] http://www.gnu.org/philosophy/free-sw.html
[10] http://us1.samba.org/samba/docs/10years.html
[11] http://us1.samba.org/samba/docs/GPL.html
[12] From the opening paragraph on the Samba website: *http://samba.org*. Accessed 20 January 2009.

```
service smb start
Starting SMB services:                                     [  OK  ]
```

3. Run `ps -eZ | grep smb` to view the `smbd` processes:

```
$ ps -eZ | grep smb
unconfined_u:system_r:smbd_t:s0 16420 ?        00:00:00 smbd
unconfined_u:system_r:smbd_t:s0 16422 ?        00:00:00 smbd
```

   The SELinux context associated with the `smbd` processes is `unconfined_u:system_r:smbd_t:s0`. The second last part of the context, `smbd_t`, is the type. A type defines a domain for processes and a type for files. In this case, the `smbd` processes are running in the smbd_t domain.

Files must be labeled correctly to allow `smbd` to access and share them. For example, `smbd` can read and write to files labeled with the `samba_share_t` type, but by default, can not access files labeled with the `httpd_sys_content_t` type, which is intended for use by the Apache HTTP Server. Booleans must be turned on to allow certain behavior, such as allowing home directories and NFS file systems to be exported through Samba, as well as to allow Samba to act as a domain controller.

10.4.2. Types

Label files with the `samba_share_t` type to allow Samba to share them. Only label files you have created, and do not relabel system files with the `samba_share_t` type: Booleans can be turned on to share such files and directories. SELinux allows Samba to write to files labeled with the `samba_share_t` type, as long as `/etc/samba/smb.conf` and Linux permissions are set accordingly.

The `samba_etc_t` type is used on certain files in `/etc/samba/`, such as `smb.conf`. Do not manually label files with the `samba_etc_t` type. If files in `/etc/samba/` are not labeled correctly, run `restorecon -R -v /etc/samba` as the root user to restore such files to their default contexts. If `/etc/samba/smb.conf` is not labeled with the `samba_etc_t` type, the `service smb start` command may fail and an SELinux denial may be logged. The following is an example denial logged to `/var/log/messages` when `/etc/samba/smb.conf` was labeled with the `httpd_sys_content_t` type:

```
setroubleshoot: SELinux is preventing smbd (smbd_t) "read" to ./smb.conf (httpd_sys_content_t). For complete
 SELinux messages. run sealert -l deb33473-1069-482b-bb50-e4cd05ab18af
```

10.4.3. Booleans

SELinux is based on the least level of access required for a service to run. Services can be run in a variety of ways; therefore, you must tell SELinux how you are running services. The following Booleans allow you to tell SELinux how you are running Samba:

`allow_smbd_anon_write`
: Having this Boolean enables allows `smbd` to write to a public directory, such as an area reserved for common files that otherwise has no special access restrictions.

`samba_create_home_dirs`
: Having this Boolean enabled allows Samba to create new home directories independently. This is often done by mechanisms such as PAM.

`samba_domain_controller`
When enabled, this Boolean allows Samba to act as a domain controller, as well as giving it permission to execute related commands such as `useradd`, `groupadd` and `passwd`.

`samba_enable_home_dirs`
Enabling this Boolean allows Samba to share users' home directories.

`samba_export_all_ro`
Export any file or directory, allowing read-only permissions. This allows files and directories that are not labeled with the `samba_share_t` type to be shared through Samba. When the `samba_export_all_ro` Boolean is on, but the `samba_export_all_rw` Boolean is off, write access to Samba shares is denied, even if write access is configured in `/etc/samba/smb.conf`, as well as Linux permissions allowing write access.

`samba_export_all_rw`
Export any file or directory, allowing read and write permissions. This allows files and directories that are not labeled with the `samba_share_t` type to be exported through Samba. Permissions in `/etc/samba/smb.conf` and Linux permissions must be configured to allow write access.

`samba_run_unconfined`
Having this Boolean enabled allows Samba to run unconfined scripts in the `/var/lib/samba/scripts` directory.

`samba_share_fusefs`
This Boolean must be enabled for Samba to share `fusefs` file systems.

`samba_share_nfs`
Disabling this Boolean prevents `smbd` from having full access to NFS shares via Samba. Enabling this Boolean will allow Samba to share NFS file systems.

`use_samba_home_dirs`
Enable this Boolean to use a remote server for Samba home directories.

`virt_use_samba`
Allow virt to manage CIFS files.

10.4.4. Configuration examples

The following examples provide real-world demonstrations of how SELinux complements the Samba server and how full function of the Samba server can be maintained.

10.4.4.1. Sharing directories you create

The following example creates a new directory, and shares that directory through Samba:

1. Run `rpm -q samba samba-common samba-client` to confirm the *samba*, *samba-common*, and *samba-client* packages are installed. If any of these packages are not installed, install them by running `yum install` ***`package-name`*** as the root user.

2. Run `mkdir /myshare` as the root user to create a new top-level directory to share files through Samba.

3. Run `touch /myshare/file1` as the root user to create an empty file. This file is used later to verify the Samba share mounted correctly.

4. SELinux allows Samba to read and write to files labeled with the samba_share_t type, as long as /etc/samba/smb.conf and Linux permissions are set accordingly. Run the following command as the root user to add the label change to file-context configuration:

```
semanage fcontext -a -t samba_share_t "/myshare(/.*)?"
```

5. Run restorecon -R -v /myshare as the root user to apply the label changes:

```
# restorecon -R -v /myshare
restorecon reset /myshare context unconfined_u:object_r:default_t:s0->system_u:object_r:samba_share_t:s0
restorecon reset /myshare/file1 context unconfined_u:object_r:default_t:s0-
>system_u:object_r:samba_share_t:s0
```

6. Edit /etc/samba/smb.conf as the root user. Add the following to the bottom of this file to share the /myshare/ directory through Samba:

```
[myshare]
comment = My share
path = /myshare
public = yes
writeable = no
```

7. A Samba account is required to mount a Samba file system. Run smbpasswd -a ***username*** as the root user to create a Samba account, where *username* is an existing Linux user. For example, smbpasswd -a testuser creates a Samba account for the Linux testuser user:

```
# smbpasswd -a testuser
New SMB password: Enter a password
Retype new SMB password: Enter the same password again
Added user testuser.
```

Running smbpasswd -a ***username***, where *username* is the username of a Linux account that does not exist on the system, causes a Cannot locate Unix account for '***username***'! error.

8. Run service smb start as the root user to start the Samba service:

```
service smb start
Starting SMB services:                                     [  OK  ]
```

9. Run smbclient -U ***username*** -L localhost to list the available shares, where *username* is the Samba account added in step 7. When prompted for a password, enter the password assigned to the Samba account in step 7 (version numbers may differ):

```
$ smbclient -U username -L localhost
Enter username's password:
Domain=[HOSTNAME] OS=[Unix] Server=[Samba 3.5.2-59.fc13]

Sharename       Type      Comment
---------       ----      -------
myshare         Disk      My share
```

```
IPC$            IPC       IPC Service (Samba Server Version 3.5.2-59.fc13)
username        Disk      Home Directories
Domain=[HOSTNAME] OS=[Unix] Server=[Samba 3.5.2-59.fc13]

Server               Comment
---------            -------

Workgroup            Master
---------            -------
```

10. Run `mkdir /test/` as the root user to create a new directory. This directory will be used to mount the `myshare` Samba share.

11. Run the following command as the root user to mount the `myshare` Samba share to `/test/`, replacing *`username`* with the username from step 7:

```
mount //localhost/myshare /test/ -o user=username
```

    Enter the password for *`username`*, which was configured in step 7.

12. Run `ls /test/` to view the `file1` file created in step 3:

```
$ ls /test/
file1
```

10.4.4.2. Sharing a website

It may not be possible to label files with the `samba_share_t` type, for example, when wanting to share a website in `/var/www/html/`. For these cases, use the `samba_export_all_ro` Boolean to share any file or directory (regardless of the current label), allowing read only permissions, or the `samba_export_all_rw` Boolean to share any file or directory (regardless of the current label), allowing read and write permissions.

The following example creates a file for a website in `/var/www/html/`, and then shares that file through Samba, allowing read and write permissions. This example assumes the *httpd*, *samba*, *samba-common*, *samba-client*, and *wget* packages are installed:

1. As the root user, create a `/var/www/html/file1.html` file. Copy and paste the following content into `/var/www/html/file1.html`:

```
<html>
<h2>File being shared through the Apache HTTP Server and Samba.</h2>
</html>
```

2. Run `ls -Z /var/www/html/file1.html` to view the SELinux context of `file1.html`:

```
$ ls -Z /var/www/html/file1.html
-rw-r--r--. root root unconfined_u:object_r:httpd_sys_content_t:s0 /var/www/html/file1.html
```

    `file1.index.html` is labeled with the `httpd_sys_content_t`. By default, the Apache HTTP Server can access this type, but Samba can not.

3. Run `service httpd start` as the root user to start the Apache HTTP Server:

```
service httpd start
Starting httpd:                                            [  OK  ]
```

4. Change into a directory your user has write access to, and run the `wget http://localhost/file1.html` command. Unless there are changes to the default configuration, this command succeeds:

```
$ wget http://localhost/file1.html
--2009-03-02 16:32:01--  http://localhost/file1.html
Resolving localhost... 127.0.0.1
Connecting to localhost|127.0.0.1|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 84 [text/html]
Saving to: `file1.html.1'

100%[======================>] 84          --.-K/s   in 0s

2009-03-02 16:32:01 (563 KB/s) - `file1.html.1' saved [84/84]
```

5. Edit `/etc/samba/smb.conf` as the root user. Add the following to the bottom of this file to share the `/var/www/html/` directory through Samba:

```
[website]
comment = Sharing a website
path = /var/www/html/
public = no
writeable = no
```

6. The `/var/www/html/` directory is labeled with the `httpd_sys_content_t` type. By default, Samba can not access files and directories labeled with the `httpd_sys_content_t` type, even if Linux permissions allow it. To allow Samba access, run the following command as the root user to turn the `samba_export_all_ro` Boolean on:

```
setsebool -P samba_export_all_ro on
```

   Do not use the `-P` option if you do not want the change to persist across reboots. Note: turning the `samba_export_all_ro` Boolean on allows Samba to access any type.

7. Run `service smb start` as the root user to start `smbd`:

```
service smb start
Starting SMB services:                                     [  OK  ]
```

10.5. File Transfer Protocol

From the *Red Hat Enterprise Linux 5 Deployment Guide*[13]:

File Transfer Protocol (FTP) is one of the oldest and most commonly used protocols found on the Internet today. Its purpose is to reliably transfer files between computer hosts on a network without requiring the user to log directly into the remote host or have knowledge of how to use the remote system. It allows users to access files on remote systems using a standard set of simple commands.[14]

The Very Secure FTP Daemon (`vsftpd`) is designed from the ground up to be fast, stable, and, most importantly, secure. Its ability to handle large numbers of connections efficiently and securely is why `vsftpd` is the only stand-alone FTP distributed with Red Hat Enterprise Linux.[15]

In Fedora, the *vsftpd* package provides the Very Secure FTP daemon. Run `rpm -q vsftpd` to see if *vsftpd* is installed:

```
$ rpm -q vsftpd
```

If you want an FTP server and the *vsftpd* package is not installed, run the following command as the root user to install it:

```
yum install vsftpd
```

10.5.1. FTP and SELinux

When running SELinux, the FTP server, `vsftpd`, runs confined by default. SELinux policy defines how `vsftpd` interacts with files, processes, and with the system in general. For example, when an authenticated user logs in via FTP, they can not read from or write to files in their home directories: SELinux prevents `vsftpd` from accessing user home directories by default. Also, by default, `vsftpd` does not have access to NFS or CIFS file systems, and anonymous users do not have write access, even if such write access is configured in `/etc/vsftpd/vsftpd.conf`. Booleans can be turned on to allow the previously mentioned access.

The following example demonstrates an authenticated user logging in, and an SELinux denial when trying to view files in their home directory:

1. Run `rpm -q vsftpd` to see if the *vsftpd* package is installed. If it is not, run `yum install vsftpd` as the root user to install it.

2. In Fedora, `vsftpd` only allows anonymous users to log in by default. To allow authenticated users to log in, edit `/etc/vsftpd/vsftpd.conf` as the root user. Uncomment the `local_enable=YES` option:

   ```
   # Uncomment this to allow local users to log in.
   ```

[13] http://www.redhat.com/docs/en-US/Red_Hat_Enterprise_Linux/5/html/Deployment_Guide/index.html
[14] The first paragraph of "Chapter 23. FTP" of the Red Hat Enterprise Linux 5 Deployment Guide: *http://www.redhat.com/docs/en-US/Red_Hat_Enterprise_Linux/5/html/Deployment_Guide/ch-ftp.html*. Copyright © 2007 Red Hat, Inc.
[15] The first paragraph of the "23.2.1. vsftpd" section of the Red Hat Enterprise Linux 5 Deployment Guide: *http://www.redhat.com/docs/en-US/Red_Hat_Enterprise_Linux/5/html/Deployment_Guide/s1-ftp-servers.html#s2-ftp-servers-vsftpd*. Copyright © 2007 Red Hat, Inc.

```
local_enable=YES
```

3. Run `service vsftpd start` as the root user to start `vsftpd`. If the service was running before editing `vsftpd.conf`, run `service vsftpd restart` as the root user to apply the configuration changes:

```
service vsftpd start
Starting vsftpd for vsftpd:                                [  OK  ]
```

4. Run `ftp localhost` as the user you are currently logged in with. When prompted for your name, make sure your username is displayed. If the correct username is displayed, press `Enter`, otherwise, enter the correct username:

```
$ ftp localhost
Connected to localhost (127.0.0.1).
220 (vsFTPd 2.1.0)
Name (localhost:username):
331 Please specify the password.
Password: Enter your password
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp>
```

5. Run the `ls` command from the `ftp` prompt. With the `ftp_home_dir` Boolean off, SELinux prevents `vsftpd` access to home directories, resulting in this command failing to return a directory listing:

```
ftp> ls
227 Entering Passive Mode (127,0,0,1,225,210).
150 Here comes the directory listing.
226 Transfer done (but failed to open directory).
```

An SELinux denial similar to the following is logged to `/var/log/messages`:

```
setroubleshoot: SELinux is preventing the ftp daemon from reading users home directories (username). For
 complete SELinux messages. run sealert -l c366e889-2553-4c16-b73f-92f36a1730ce
```

6. Enable the `ftp_home_dir` Boolean by running the following command as the root user:

```
# setsebool -P ftp_home_dir=1
```

注記

Do not use the -P option if you do not want changes to persist across reboots.

Run the `ls` command again from the `ftp` prompt. Now that SELinux is allowing home directory browsing via the `ftp_home_dir` Boolean, the directory is displayed:

```
ftp> ls
227 Entering Passive Mode (127,0,0,1,56,215).
150 Here comes the directory listing.
-rw-rw-r--    1 501      501             0 Mar 30 09:22 file1
-rw-rw-r--    1 501      501             0 Mar 30 09:22 file2
226 Directory Send OK.
ftp>
```

10.5.2. Types

By default, anonymous users have read access to files in `/var/ftp/` when they log in via FTP. This directory is labeled with the `public_content_t` type, allowing only read access, even if write access is configured in `/etc/vsftpd/vsftpd.conf`. The `public_content_t` type is accessible to other services, such as Apache HTTP Server, Samba, and NFS.

Use one of the following types to share files through FTP:

`public_content_t`
: Label files and directories you have created with the `public_content_t` type to share them read-only through vsftpd. Other services, such as Apache HTTP Server, Samba, and NFS, also have access to files labeled with this type. Files labeled with the `public_content_t` type can not be written to, even if Linux permissions allow write access. If you require write access, use the `public_content_rw_t` type.

`public_content_rw_t`
: Label files and directories you have created with the `public_content_rw_t` type to share them with read and write permissions through `vsftpd`. Other services, such as Apache HTTP Server, Samba, and NFS, also have access to files labeled with this type; however, Booleans for each service must be turned on before such services can write to files labeled with this type.

10.5.3. Booleans

SELinux is based on the least level of access required for a service to run. Services can be run in a variety of ways; therefore, you must tell SELinux how you are running services. The following Booleans allow you to tell SELinux how you are running `vsftpd`:

`allow_ftpd_anon_write`
: When disabled, this Boolean prevents `vsftpd` from writing to files and directories labeled with the `public_content_rw_t` type. Turn this Boolean on to allow users to upload files via FTP. The directory where files are uploaded to must be labeled with the `public_content_rw_t` type and Linux permissions set accordingly.

`allow_ftpd_full_access`
: When this Boolean is on, only Linux permissions are used to control access, and authenticated users can read and write to files that are not labeled with the `public_content_t` or `public_content_rw_t` types.

`allow_ftpd_use_cifs`
: Having this Boolean enabled allows `vsftpd` to access files and directories labeled with the `cifs_t` type; therefore, having this Boolean enabled allows you to share file systems mounted via Samba through `vsftpd`.

`allow_ftpd_use_nfs`
: Having this Boolean enabled allows `vsftpd` to access files and directories labeled with the `nfs_t` type; therefore, having this Boolean enabled allows you to share file systems mounted via NFS through `vsftpd`.

`ftp_home_dir`
: Having this Boolean enabled allows authenticated users to read and write to files in their home directories. When this Boolean is off, attempting to download a file from a home directory results in an error such as `550 Failed to open file`. An SELinux denial is logged to `/var/log/messages`.

`ftpd_connect_db`
: Allow FTP daemons to initiate a connection to a database.

`httpd_enable_ftp_server`
: Allow `httpd` to listen on the FTP port and act as a FTP server.

`tftp_anon_write`
: Having this Boolean enabled allows TFTP access to a public directory, such as an area reserved for common files that otherwise has no special access restrictions.

10.5.4. Configuration Examples

10.5.4.1. Uploading to an FTP site

The following example creates an FTP site that allows a dedicated user to upload files. It creates the directory structure and the required SELinux configuration changes:

1. Run `mkdir -p /myftp/pub` as the root user to create a new top-level directory.

2. Set Linux permissions on the `/myftp/pub/` directory to allow a Linux user write access. This example changes the owner and group from root to owner user1 and group root. Replace user1 with the user you want to give write access to:

   ```
   # chown user1:root /myftp/pub
   # chmod 775 /myftp/pub
   ```

   The `chown` command changes the owner and group permissions. The `chmod` command changes the mode, allowing the user1 user read, write, and execute permissions, and members of the root group read, write, and execute permissions. Everyone else has read and execute permissions: this is required to allow the Apache HTTP Server to read files from this directory.

3. When running SELinux, files and directories must be labeled correctly to allow access. Setting Linux permissions is not enough. Files labeled with the `public_content_t` type allow them to be read by FTP, Apache HTTP Server, Samba, and rsync. Files labeled with the `public_content_rw_t` type can be written to by FTP. Other services, such as Samba, require Booleans to be set before they can write to files labeled with the `public_content_rw_t` type. Label the top-level directory (`/myftp/`) with the `public_content_t` type, to prevent copied or newly-created files under `/myftp/` from being written to or modified by services. Run the following command as the root user to add the label change to file-context configuration:

   ```
   ```

```
semanage fcontext -a -t public_content_t /myftp
```

4. Run `restorecon -R -v /myftp/` to apply the label change:

```
# restorecon -R -v /myftp/
restorecon reset /myftp context unconfined_u:object_r:default_t:s0->system_u:object_r:public_content_t:s0
```

5. Confirm `/myftp` is labeled with the `public_content_t` type, and `/myftp/pub/` is labeled with the `default_t` type:

```
$ ls -dZ /myftp/
drwxr-xr-x. root root system_u:object_r:public_content_t:s0 /myftp/
$ ls -dZ /myftp/pub/
drwxrwxr-x. user1 root unconfined_u:object_r:default_t:s0 /myftp/pub/
```

6. FTP must be allowed to write to a directory before users can upload files via FTP. SELinux allows FTP to write to directories labeled with the `public_content_rw_t` type. This example uses `/myftp/pub/` as the directory FTP can write to. Run the following command as the root user to add the label change to file-context configuration:

```
semanage fcontext -a -t public_content_rw_t "/myftp/pub(/.*)?"
```

7. Run `restorecon -R -v /myftp/pub` as the root user to apply the label change:

```
# restorecon -R -v /myftp/pub
restorecon reset /myftp/pub context system_u:object_r:default_t:s0-
>system_u:object_r:public_content_rw_t:s0
```

8. The `allow_ftpd_anon_write` Boolean must be on to allow `vsftpd` to write to files that are labeled with the `public_content_rw_t` type. Run the following command as the root user to turn this Boolean on:

```
setsebool -P allow_ftpd_anon_write on
```

   Do not use the `-P` option if you do not want changes to persist across reboots.

The following example demonstrates logging in via FTP and uploading a file. This example uses the user1 user from the previous example, where user1 is the dedicated owner of the `/myftp/pub/` directory:

1. Run `cd ~/` to change into your home directory. Then, run `mkdir myftp` to create a directory to store files to upload via FTP.

2. Run `cd ~/myftp` to change into the `~/myftp/` directory. In this directory, create an `ftpupload` file. Copy the following contents into this file:

```
File upload via FTP from a home directory.
```

3. Run `getsebool allow_ftpd_anon_write` to confirm the `allow_ftpd_anon_write` Boolean is on:

```
$ getsebool allow_ftpd_anon_write
allow_ftpd_anon_write --> on
```

   If this Boolean is off, run `setsebool -P allow_ftpd_anon_write on` as the root user to turn it on. Do not use the `-P` option if you do not want the change to persist across reboots.

4. Run `service vsftpd start` as the root user to start `vsftpd`:

```
# service vsftpd start
Starting vsftpd for vsftpd:                                [  OK  ]
```

5. Run `ftp localhost`. When prompted for a username, enter the the username of the user who has write access, then, enter the correct password for that user:

```
$ ftp localhost
Connected to localhost (127.0.0.1).
220 (vsFTPd 2.1.0)
Name (localhost:username):
331 Please specify the password.
Password: Enter the correct password
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp>
```

10.6. Network File System

From the *Red Hat Linux Reference Guide*[16]:

NFS (Network File System) allows hosts to mount partitions on a remote system and use them as though they are local file systems. This allows the system administrator to store resources in a central location on the network, providing authorized users continuous access to them.

In Fedora, the *nfs-utils* package is required for full NFS support. Run `rpm -q nfs-utils` to see if the *nfs-utils* is installed. If it is not installed and you want to use NFS, run the following command as the root user to install it:

```
yum install nfs-utils
```

10.6.1. NFS and SELinux

When running SELinux, the `NFS` daemons are confined by default. SELinux policy does not allow NFS to share files by default. If you want to share NFS partitions, this can be configured via the `nfs_export_all_ro` and `nfs_export_all_rw` Booleans, as described below. These Booleans are however not required when files to be shared are labeled with the `public_content_t`

[16] http://www.redhat.com/docs/manuals/linux/RHL-9-Manual/ref-guide/ch-nfs.html

or public_content_rw_t types. NFS can share files labeled with these types even if the nfs_export_all_ro and nfs_export_all_rw Booleans are off.

10.6.2. Types

By default, mounted NFS file systems on the client side are labeled with a default context defined by policy for NFS file systems. In common policies, this default context uses the nfs_t type.The following types are used with NFS. Different types allow you to configure flexible access:

var_lib_nfs_t
: This type is used for existing and new files copied to or created in the /var/lib/nfs directory. This type should not need to be changed in normal operation. To restore changes to the default settings, run the restorecon -R -v /var/lib/nfs command as the root user.

nfsd_exec_t
: The /usr/sbin/rpc.nfsd file is labeled with the nfsd_exec_t, as are other system executables and libraries related to NFS. Users should not label any files with this type. nfsd_exec_t will transition to nfs_t.

10.6.3. Booleans

SELinux is based on the least level of access required for a service to run. Services can be run in a variety of ways; therefore, you must tell SELinux how you are running services. The following Booleans allow you to tell SELinux how you are running NFS:

allow_ftpd_use_nfs
: When enabled, this Boolean allows ftpd access to NFS mounts.

allow_nfsd_anon_write
: When enabled, this Boolean allows nfsd to write to a public directory anonymously; such as to an area reserved for common files that otherwise has no special access restrictions.

httpd_use_nfs
: When enabled, this Boolean will allow httpd to access files stored on a NFS filesystem.

nfs_export_all_ro
: Export any file or directory via NFS, allowing read-only permissions.

nfs_export_all_rw
: Export any file or directory via NFS, allowing read and write permissions.

qemu_use_nfs
: Allow qemu to use NFS file systems.

samba_share_nfs
: When disabled, this Boolean prevents smbd from having full access to NFS shares via Samba. Enabling this Boolean will allow Samba to share NFS file systems.

use_nfs_home_dirs
: Having this Boolean enabled adds support for NFS home directories.

virt_use_nfs
: Allow virt to use NFS files.

xen_use_nfs
: Allow xen to manage NFS files.

10.6.4. Configuration Examples

10.6.4.1. Sharing directories using NFS

The example in this section creates a directory and shares it using NFS and SELinux. Two hosts are used in this example; a NFS server with a hostname of `nfs-srv` with an IP address of `192.168.1.1`, and a client with a hostname of `nfs-client` and an IP address of `192.168.1.100`. Both hosts are on the same subnet (192.168.1.0/24). This is an example only and assumes that the *nfs-utils* package is installed, that the SELinux targeted policy is used, and that SELinux is running in enforced mode.

This example will show that while even with full network availability and Linux file permissions granting access to all users via NFS, SELinux is still able to block mounting of NFS file systems unless the proper permissions are given via SELinux Booleans.

10.6.4.1.1. Server setup

Steps 1-10 below should be performed on the NFS server, `nfs-srv`.

1. Run the `setsebool` command to disable read/write mounting of NFS file systems:

```
setsebool -P nfs_export_all_rw off
```

注記

Do not use the `-P` option if you do not want `setsebool` changes to persist across reboots.

2. Run `rpm -q nfs-utils` to confirm the *nfs-utils* package is installed. The *nfs-utils* package provides support programs for using NFS and should be installed on a NFS server and on any clients in use. If this package is not installed, install it by running `yum install` ***nfs-utils*** as the root user.

3. Run `mkdir /myshare` as the root user to create a new top-level directory to share using NFS.

4. Run `touch /myshare/file1` as the root user to create a new empty file in the shared area. This file will be accessed later by the client.

5. To show that SELinux is still able to block access even when Linux permissions are completely open, give the `/myshare` directory full Linux access rights for all users:

```
# chmod -R 777 /myshare
```

警告

This is an example only and these permissions should not be used in a production system.

6. Edit the /etc/exports file and add the following line to the top of the file:

```
/myshare  192.168.1.100(rw)
```

   This entry shows the full path on the server to the shared folder /myshare, the host or network range that nfs-srv will share to (in this case the IP address of a single host, nfs-client at 192.168.1.100), and finally the share permissions. Read and write permissions are given here, as indicated by (rw).

7. The TCP and UDP ports used for NFS are assigned dynamically by rpcbind, which can cause problems when creating firewall rules. To simplify the process of allowing NFS traffic through the firewall in this example, edit the */etc/sysconfig/nfs* file and uncomment the MOUNTD_PORT,STATD_PORT,LOCKD_TCPPORT and LOCKD_UDPPORT variables. Changing the port numbers in this file is not required for this example.

   Ensure that incoming connections on TCP ports 111, 892 and 2049 are allowed through the server's firewall. This can be done via the *system-config-firewall* tool in Fedora.

8. Run service nfs start as the root user to start NFS and its related services:

```
# service nfs start
Starting NFS services:  [  OK  ]
Starting NFS quotas:  [  OK  ]
Starting NFS daemon:  [  OK  ]
Starting NFS mountd:  [  OK  ]
```

9. To ensure that the NFS subsystem export table is updated, run exportfs -rv as the root user:

```
# exportfs -rv
exporting 192.168.1.100:/myshare
```

10. Run showmount -e as the root user to show all exported file systems:

```
# showmount -e
Export list for nfs-srv:
/myshare 192.168.1.100
```

At this point the server nfs-srv has been configured to allow NFS communications to nfs-client at 192.168.1.100, and full Linux file systems permissions are active. If SELinux were disabled, the client would be able to mount this share and have full access over it. However, as the nfs_export_all_rw Boolean is disabled, the client is currently not able to mount this file system, as shown below. This step should be performed on the client, nfs-client:

```
[nfs-client]# mkdir /myshare
[nfs-client]# mount.nfs 192.168.1.1:/myshare /myshare
mount.nfs: access denied by server while mounting 192.168.1.1:/myshare/
```

Enable the SELinux Boolean that was disabled in Step 1 above, and the client will be able to successfully mount the shared file system. This step should be performed on the NFS server, nfs-srv:

```
[nfs-srv]# setsebool -P nfs_export_all_rw on
```

Now try to mount the NFS file system again. This step should be performed on the NFS client, `nfs-client`:

```
[nfs-client]# mount.nfs 192.168.1.1:/myshare /myshare
[nfs-client]#
[nfs-client]# ls /myshare
total 0
-rwxrwxrwx.  1 root root 0 2009-04-16 12:07 file1
[nfs-client]#
```

The file system has been mounted successfully by the client. This example demonstrates how SELinux adds another layer of protection and can still enforce SELinux permissions even when Linux permissions are set to give full rights to all users.

10.7. Berkeley Internet Name Domain

BIND performs name resolution services via the `named` daemon. BIND lets users locate computer resources and services by name instead of numerical addresses.

In Fedora, the *bind* package provides a DNS server. Run `rpm -q bind` to see if the *bind* package is installed. If it is not installed and you want to use BIND, run the following command as the root user to install it:

```
yum install bind
```

10.7.1. BIND and SELinux

The default permissions on the `/var/named/slaves`,`/var/named/dynamic` and `/var/named/data` directories allow zone files to be updated via zone transfers and dynamic DNS updates. Files in `/var/named` are labeled with the `name_zone_t` type, which is used for master zone files.

For a slave server, configure `/etc/named.conf` to place slave zones in `/var/named/slaves`. The following is an example of a domain entry in `/etc/named.conf` for a slave DNS server that stores the zone file for `testdomain.com` in `/var/named/slaves`:

```
zone "testdomain.com" {
   type slave;
   masters { IP-address; };
   file "/var/named/slaves/db.testdomain.com";
         };
```

If a zone file is labeled `name_zone_t`, the `named_write_master_zones` Boolean must be enabled to allow zone transfers and dynamic DNS to update the zone file. Also, the mode of the parent directory has to be changed to allow the `named` user or group read, write and execue access.

If zone files in `/var/named/` are labeled with `name_cache_t` type, a file system relabel or running `restorecon -R /var/` will change their type to `named_zone_t`.

10.7.2. Types

The following types are used with BIND. Different types allow you to configure flexible access:

named_zone_t
: Used for master zone files. Other services can not modify files of this type. named can only modify files of this type if the named_write_master_zones Boolean is turned on.

named_cache_t
: By default, named can write to files labeled with this type, without additional Booleans being set. Files copied or created in the /var/named/slaves,/var/named/dynamic and /var/named/data directories are automatically labeled with the named_cache_t type.

10.7.3. Booleans

SELinux is based on the least level of access required for a service to run. Services can be run in a variety of ways; therefore, you must tell SELinux how you are running services. The following Booleans allow you to tell SELinux how you are running NFS:

named_write_master_zones
: When disabled, this Boolean prevents named from writing to zone files or directories labeled with the named_zone_t type. named does not usually need to write to zone files; but in the case that it needs to, or if a secondary server needs to write to zone files, enable this Boolean to allow this action.

10.7.4. Configuration Examples

10.7.4.1. Dynamic DNS

BIND allows hosts to update their records in DNS and zone files dynamically. This is used when a host computer's IP address changes frequently and the DNS record requires real-time modification.

Use the /var/named/dynamic directory for zone files you want updated via dynamic DNS. Files created in or copied into /var/named/dynamic inherit Linux permissions that allow named to write to them. As such files are labeled with the named_cache_t type, SELinux allows named to write to them.

If a zone file in /var/named/dynamic is labeled with the named_zone_t type, dynamic DNS updates may not be successful for a certain period of time as the update needs to be written to a journal first before being merged. If the zone file is labeled with the named_zone_t type when the journal attempts to be merged, an error such as the following is logged to /var/log/messages:

```
named[PID]: dumping master file: rename: /var/named/dynamic/zone-name: permission denied
```

Also, the following SELinux denial is logged to /var/log/messages:

```
setroubleshoot: SELinux is preventing named (named_t) "unlink" to zone-name (named_zone_t)
```

To resolve this labeling issue, run the restorecon -R -v /var/named/dynamic command as the Linux root user.

10.8. Concurrent Versioning System

The Concurrent Versioning System (CVS) is a free revision-control system. It is used to monitor and keep track of modifications to a central set of files which are usually accessed by several

different users. It is commonly used by programmers to manage a source code repository and is widely used by open source programmers.

In Fedora, the *cvs* package provides CVS. Run `rpm -q cvs` to see if the *cvs* package is installed. If it is not installed and you want to use CVS, run the following command as the root user to install it:

```
yum install cvs
```

10.8.1. CVS and SELinux

The `cvs` daemon runs as `cvs_t`. By default in Fedora, CVS is only allowed to read and write certain directories. The label `cvs_data_t` defines which areas the `cvs` daemon has read and write access to. When using CVS with SELinux, assigning the correct label is essential for clients to have full access to the area reserved for CVS data.

10.8.2. Types

The following types are used with CVS. Different types allow you to configure flexible access:

`cvs_data_t`
: This type is used for data in a CVS repository. CVS can only gain full access to data if it has this type.

`cvs_exec_t`
: This type is used for the `/usr/bin/cvs` binary.

10.8.3. Booleans

SELinux is based on the least level of access required for a service to run. Services can be run in a variety of ways; therefore, you must tell SELinux how you are running services. The following Boolean allows you to tell SELinux how you are running CVS:

`allow_cvs_read_shadow`
: This Boolean allows the `cvs` daemon to access the `/etc/shadow` file for user authentication.

10.8.4. Configuration Examples

10.8.4.1. Setting up CVS

This example describes a simple CVS setup and an SELinux configuration which allows remote access. Two hosts are used in this example; a CVS server with a hostname of `cvs-srv` with an IP address of `192.168.1.1` and a client with a hostname of `cvs-client` and an IP address of `192.168.1.100`. Both hosts are on the same subnet (192.168.1.0/24). This is an example only and assumes that the *cvs* and *xinetd* packages are installed, that the SELinux targeted policy is used, and that SELinux is running in enforced mode.

This example will show that even with full DAC permissions, SELinux can still enforce policy rules based on file labels and only allow access to certain areas that have been specifically labeled for access by CVS.

10.8.4.2. Server setup

注記

Steps 1-9 should be performed on the CVS server, `cvs-srv`.

1. As the root user, install the *cvs* and *xinetd* packages. Run `rpm -q cvs` to see if the *cvs* package is installed. If it is not installed, run `yum install cvs` as the root user to install it. Run `rpm -q xinetd` to see if the *xinetd* package is installed. If it is not installed, run `yum install xinetd` as the root user to install it.

2. Create a group named `CVS`. This can be done via the `groupadd CVS` command as the root user, or by using the *system-config-users* tool.

3. Create a user with a username of `cvsuser` and make this user a member of the CVS group. This can be done using the *system-config-users* tool.

4. Edit the `/etc/services` file and make sure that the CVS server has uncommented entries looking similar to the following:

```
cvspserver 2401/tcp   # CVS client/server operations
cvspserver 2401/udp   # CVS client/server operations
```

5. Create the CVS repository in the root area of the file system. When using SELinux, it is best to have the repository in the root file system so that recursive labels can be given to it without affecting any other subdirectories. For example, as the root user, create a `/cvs` directory to house the repository:

```
[root@cvs-srv]# mkdir /cvs
```

6. Give full permissions to the `/cvs` directory to all users:

```
[root@cvs-srv]# chmod -R 777 /cvs
```

警告

This is an example only and these permissions should not be used in a production system.

7. Edit the `/etc/xinetd.d/cvs` file and make sure that the CVS section is uncommented and configured to use the `/cvs` directory. The file should look similar to:

```
service cvspserver
{
 disable = no
 port   = 2401
 socket_type  = stream
 protocol  = tcp
 wait   = no
 user   = root
 passenv   = PATH
 server   = /usr/bin/cvs
 env   = HOME=/cvs
 server_args  = -f --allow-root=/cvs pserver
# bind   = 127.0.0.1
```

8. Start the `xinetd` daemon by running `service xinetd start` as the root user.

9. Add a rule which allows inbound connections using TCP on port 2401 by using the *system-config-firewall* tool.

10. As the `cvsuser` user, run the following command:

```
[cvsuser@cvs-client]$ cvs -d /cvs init
```

11. At this point, CVS has been configured but SELinux will still deny logins and file access. To demonstrate this, set the $CVSROOT variable on `cvs-client` and try to log in remotely. The following step should be performed on `cvs-client`:

```
[cvsuser@cvs-client]$ export CVSROOT=:pserver:cvsuser@192.168.1.1:/cvs
[cvsuser@cvs-client]$
[cvsuser@cvs-client]$ cvs login
Logging in to :pserver:cvsuser@192.168.1.1:2401/cvs
CVS password: ********
cvs [login aborted]: unrecognized auth response from 192.168.100.1: cvs pserver: cannot open /cvs/CVSROOT/
config: Permission denied
```

SELinux has blocked access. In order to get SELinux to allow this access, the following step should be performed on `cvs-srv`:

12. Change the context of the `/cvs` directory as the root user in order to recursively label any existing and new data in the `/cvs` directory, giving it the `cvs_data_t` type:

```
[root@cvs-srv]# semanage fcontext -a -t cvs_data_t '/cvs(/.*)?'
[root@cvs-srv]# restorecon -R -v /cvs
```

13. The client, `cvs-client` should now be able to log in and access all CVS resources in this repository:

```
[cvsuser@cvs-client]$ export CVSROOT=:pserver:cvsuser@192.168.1.1:/cvs
[cvsuser@cvs-client]$
[cvsuser@cvs-client]$ cvs login
Logging in to :pserver:cvsuser@192.168.1.1:2401/cvs
CVS password: ********
[cvsuser@cvs-client]$
```

10.9. Squid Caching Proxy

From the *Squid Caching Proxy*[17] project page:

"Squid is a caching proxy for the Web supporting HTTP, HTTPS, FTP, and more. It reduces bandwidth and improves response times by caching and reusing frequently-requested web pages. Squid has extensive access controls and makes a great server accelerator."

In Fedora, the *squid* package provides the Squid Caching Proxy. Run `rpm -q squid` to see if the *squid* package is installed. If it is not installed and you want to use squid, run the following command as the root user to install it:

```
# yum install squid
```

10.9.1. Squid Caching Proxy and SELinux

When SELinux is enabled, squid runs confined by default. Confined processes run in their own domains, and are separated from other confined processes. If a confined process is compromised by an attacker, depending on SELinux policy configuration, an attacker's access to resources and the possible damage they can do is limited. The following example demonstrates the squid processes running in their own domain. This example assumes the squid package is installed:

1. Run `getenforce` to confirm SELinux is running in enforcing mode:

   ```
   $ getenforce
   Enforcing
   ```

   The `getenforce` command returns `Enforcing` when SELinux is running in enforcing mode.

2. Run `service squid start` as the root user to start `squid`:

   ```
   # service squid start
   Starting squid:                                    [  OK  ]
   ```

3. Run `ps -eZ | grep squid` to view the `squid` processes:

   ```
   $ ps -eZ | grep squid
   unconfined_u:system_r:squid_t:s0 2522 ?        00:00:00 squid
   unconfined_u:system_r:squid_t:s0 2524 ?        00:00:00 squid
   unconfined_u:system_r:squid_t:s0 2526 ?        00:00:00 ncsa_auth
   unconfined_u:system_r:squid_t:s0 2527 ?        00:00:00 ncsa_auth
   unconfined_u:system_r:squid_t:s0 2528 ?        00:00:00 ncsa_auth
   unconfined_u:system_r:squid_t:s0 2529 ?        00:00:00 ncsa_auth
   unconfined_u:system_r:squid_t:s0 2530 ?        00:00:00 ncsa_auth
   unconfined_u:system_r:squid_t:s0 2531 ?        00:00:00 unlinkd
   ```

   The SELinux context associated with the `squid` processes is `unconfined_u:system_r:squid_t:s0`. The second last part of the context, `squid_t`, is the type. A

[17] http://www.squid-cache.org/

type defines a domain for processes and a type for files. In this case, the squid processes are running in the squid_t domain.

SELinux policy defines how processes running in confined domains, such as squid_t, interact with files, other processes, and the system in general. Files must be labeled correctly to allow squid access to them.

When /etc/squid/squid.conf is configured so squid listens on a port other than the default TCP ports 3128, 3401 or 4827, the semanage port command must be used to add the required port number to the SELinux policy configuration. The following example demonstrates configuring squid to listen on a port that is not initially defined in SELinux policy configuration for squid, and, as a consequence, squid failing to start. This example also demonstrates how to then configure the SELinux system to allow squid to successfully listen on a non-standard port that is not already defined in the policy. This example assumes the *squid* package is installed. Run each command in the example as the root user:

1. Run service squid status to confirm squid is not running:

```
# service squid status
squid is stopped
```

   If the output differs, run service squid stop to stop the process:

```
# service squid stop
Stopping squid:                                            [  OK  ]
```

2. Run semanage port -l | grep -w squid_port_t to view the ports SELinux allows squid to listen on:

```
semanage port -l | grep -w -i squid_port_t
squid_port_t                   tcp      3128, 3401, 4827
squid_port_t                   udp      3401, 4827
```

3. Edit /etc/squid/squid.conf as the root user. Configure the http_port option so it lists a port that is not configured in SELinux policy configuration for squid. In this example, squid is configured to listen on port 10000:

```
# Squid normally listens to port 3128
http_port 10000
```

4. Run service squid start to start squid:

```
# service squid start
Starting squid: .....                                      [FAILED]
```

   An SELinux denial similar to the following is logged to /var/log/messages:

```
```

```
localhost setroubleshoot: SELinux is preventing the squid (squid_t) from binding to port 1000. For complete
 SELinux messages. run sealert -l 97136444-4497-4fff-a7a7-c4d8442db982
```

5. For SELinux to allow `squid` to listen on port 10000, as used in this example, the following command is required:

```
# semanage port -a -t squid_port_t -p tcp 10000
```

6. Run `service squid start` again to start `squid` and have it listen on the new port:

```
# service squid start
Starting squid:        [  OK  ]
```

7. Now that SELinux has been configured to allow `squid` to listen on a non-standard port (TCP 10000 in this example), `squid` starts successfully on this port.

10.9.2. Types

Type Enforcement is the main permission control used in SELinux targeted policy. All files and processes are labeled with a type: types define a domain for processes and a type for files. SELinux policy rules define how types access each other, whether it be a domain accessing a type, or a domain accessing another domain. Access is only allowed if a specific SELinux policy rule exists that allows it.

The following types are used with `squid`. Different types allow you to configure flexible access:

`httpd_squid_script_exec_t`
: This type is used for utilities such as `cachemgr.cgi`, which provides a variety of statistics about squid and its configuration.

`squid_cache_t`
: Use this type for data that is cached by squid, as defined by the `cache_dir` directive in `/etc/squid/squid.conf`. By default, files created in or copied into `/var/cache/squid` and `/var/spool/squid` are labeled with the `squid_cache_t` type. Files for the *squidGuard*[18] URL redirector plugin for `squid` created in or copied to `/var/squidGuard` are also labeled with the `squid_cache_t` type. Squid is only able to use files and directories that are labeled with this type for its cached data.

`squid_conf_t`
: This type is used for the directories and files that `squid` uses for its configuration. Existing files, or those created in or copied to `/etc/squid` and `/usr/share/squid` are labeled with this type, including error messages and icons.

`squid_exec_t`
: This type is used for the squid binary, `/usr/sbin/squid`.

`squid_log_t`
: This type is used for logs. Existing files, or those created in or copied to `/var/log/squid` or `/var/log/squidGuard` must be labeled with this type.

[18] http://www.squidguard.org/

`squid_initrc_exec_t`
: This type is used for the initialization file required to start `squid` which is located at `/etc/rc.d/init.d/squid`.

`squid_var_run_t`
: This type is used by files in `/var/run`, especially the process id (PID) named `/var/run/squid.pid` which is created by squid when it runs.

10.9.3. Booleans

SELinux is based on the least level of access required for a service to run. Services can be run in a variety of ways; therefore, you must tell SELinux how you are running services. The following Boolean allows you to tell SELinux how you are running Squid:

`squid_connect_any`
: When enabled, this Boolean permits squid to initiate a connection to a remote host on any port.

`squid_use_tproxy`
: When enabled, this Boolean allows Squid to run as a transparent proxy.

10.9.4. Configuration Examples

10.9.4.1. Squid Connecting to Non-Standard Ports

The following example provides a real-world demonstration of how SELinux complements Squid by enforcing the above Boolean and by default only allowing access to certain ports. This example will then demonstrate how to change the Boolean and show that access is then allowed.

Note that this is an example only and demonstrates how SELinux can affect a simple configuration of Squid. Comprehensive documentation of Squid is beyond the scope of this document. Refer to the official *Squid documentation*[19] for further details. This example assumes that the Squid host has two network interfaces, Internet access, and that any firewall has been configured to allow access on the internal interface using the default TCP port on which Squid listens (TCP 3128).

1. As the root user, install the *squid* package. Run `rpm -q squid` to see if the *squid* package is installed. If it is not installed, run `yum install squid` as the root user to install it.

2. Edit the main configuration file, `/etc/squid/squid.conf` and confirm that the `cache_dir` directive is uncommented and looks similar to the following:

   ```
   cache_dir ufs /var/spool/squid 100 16 256
   ```

   This line specifies the default settings for the `cache_dir` directive to be used in this example; it consists of the Squid storage format (ufs), the directory on the system where the cache resides (/var/spool/squid), the amount of disk space in megabytes to be used for the cache (100), and finally the number of first-level and second-level cache directories to be created (16 and 256 respectively).

[19] http://www.squid-cache.org/Doc/

3. In the same configuration file, make sure the `http_access allow localnet` directive is uncommented. This allows traffic from the `localnet` ACL which is automatically configured in a default installation of Squid on Fedora 13. It will allow client machines on any existing RFC1918 network to have access through the proxy, which is sufficient for this simple example.

4. In the same configuration file, make sure the `visible_hostname` directive is uncommented and is configured to the hostname of the machine. The value should be the fully qualified domain name (FQDN) of the host:

```
visible_hostname squid.example.com
```

5. As the root user, run `service squid start` to start `squid`. As this is the first time `squid` has started, this command will initialise the cache directories as specified above in the `cache_dir` directive and will then start the `squid` daemon. The output is as follows if `squid` starts successfully:

```
# /sbin/service squid start
init_cache_dir /var/spool/squid... Starting squid: .       [  OK  ]
```

6. Confirm that the `squid` process ID (PID) has started as a confined service, as seen here by the `squid_var_run_t` value:

```
# ls -lZ /var/run/squid.pid
-rw-r--r--. root squid unconfined_u:object_r:squid_var_run_t:s0 /var/run/squid.pid
```

7. At this point, a client machine connected to the `localnet` ACL configured earlier is successfully able to use the internal interface of this host as its proxy. This can be configured in the settings for all common web browsers, or system-wide. Squid is now listening on the default port of the target machine (TCP 3128), but the target machine will only allow outgoing connections to other services on the Internet via common ports. This is a policy defined by SELinux itself. SELinux will deny access to non-standard ports, as shown in the next step:

8. When a client makes a request using a non-standard port through the Squid proxy such as a website listening on TCP port 10000, a denial similar to the following is logged:

```
SELinux is preventing the squid daemon from connecting to network port 10000
```

9. To allow this access, the `squid_connect_any` Boolean must be modified, as it is disabled by default. To turn the `squid_connect_any` Boolean on, run the following command as the root user:

```
# setsebool -P squid_connect_any on
```

注記

Do not use the -P option if you do not want `setsebool` changes to persist across reboots.

10. The client will now be able to access non-standard ports on the Internet as Squid is now permitted to initiate connections to any port, on behalf of its clients.

10.10. MySQL

From the *MySQL*[20] project page:

"The MySQL® database has become the world's most popular open source database because of its consistent fast performance, high reliability and ease of use. It's used on every continent -- Yes, even Antarctica! -- by individual Web developers as well as many of the world's largest and fastest-growing organizations to save time and money powering their high-volume Web sites, business-critical systems and packaged software -- including industry leaders such as Yahoo!, Alcatel-Lucent, Google, Nokia, YouTube, and Zappos.com."

In Fedora, the *mysql-server* package provides MySQL. Run `rpm -q mysql-server` to see if the *mysql-server* package is installed. If it is not installed, run the following command as the root user to install it:

```
yum install mysql-server
```

10.10.1. MySQL and SELinux

When MySQL is enabled, it runs confined by default. Confined processes run in their own domains, and are separated from other confined processes. If a confined process is compromised by an attacker, depending on SELinux policy configuration, an attacker's access to resources and the possible damage they can do is limited. The following example demonstrates the MySQL processes running in their own domain. This example assumes the *mysql* package is installed:

1. Run `getenforce` to confirm SELinux is running in enforcing mode:

```
$ getenforce
Enforcing
```

   The `getenforce` command returns `Enforcing` when SELinux is running in enforcing mode.

2. Run `service mysqld start` as the root user to start `mysqld`:

[20] http://www.mysql.com/why-mysql/

```
# service mysqld start
Initializing MySQL database:  Installing MySQL system tables... [  OK  ]
Starting MySQL:                                            [  OK  ]
```

3. Run `ps -eZ | grep mysqld` to view the `mysqld` processes:

```
$ ps -eZ | grep mysqld
unconfined_u:system_r:mysqld_safe_t:s0 6035 pts/1 00:00:00 mysqld_safe
unconfined_u:system_r:mysqld_t:s0 6123 pts/1   00:00:00 mysqld
```

The SELinux context associated with the `mysqld` processes is `unconfined_u:system_r:mysqld_t:s0`. The second last part of the context, `mysqld_t`, is the type. A type defines a domain for processes and a type for files. In this case, the `mysqld` processes are running in the `mysqld_t` domain.

10.10.2. Types

Type Enforcement is the main permission control used in SELinux targeted policy. All files and processes are labeled with a type: types define a domain for processes and a type for files. SELinux policy rules define how types access each other, whether it be a domain accessing a type, or a domain accessing another domain. Access is only allowed if a specific SELinux policy rule exists that allows it.

The following types are used with `mysql`. Different types allow you to configure flexible access:

`mysqld_db_t`
: This type is used for the location of the MySQL database. In Fedora 12, the default location for the database is `/var/lib/mysql`, however this can be changed. If the location for the MySQL database is changed, the new location must be labeled with this type. Refer to the following example for instructions on how to change the default database location and how to label the new section appropriately.

`mysqld_etc_t`
: This type is used for the MySQL main configuration file `/etc/my.cnf` and any other configuration files in the `/etc/mysql` directory.

`mysqld_exec_t`
: This type is used for the `mysqld` binary located at `/usr/libexec/mysqld`, which is the default location for the MySQL binary on Fedora 12. Other systems may locate this binary at `/usr/sbin/mysqld` which should also be labeled with this type.

`mysqld_initrc_exec_t`
: This type is used for the initialization file for MySQL, located at `/etc/rc.d/init.d/mysqld` by default in Fedora 12.

`mysqld_log_t`
: Logs for MySQL need to be labeled with this type for proper operation. All log files in `/var/log/` matching the `mysql.*` wildcard must be labeled with this type.

`mysqld_var_run_t`
: This type is used by files in `/var/run/mysqld`, specifically the process id (PID) named `/var/run/mysqld/mysqld.pid` which is created by the `mysqld` daemon when it runs. This type is also used for related socket files such as `/var/lib/mysql/mysql.sock`. Files such as these must be labeled correctly for proper operation as a confined service.

10.10.3. Booleans

SELinux is based on the least level of access required for a service to run. Services can be run in a variety of ways; therefore, you must tell SELinux how you are running services. The following Boolean allows you to tell SELinux how you are running MySQL:

`exim_can_connect_db`
: When enabled, this Boolean allows the `exim` mailer to initiate connections to a database server.

`ftpd_connect_db`
: When enabled, this Boolean allows `ftp` daemons to initiate connections to a database server.

`httpd_can_network_connect_db`
: Enabling this Boolean is required for a web server to communicate with a database server.

10.10.4. Configuration Examples

10.10.4.1. MySQL Changing Database Location

When using Fedora 12, the default location for MySQL to store its database is `/var/lib/mysql`. This is where SELinux expects it to be by default, and hence this area is already labeled appropriately for you, using the `mysqld_db_t` type.

The area where the database is located can be changed depending on individual environment requirements or preferences, however it is important that SELinux is aware of this new location - that it is labeled accordingly. This example explains how to change the location of a MySQL database and then how to label the new location so that SELinux can still provide its protection mechanisms to the new area based on its contents.

Note that this is an example only and demonstrates how SELinux can affect MySQL. Comprehensive documentation of MySQL is beyond the scope of this document. Refer to the official *MySQL documentation*[21] for further details. This example assumes that the *mysql-server* package is installed and that there is a valid database in the default location of `/var/lib/mysql`.

1. Run `ls -lZ /var/lib/mysql` to view the SELinux context of the default database location for `mysql`:

```
# ls -lZ /var/lib/mysql
drwx------. mysql mysql unconfined_u:object_r:mysqld_db_t:s0 mysql
```

   This shows `mysqld_db_t` which is the default context element for the location of database files. This context will have to be manually applied to the new database location that will be used in this example in order for it to function properly.

2. Enter `mysqlshow -u root -p` and enter the `mysqld` root password to show the available databases:

```
# mysqlshow -u root -p
Enter password: *******
+--------------------+
```

[21] http://dev.mysql.com/doc/

```
|     Databases      |
+--------------------+
| information_schema |
| mysql              |
| test               |
| wikidb             |
+--------------------+
```

3. Shut down the `mysqld` daemon with `service mysqld stop` as the root user:

```
# service mysqld stop
Stopping MySQL:                                            [  OK  ]
```

4. Create a new directory for the new location of the database(s). In this example, `/opt/mysql` is used:

```
# mkdir -p /opt/mysql
```

5. Copy the database files from the old location to the new location:

```
# cp -R /var/lib/mysql/* /opt/mysql/
```

6. Change the ownership of this location to allow access by the mysql user and group. This sets the traditional Unix permissions which SELinux will still observe.

```
# chown -R mysql:mysql /opt/mysql
```

7. Run `ls -lZ /opt` to see the initial context of the new directory:

```
# ls -lZ /opt
drwxr-xr-x. mysql mysql unconfined_u:object_r:usr_t:s0   mysql
```

   The context `usr_t` of this newly created directory is not currently suitable to SELinux as a location for MySQL database files. Once the context has been changed, MySQL will be able to function properly in this area.

8. Open the main MySQL configuration file `/etc/my.cnf` with a text editor and modify the `datadir` option so that it refers to the new location. In this example the value that should be entered is `/opt/mysql`.

```
[mysqld]
datadir=/opt/mysql
```

   Save this file and exit.

9. Run `service mysqld start` as the root user to start `mysqld`. At this point a denial will be logged to `/var/log/messages`:

```
# service mysqld start
```

```
Timeout error occurred trying to start MySQL Daemon.
Starting MySQL:                                            [FAILED]

# tail -f /var/log/messages

localhost setroubleshoot: SELinux is preventing mysqld (mysqld_t) "write" usr_t. For complete SELinux
 messages. run sealert -l 50d8e725-994b-499c-9caf-a676c50fb802
```

The reason for this denial is that `/opt/mysql` is not labeled correctly for MySQL data files. SELinux is stopping MySQL from having access to the content labeled as `usr_t`. Perform the following steps to resolve this problem:

10. Run the `semanage` command to add a context mapping for `/opt/mysql`:

```
semanage fcontext -a -t mysqld_db_t "/opt/mysql(/.*)?"
```

11. This mapping is written to the `/etc/selinux/targeted/contexts/files/file_contexts.local` file:

```
# grep -i mysql /etc/selinux/targeted/contexts/files/file_contexts.local

/opt/mysql(/.*)?    system_u:object_r:mysqld_db_t:s0
```

12. Now use the `restorecon` command to apply this context mapping to the running system:

```
restorecon -R -v /opt/mysql
```

13. Now that the `/opt/mysql` location has been labeled with the correct context for MySQL, the `mysqld` daemon starts:

```
# service mysqld start
Starting MySQL:                                            [  OK  ]
```

14. Confirm the context has changed for `/opt/mysql`:

```
ls -lZ /opt
drwxr-xr-x. mysql mysql system_u:object_r:mysqld_db_t:s0 mysql
```

15. The location has been changed and labeled, and the `mysqld` daemon has started successfully. At this point all running services should be tested to confirm normal operation.

10.11. PostgreSQL

From the *PostgreSQL*[22] project page:

[22] http://www.postgresql.org/about/

"PostgreSQL is a powerful, open source object-relational database system. It has more than 15 years of active development and a proven architecture that has earned it a strong reputation for reliability, data integrity, and correctness."

In Fedora, the *postgresql-server* package provides PostgreSQL. Run `rpm -q postgresql-server` to see if the *postgresql-server* package is installed. If it is not installed, run the following command as the root user to install it:

```
yum install postgresql-server
```

10.11.1. PostgreSQL and SELinux

When PostgreSQL is enabled, it runs confined by default. Confined processes run in their own domains, and are separated from other confined processes. If a confined process is compromised by an attacker, depending on SELinux policy configuration, an attacker's access to resources and the possible damage they can do is limited. The following example demonstrates the PostgreSQL processes running in their own domain. This example assumes the *postgresql-server* package is installed:

1. Run `getenforce` to confirm SELinux is running in enforcing mode:

   ```
   $ getenforce
   Enforcing
   ```

   The `getenforce` command returns `Enforcing` when SELinux is running in enforcing mode.

2. Run `service postgresql start` as the root user to start `postgresql`:

   ```
   service postgresql start
   Starting postgresql service:                               [  OK  ]
   ```

3. Run `ps -eZ | grep postgres` to view the `postgresql` processes:

   ```
   ps -eZ | grep postgres
   unconfined_u:system_r:postgresql_t:s0 395 ?    00:00:00 postmaster
   unconfined_u:system_r:postgresql_t:s0 397 ?    00:00:00 postmaster
   unconfined_u:system_r:postgresql_t:s0 399 ?    00:00:00 postmaster
   unconfined_u:system_r:postgresql_t:s0 400 ?    00:00:00 postmaster
   unconfined_u:system_r:postgresql_t:s0 401 ?    00:00:00 postmaster
   unconfined_u:system_r:postgresql_t:s0 402 ?    00:00:00 postmaster
   ```

   The SELinux context associated with the `postgresql` processes is `unconfined_u:system_r:postgresql_t:s0`. The second last part of the context, `postgresql_t`, is the type. A type defines a domain for processes and a type for files. In this case, the `postgresql` processes are running in the `postgresql_t` domain.

10.11.2. Types

Type Enforcement is the main permission control used in SELinux targeted policy. All files and processes are labeled with a type: types define a domain for processes and a type for files. SELinux policy rules define how types access each other, whether it be a domain accessing a

type, or a domain accessing another domain. Access is only allowed if a specific SELinux policy rule exists that allows it.

The following types are used with `postgresql`. Different types allow you to configure flexible access:

`postgresql_db_t`
: This type is used for several locations. The locations labeled with this type are used for data files for PostgreSQL:
  - `/usr/lib/pgsql/test/regres`
  - `/usr/share/jonas/pgsql`
  - `/var/lib/pgsql/data`
  - `/var/lib/postgres(ql)?`

`postgresql_etc_t`
: This type is used for configuration files in `/etc/postgresql`.

`postgresql_exec_t`
: This type is used for several locations. The locations labeled with this type are used for binaries for PostgreSQL:
  - `/usr/bin/initdb(.sepgsql)?`
  - `/usr/bin/(se)?postgres`
  - `/usr/lib(64)?/postgresql/bin/.*`
  - `/usr/lib/phsql/test/regress/pg_regress`

`postgresql_initrc_exec_t`
: This type is used for the PostgreSQL initialization file located at `/etc/rc.d/init.d/postgresql`.

`postgresql_log_t`
: This type is used for several locations. The locations labeled with this type are used for log files:
  - `/var/lib/pgsql/logfile`
  - `/var/lib/pgsql/pgstartup.log`
  - `/var/lib/sepgsql/pgstartup.log`
  - `/var/log/postgresql`
  - `/var/log/postgres.log.*`
  - `/var/log/rhdb/rhdb`
  - `/var/log/sepostgresql.log.*`

`postgresql_var_run_t`
: This type is used for run-time files for PostgreSQL, such as the process id (PID) in `/var/run/postgresql`.

10.11.3. Booleans

SELinux is based on the least level of access required for a service to run. Services can be run in a variety of ways; therefore, you must tell SELinux how you are running services. The following Boolean allows you to tell SELinux how you are running PostgreSQL:

`allow_user_postgresql_connect`
: Having this Boolean enabled allows any user domain (as defined by PostgreSQL) to make connections to the database server.

10.11.4. Configuration Examples

10.11.4.1. PostgreSQL Changing Database Location

When using Fedora 12, the default location for PostgreSQL to store its database is `/var/lib/pgsql/data`. This is where SELinux expects it to be by default, and hence this area is already labeled appropriately for you, using the `postgresql_db_t` type.

The area where the database is located can be changed depending on individual environment requirements or preferences, however it is important that SELinux is aware of this new location - that it is labeled accordingly. This example explains how to change the location of a PostgreSQL database and then how to label the new location so that SELinux can still provide its protection mechanisms to the new area based on its contents.

Note that this is an example only and demonstrates how SELinux can affect PostgreSQL. Comprehensive documentation of PostgreSQL is beyond the scope of this document. Refer to the official *PostgreSQL documentation*[23] for further details. This example assumes that the *postgresql-server* package is installed.

1. Run `ls -lZ /var/lib/pgsql` to view the SELinux context of the default database location for `postgresql`:

   ```
   # ls -lZ /var/lib/pgsql
   drwx------. postgres postgres system_u:object_r:postgresql_db_t:s0 data
   ```

   This shows `postgresql_db_t` which is the default context element for the location of database files. This context will have to be manually applied to the new database location that will be used in this example in order for it to function properly.

2. Create a new directory for the new location of the database(s). In this example, `/opt/postgresql/data` is used. If you use a different location, replace the text in the following steps with your location:

   ```
   # mkdir -p /opt/postgresql/data
   ```

3. Perform a directory listing of the new location. Note that the initial context of the new directory is *usr_t*. This context is not sufficient for SELinux to offer its protection mechanisms to PostgreSQL. Once the context has been changed, it will be able to function properly in the new area.

[23] http://www.postgresql.org/docs/

```
# ls -lZ /opt/postgresql/
drwxr-xr-x. root root unconfined_u:object_r:usr_t:s0   data
```

4. Change the ownership of the new location to allow access by the postgres user and group. This sets the traditional Unix permissions which SELinux will still observe.

```
# chown -R postgres:postgres /opt/postgresql
```

5. Open the PostgreSQL init file `/etc/rc.d/init.d/postgresql` with a text editor and modify all `PGDATA` and `PGLOG` variables to point to the new location:

```
# vi /etc/rc.d/init.d/postgresql
PGDATA=/opt/postgresql/data
PGLOG=/opt/postgresql/data/pgstartup.log
```

   Save this file and exit the text editor.

6. Initialize the database in the new location.

```
su - postgres -c "initdb -D /opt/postgresql/data"
```

7. Run the `semanage` command to add a context mapping for `/opt/postgresql` and any other directories/files within it:

```
semanage fcontext -a -t postgresql_db_t "/opt/postgresql(/.*)?"
```

8. This mapping is written to the `/etc/selinux/targeted/contexts/files/file_contexts.local` file:

```
# grep -i postgresql /etc/selinux/targeted/contexts/files/file_contexts.local

/opt/postgresql(/.*)?    system_u:object_r:postgresql_db_t:s0
```

9. Now use the `restorecon` command to apply this context mapping to the running system:

```
restorecon -R -v /opt/postgresql
```

10. Now that the `/opt/postgresql` location has been labeled with the correct context for PostgreSQL, the `mysqld` service will start successfully:

```
# service postgresql start
Starting postgreSQL service:                               [  OK  ]
```

11. Confirm the context is correct for `/opt/postgresql`:

```
ls -lZ /opt
drwxr-xr-x. root root system_u:object_r:postgresql_db_t:s0 postgresql
```

12. Check with the ps command that the postgresql process displays the new location:

```
# ps aux | grep -i postmaster

postgres 21564  0.3  0.3  42308  4032 ?        S    10:13   0:00 /usr/bin/postmaster -p 5432 -D /opt/
postgresql/data
```

13. The location has been changed and labeled, and the postgresql daemon has started successfully. At this point all running services should be tested to confirm normal operation.

10.12. rsync

From the *Rsync*[24] project page:

"rsync is an open source utility that provides fast incremental file transfer."

When using Fedora, the *rsync* package provides rsync. Run rpm -q rsync to see if the *rsync* package is installed. If it is not installed, run the following command as the root user to install it:

```
yum install rsync
```

10.12.1. rsync and SELinux

From the Fedora 12 SELinux *rsync_selinux(8)* man page: "SELinux requires files to have an extended attribute to define the file type. Policy governs the access daemons have to these files. If you want to share files using the rsync daemon, you must label the files and directories *public_content_t*."

Like most services, correct labeling is required for SELinux to perform its protection mechanisms over rsync.

10.12.2. Types

Type Enforcement is the main permission control used in SELinux targeted policy. All files and processes are labeled with a type: types define a domain for processes and a type for files. SELinux policy rules define how types access each other, whether it be a domain accessing a type, or a domain accessing another domain. Access is only allowed if a specific SELinux policy rule exists that allows it.

The following types are used with rsync. Different types all you to configure flexible access:

public_content_t
: This is a generic type used for the location of files (and the actual files) to be shared via rsync. If a special directory is created to house files to be shared with rsync, the directory and its contents need to have this label applied to them.

[24] http://www.samba.org/rsync/

`rsync_exec_t`
: This type is used for the `/usr/bin/rsync` system binary.

`rsync_log_t`
: This type is used for the `rsync` log file, located at `/var/log/rsync.log` by default. To change the location of the file rsync logs to, use the `--log-file=FILE` option to the `rsync` command at run-time.

`rsync_var_run_t`
: This type is used for the `rsyncd` lock file, located at `/var/run/rsyncd.lock`. This lock file is used by the `rsync` server to manage connection limits.

10.12.3. Booleans

SELinux is based on the least level of access required for a service to run. Services can be run in a variety of ways; therefore, you must tell SELinux how you are running services. The following Boolean allows you to tell SELinux how you are running rsync:

`allow_rsync_anon_write`
: Having this Boolean enabled allows `rsync` in the *rsync_t* domain to manage files, links and directories that have a type of *public_content_rw_t*. Often these are public files used for public file transfer services. Files and directories must be labeled `public_content_rw_t`.

`rsync_client`
: Having this Boolean enabled aloows `rsync` to initiate connections to ports defined as *rsync_port_t*, as well as allowing `rsync` to manage files, links and directories that have a type of *rsync_data_t*. Note that the `rsync` daemon must be in the *rsync_t* domain in order for SELinux to enact its control over `rsync`. The configuration example in this chapter demonstrates `rsync` running in the *rsync_t* domain.

`rsync_export_all_ro`
: Having this Boolean enabled allows `rsync` in the *rsync_t* domain to export NFS and CIFS file systems with read-only access to clients.

10.12.4. Configuration Examples

10.12.4.1. Rsync as a daemon

When using Fedora, rsync can be used as a daemon so that multiple clients can directly communicate with it as a central server, in order to house centralized files and keep them synchronized. The following example will demonstrate running rsync as a daemon over a network socket in the correct domain, and how SELinux expects this daemon to be running on a pre-defined (in SELinux policy) TCP port. This example will then show how to modify SELinux policy to allow the `rsync` daemon to run normally on a non-standard port.

This example will be performed on a single system to demonstrate SELinux policy and its control over local daemons and processes. Note that this is an example only and demonstrates how SELinux can affect rsync. Comprehensive documentation of rsync is beyond the scope of this document. Refer to the official *rsync documentation*[25] for further details. This example assumes that the *rsync*, *setroubleshoot-server* and *audit* packages are installed, that the SELinux targeted policy is used and that SELinux is running in enforcing mode.

[25] http://www.samba.org/rsync/documentation.html

Getting rsync to launch as rsync_t

1. Run `getenforce` to confirm SELinux is running in enforcing mode:

```
$ getenforce
Enforcing
```

   The `getenforce` command returns `Enforcing` when SELinux is running in enforcing mode.

2. Run the `which` command to confirm that the rsync binary is in the system path:

```
$ which rsync
/usr/bin/rsync
```

3. When running `rsync` as a daemon, a configuration file should be used and saved as `/etc/rsyncd.conf`. Note that the following configuration file used in this example is very simple and is not indicative of all the possible options that are available, rather it is just enough to demonstrate the `rsync` daemon:

```
log file = /var/log/rsyncd.log
pid file = /var/run/rsyncd.pid
lock file = /var/run/rsync.lock
[files]
        path = /srv/files
        comment = file area
        read only = false
 timeout = 300
```

4. Now that a simple configuration file exists for rsync to operate in daemon mode, this step demonstrates that simply running `rsync --daemon` is not sufficient for SELinux to offer its protection over rsync. Refer to the following output:

```
# rsync --daemon

# ps x | grep rsync
 8231 ?        Ss     0:00 rsync --daemon
 8233 pts/3    S+     0:00 grep rsync

# ps -eZ | grep rsync
unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023 8231 ? 00:00:00 rsync
```

   Note that in the output from the final `ps` command, the context shows the `rsync` daemon running in the `unconfined_t` domain. This indicates that rsync has not transitioned to the `rsync_t` domain as it was launched by the `rsync --daemon` command. At this point SELinux can not enforce its rules and policy over this daemon. Refer to the following steps to see how to fix this problem. In the following steps, `rsync` will transition to the `rsync_t` domain by launching it from a properly-labeled init script. Only then can SELinux and its protection mechanisms have an effect over `rsync`. This `rsync` process should be killed before proceeding to the next step.

5. A custom init script for rsync is needed for this step. There is an example init script available at *http://www.fredshack.com/docs/rsync.html*. Save it to `/etc/rc.d/init.d/rsyncd`. The following steps show how to label this script as `initrc_exec_t`:

6. Run the semanage command to add a context mapping for /etc/rc.d/init.d/rsyncd:

```
semanage fcontext -a -t initrc_exec_t "/etc/rc.d/init.d/rsyncd"
```

7. This mapping is written to the /etc/selinux/targeted/contexts/files/file_contexts.local file:

```
# grep rsync /etc/selinux/targeted/contexts/files/file_contexts.local

/etc/rc.d/init.d/rsyncd    system_u:object_r:initrc_exec_t:s0
```

8. Now use the restorecon command to apply this context mapping to the running system:

```
restorecon -R -v /etc/rc.d/init.d/rsyncd
```

9. Run the ls to confirm the script has been labeled appropriately. Note that in the following output the script has been labeled as initrc_exec_t:

```
 ls -lZ /etc/rc.d/init.d/rsyncd
-rwxr-xr-x. root root system_u:object_r:initrc_exec_t:s0 /etc/rc.d/init.d/rsyncd
```

10. Launch rsyncd via the new script. Now that rsync has started from an init script that has been appropriately labeled, the process will start as rsync_t:

```
# /etc/rc.d/init.d/rsync start
Starting rsyncd:                                           [  OK  ]

ps -eZ | grep rsync
unconfined_u:system_r:rsync_t:s0 9794 ?        00:00:00 rsync
```

    SELinux can now enforce its protection mechanisms over the rsync daemon as it is now runing in the rsync_t domain.

This example demonstrated how to get rsyncd running in the rsync_t domain. The next example shows how to get this daemon successfully running on a non-default port. TCP port 10000 is used in the next example.

Running the rsync daemon on a non-default port

1. Modify the /etc/rsyncd.conf file and add the port = 10000 line at the top of the file in the global configuration area (ie., before any file areas are defined). The new configuration file will look like:

```
log file = /var/log/rsyncd.log
pid file = /var/run/rsyncd.pid
lock file = /var/run/rsync.lock
port = 10000
[files]
        path = /srv/files
        comment = file area
        read only = false
 timeout = 300
```

2. After launching rsync from the init script with this new setting, a denial similar to the following is logged by SELinux:

```
Jul 22 10:46:59 localhost setroubleshoot: SELinux is preventing the rsync (rsync_t) from binding to port
 10000. For complete SELinux messages. run sealert -l c371ab34-639e-45ae-9e42-18855b5c2de8
```

3. Run the `semanage` command to add TCP port 10000 to SELinux policy in `rsync_port_t`:

```
# semanage port -a -t rsync_port_t -p tcp 10000
```

4. Now that TCP port 10000 has been added to SELinux policy for `rsync_port_t`, `rsyncd` will start and operate normally on this port:

```
# /etc/rc.d/init.d/rsync start
Starting rsyncd:                                           [  OK  ]
```

```
# netstat -lnp | grep 10000
tcp        0      0 0.0.0.0:10000   0.0.0.0:*      LISTEN      9910/rsync
```

SELinux has had its policy modified and is now permitting `rsyncd` to operate on TCP port 10000.

10.13. Postfix

From the *Postfix*[26] project page:

"What is Postfix? It is Wietse Venema's mailer that started life at IBM research as an alternative to the widely-used Sendmail program. Postfix attempts to be fast, easy to administer, and secure. The outside has a definite Sendmail-ish flavor, but the inside is completely different."

In Fedora, the *postfix* package provides postfix. Run `rpm -q postfix` to see if the *postfix* package is installed. If it is not installed, run the following command as the root user to install it:

```
yum install postfix
```

10.13.1. Postfix and SELinux

When Postfix is enabled, it runs confined by default. Confined processes run in their own domains, and are separated from other confined processes. If a confined process is compromised by an attacker, depending on SELinux policy configuration, an attacker's access to resources and the possible damage they can do is limited. The following example demonstrates the Postfix and related processes running in their own domain. This example assumes the *postfix* package is installed and that the Postfix service has been started:

1. Run `getenforce` to confirm SELinux is running in enforcing mode:

```
```

[26] http://www.postfix.org/

```
$ getenforce
Enforcing
```

The `getenforce` command returns `Enforcing` when SELinux is running in enforcing mode.

2. Run `service postfix start` as the root user to start `postfix`:

```
service postfix start
Starting postfix:                          [  OK  ]
```

3. Run `ps -eZ | grep postfix` to view the `postfix` processes:

```
ps -eZ | grep postfix
system_u:system_r:postfix_master_t:s0 1651 ?   00:00:00 master
system_u:system_r:postfix_pickup_t:s0 1662 ?   00:00:00 pickup
system_u:system_r:postfix_qmgr_t:s0 1663 ?     00:00:00 qmgr
```

For example, the SELinux context associated with the Postfix `master` process is `unconfined_u:system_r:postfix_master_t:s0`. The second last part of the context, `postfix_master_t`, is the type for this process. A type defines a domain for processes and a type for files. In this case, the `master` process is running in the `postfix_master_t` domain.

10.13.2. Types

Type Enforcement is the main permission control used in SELinux targeted policy. All files and processes are labeled with a type: types define a domain for processes and a type for files. SELinux policy rules define how types access each other, whether it be a domain accessing a type, or a domain accessing another domain. Access is only allowed if a specific SELinux policy rule exists that allows it.

The following types are used with `Postfix`. Different types all you to configure flexible access:

`postfix_etc_t`
: This type is used for configuration files for Postfix in `/etc/postfix`.

`postfix_data_t`
: This type is used for Postfix data files in `/var/lib/postfix`.

注記

To see the full list of files and their types for Postfix, run the following command:

```
$ grep postfix /etc/selinux/targeted/contexts/files/file_contexts
```

10.13.3. Booleans

SELinux is based on the least level of access required for a service to run. Services can be run in a variety of ways; therefore, you must tell SELinux how you are running services. The following Boolean allows you to tell SELinux how you are running Postfix:

`allow_postfix_local_write_mail_spool`
: Having this Boolean enables Postfix to write to the local mail spool on the system. Postfix requires this Boolean to be enabled for normal operation when local spools are used.

10.13.4. Configuration Examples

10.13.4.1. SpamAssassin and Postfix

From the *SpamAssassin*[27] project page:

"Open Source mail filter, written in Perl, to identify spam using a wide range of heuristic tests on mail headers and body text. Free software."

When using Fedora, the *spamassassin* package provides SpamAssassin. Run `rpm -q spamassassin` to see if the *spamassassin* package is installed. If it is not installed, run the following command as the root user to install it:

```
yum install spamassassin
```

SpamAssassin operates in tandom with a mailer such as Postfix to provide spam-filtering capabilities. In order for SpamAssassin to effectively intercept, analyze and filter mail, it must listen on a network interface. The default port for SpamAssassin is TCP/783, however this can be changed. The following example provides a real-world demonstration of how SELinux complements SpamAssassin by only allowing it access to a certain port by default. This example will then demonstrate how to change the port and have SpamAssassin operate on a non-default port.

Note that this is an example only and demonstrates how SELinux can affect a simple configuration of SpamAssassin. Comprehensive documentation of SpamAssassin is beyond the scope of this document. Refer to the official *SpamAssassin documentation*[28] for further details. This example assumes the *spamassassin* is installed, that any firewall has been configured to allow access on the ports in use, that the SELinux targeted policy is used, and that SELinux is running in enforcing mode:

Running SpamAssassin on a non-default port

1. Run the `semanage` command to show the port that SELinux allows `spamd` to listen on by default:

   ```
   # semanage port -l | grep spamd
   spamd_port_t  tcp 783
   ```

   This output shows that TCP/783 is defined in `spamd_port_t` as the port for SpamAssassin to operate on.

2. Edit the `/etc/sysconfig/spamassassin` configuration file and modify it so that it will start SpamAssassin on the example port TCP/10000:

   ```
   # Options to spamd
   ```

[27] http://spamassassin.apache.org/
[28] http://spamassassin.apache.org/doc.html

```
SPAMDOPTIONS="-d -p 10000 -c m5 -H"
```

This line now specifies that SpamAssassin will operate on port 10000. The rest of this example will show how to modify SELinux policy to allow this socket to be opened.

3. Start SpamAssassin and an error message similar to the following will appear:

```
/etc/init.d/spamassassin start
Starting spamd: [2203] warn: server socket setup failed, retry 1: spamd: could not create INET socket on
 127.0.0.1:10000: Permission denied
[2203] warn: server socket setup failed, retry 2: spamd: could not create INET socket on 127.0.0.1:10000:
 Permission denied
[2203] error: spamd: could not create INET socket on 127.0.0.1:10000: Permission denied
spamd: could not create INET socket on 127.0.0.1:10000: Permission denied
                                                           [FAILED]
```

This output means that SELinux has blocked access to this port.

4. A denial similar to the following will be logged by SELinux:

```
SELinux is preventing the spamd (spamd_t) from binding to port 10000.
```

5. As the root user, run the semanage command to modify SELinux policy in order to allow SpamAssassin to operate on the example port (TCP/10000):

```
semanage port -a -t spamd_port_t -p tcp 10000
```

6. Confirm that SpamAssassin will now start and is operating on TCP port 10000:

```
# /etc/init.d/spamassassin start
Starting spamd:     [ OK ]

# netstat -lnp | grep 10000
tcp 0 0 127.0.0.1:10000 0.0.0.0:* LISTEN 2224/spamd.pid
```

7. At this point, spamd is properly operating on TCP port 10000 as it has been allowed access to that port by SELinux policy.

付録A 暗号の標準

A.1. 同期式の暗号

A.1.1. Advanced Encryption Standard - AES

暗号において、Advanced Encryption Standard (AES) はアメリカ政府によって採用された暗号標準です。この標準は、Rijndael として公開された元々のより大きなコレクションから採用された、3つのブロック暗号 AES-128, AES-192 および AES-256 から構成されます。各 AES 暗号は、それぞれキーの大きさ 128, 192 および 256 bit とともに 128-bit のブロックサイズをを持ちます。AES 暗号は詳細に分析されてきて、その前進である Data Encryption Standard (DES) と同様に、今では世界中で使用されています。[1]

A.1.1.1. AES の使用

A.1.1.2. AES の歴史

AES は、5年間の標準化プロセスの後、2001年11月26日に U.S. FIPS PUB 197 (FIPS 197) として National Institute of Standards and Technology (NIST) によりアナウンスされました。そこでは、Rijndael が最適であると選択される前に、15の競合する設計が提案され、評価されました。2002年5月26日に標準として有効になりました。多くの異なる暗号化パッケージにおいて利用可能です。AES は、初めて一般にアクセス可能であり、トップシークレット情報のために NSA により承認されたオープンな暗号です（以下にある AES のセキュリティを参照してください）。[2]

Rijndael は2人のベルギー人暗号学者 Joan Daemon と Vincent Rijmen により開発され、彼らにより AES 選定プロセスへ投稿されました。Rijndael ([r#inda:l] と発音) は発明者2人の名前のかばん語です。[3]

A.1.2. Data Encryption Standard - DES

Data Encryption Standard (DES) は、1976年にアメリカに対する公式な Federal Information Processing Standard (FIPS) として National Bureau of Standards により選択され、その後国際的に広く恩恵を受けている、ブロック暗号（共有秘密暗号の形式）です。56-bit 鍵を使用する対称鍵アルゴリズムに基づいています。アルゴリズムは当初、秘密の設計要素、相対的に短い鍵長および National Security Agency (NSA) のバックドアに関する疑惑とともに議論の的になりました。結果として、DES はブロック暗号と暗号解析の現代の知識に動機づけられた学術的な厳しい詳細な調査を受けました。[4]

A.1.2.1. DES の

A.1.2.2. DES の歴史

DESは今や多くのアプリケーションに対して安全ではないと考えられています。おもに 56-bit 鍵の大きさが小さすぎることによります; 1999年1月、distributed.net と Electronic Frontier Foundation は公に協力して、DES 鍵を22時間15分で解読しました（年表参照）。また、実際には実装できませんが、暗号において理論

[1] "Advanced Encryption Standard." *Wikipedia.* 14 November 2009 *http://en.wikipedia.org/wiki/Advanced_Encryption_Standard*
[2] "Advanced Encryption Standard." *Wikipedia.* 14 November 2009 *http://en.wikipedia.org/wiki/Advanced_Encryption_Standard*
[3] "Advanced Encryption Standard." *Wikipedia.* 14 November 2009 *http://en.wikipedia.org/wiki/Advanced_Encryption_Standard*
[4] "Data Encryption Standard." *Wikipedia.* 14 November 2009 *http://en.wikipedia.org/wiki/Data_Encryption_Standard*

的に弱いことが説明されるという、いくつかの解析的な結論があります。理論的な攻撃があるにも関わらず、アルゴリズムは 3-DES の形でほとんど安全であると考えられています。近年、暗号は Advanced Encryption Standard (AES) に置き換えられてきています。[5]

いくつかのドキュメントにおいて、標準としての DES と DEA (the Data Encryption Algorithm) として参照される DES アルゴリズムを区別しています。発音されるとき、"DES" は、省略形としてスペルされたものとしても (/ˌdiːˌiːˈɛs/) 、1音節の略語としても (/ˈdɛz/) 発音されます。[6]

A.2. 公開鍵暗号

公開鍵暗号は、多くの暗号アルゴリズムと暗号化システムにより採用されている、暗号的なアプローチです。その際立った特徴は、対象の鍵アルゴリズムの代わりに、またはそれに加えて、非対称の鍵アルゴリズムを使用することです。公開鍵-秘密鍵暗号の技術を使用することで、以前は知られていなかった、コミュニケーションや認証メッセージを保護する多くの方法が実用的になりました。対称鍵アルゴリズムを使うときに必要となるような、1つかそれより多い秘密鍵の始めの安全な交換が必要なくなりました。電子署名を作成するためにも使用されます。[7]

公開鍵暗号は、世界中で基本的かつ広範囲に使用される技術です。また、Transport Layer Security (TLS) (SSL の後継), PGP および GPG のようなインターネット標準として基礎となるアプローチです。[8]

公開鍵暗号において使用される特徴的な技術は非対称の鍵アルゴリズムの使用です。ここで、メッセージを暗号化するために使われる鍵は、復号するために使われる鍵を同じではありません。各ユーザーは、一組の暗号鍵— 公開鍵と秘密鍵を持ちます。公開鍵が広く配布されるかもしれないのに対して、秘密鍵は秘密にしておきます。メッセージは受信者の公開鍵で暗号化され、対応する秘密鍵でのみ復号することができます。鍵は数学的に関連していますが、秘密鍵は公開鍵からうまく導くことができません（つまり、実際のまたは計画された実践）。1970年代半ばに始まった暗号の実践の変革をもたらす、そのようなアルゴリズムを発見しました。[9]

対照的に、数千年の間使用されてきたバリエーションである、対称鍵暗号は、暗号化と復号のために送信者と受信者により共有される1つの秘密鍵（プライベートに保たなければいけない、このように共通の用語の曖昧さの原因であるもの）を使用します。対称の暗号化スキーマを使用するために、送信者と受信者が前もって安全に鍵を共有しなければいけません。[10]

対称鍵アルゴリズムがほとんど常に計算的に集約的であるので、鍵交換アルゴリズムを用いて鍵を交換して、その鍵と対称鍵アルゴリズムを用いてデータを転送します。たとえば、PGP、およびスキームの SSL/TLS ファミリーはこれをします。結果としてハイブリッド暗号システムと呼ばれます。[11]

A.2.1. Diffie-Hellman

Diffie–Hellman 鍵交換 (D–H) は、お互いに事前に知識を持たない2者が、安全ではないコミュニケーション・チャネル上で共有の秘密鍵を共同で確立できるようにする、暗号のプロトコルです。そして、この鍵は対称鍵暗号を用いて以降のコミュニケーションを暗号化するために使用されます。[12]

A.2.1.1. Diffie-Hellman の歴史

スキーマは1976年に Whitfield Diffie と Martin Hellman により初めて公開されました。しかしながら後から、GCHQ の British signals intelligence agency の中で Malcolm J. Williamson によりまったく別に数

[5] "Data Encryption Standard." *Wikipedia.* 14 November 2009 *http://en.wikipedia.org/wiki/Data_Encryption_Standard*
[6] "Data Encryption Standard." *Wikipedia.* 14 November 2009 *http://en.wikipedia.org/wiki/Data_Encryption_Standard*
[7] "Public-key Encryption." *Wikipedia.* 14 November 2009 *http://en.wikipedia.org/wiki/Public-key_cryptography*
[8] "Public-key Encryption." *Wikipedia.* 14 November 2009 *http://en.wikipedia.org/wiki/Public-key_cryptography*
[9] "Public-key Encryption." *Wikipedia.* 14 November 2009 *http://en.wikipedia.org/wiki/Public-key_cryptography*
[10] "Public-key Encryption." *Wikipedia.* 14 November 2009 *http://en.wikipedia.org/wiki/Public-key_cryptography*
[11] "Public-key Encryption." *Wikipedia.* 14 November 2009 *http://en.wikipedia.org/wiki/Public-key_cryptography*
[12] "Diffie-Hellman." *Wikipedia.* 14 November 2009 *http://en.wikipedia.org/wiki/Diffie-Hellman*

年早く発明されていたが、秘密にされていたことがわかりました。2002年、Hellman は、公開鍵暗号の発明に対する貢献を認めて Diffie–Hellman–Merkle 鍵交換と呼ばれるアルゴリズムを提案しました（Hellman, 2002）。[13]

Diffie–Hellman 鍵合意それ自身は、匿名の（認証されない）鍵合意プロトコルであるにも関わらず、いろいろな認証されたプロトコルに対する基礎を提供し、Transport Layer Security の超短期モード（暗号スイートに依存して EDH または DHE として参照されます）において、完全な順方向の秘密を提供するために使用されます。[14]

U.S. Patent 4,200,770 (現在、失効) は、アルゴリズムが説明されていて、発明者として Hellman, Diffie と Merkle がクレジットされています。[15]

A.2.2. RSA

暗号学において、RSA (初めて公的にそれを説明した Rivest, Shamir および Adleman を意味します。以下参照。) は公開鍵暗号のアルゴリズムです。それは、暗号と同様に署名にも適しているとして知られる最初のアルゴリズムで、公開鍵暗号において始めての大きな優位性の1つでした。RSA は、電子商取引のプロトコルにおいて広く使用され、十分に長い鍵が与えられ、更新の実装が使われていて、安全であると考えれらています。[16]

A.2.3. DSA

Digital Signature Algorithm (DSA) は電子署名に対する United States Federal Government standard または FIPS です。Digital Signature Standard (DSS) で使用するために、1991年8月に National Institute of Standards and Technology (NIST) により提案され、FIPS 186 で指定され、1993年に適用されました。わずかな改訂が FIPS 186-1 として1996年に発行されました。この標準は、さらに FIPS 186-2 として2000年に、再び FIPS 186-3 として2009年に、拡張されました。[17]

A.2.4. SSL/TLS

Transport Layer Security (TLS) とその前進である Secure Socket Layer (SSL) は、インターネットのようなネットワークにおけるコミュニケーションに対してセキュリティを提供する暗号プロトコルです。TLS と SSL は、エンドからエンドへのトランスポート層におけるネットワーク接続のセグメントを暗号化します。プロトコルのいくつかのバージョンは、ウェブ閲覧、電子メール、インターネット FAX、インスタント・メッセージおよび voice-over-IP (VoIP) のようなアプリケーションにおいて広く使われます。TLS は IETF 標準トラックプロトコルです。それは、Netscape 社により開発された以前の SSL 仕様に基づいた、RFC 5246 で最終更新されました。

TLS プロトコルは、クライアント/サーバーのアプリケーションが、盗聴や改ざんを防ぐために設計された方法で、ネットワークを越えたコミュニケーションできるようにします。TLS は暗号を用いてインターネット上でエンドポイント認証と通信の秘密を提供します。TLS は 1024 bit および 2048 bit 強度を持つ RSA セキュリティを提供します。

一般的なエンドユーザー/ブラウザの使い方において、TLS 認証は一方的です: サーバーのみが認証されます（クライアントはサーバーのアイデンティティを知っています）が、逆は真ではありません（クライアントは認証されないか、匿名のままです）。

TLS はより安全な相互接続モード（一般的にエンタープライズ・アプリケーションで使われます）もサポートします。それは、"対話" の両端が誰とコミュニケーションしているか保証できます（それらが相手方の証明書にあるアイデンティティ情報を入念に精査することが提供されます）。これは相互認証または 2SSL として知られてい

[13] "Diffie-Hellman." *Wikipedia.* 14 November 2009 *http://en.wikipedia.org/wiki/Diffie-Hellman*
[14] "Diffie-Hellman." *Wikipedia.* 14 November 2009 *http://en.wikipedia.org/wiki/Diffie-Hellman*
[15] "Diffie-Hellman." *Wikipedia.* 14 November 2009 *http://en.wikipedia.org/wiki/Diffie-Hellman*
[16] "RSA" *Wikipedia* 14 April 2010 *http://en.wikipedia.org/wiki/RSA*
[17] "Digital Signature Algorithm" *Wikipedia* 14 April 2010 *http://en.wikipedia.org/wiki/Digital_Signature_Algorithm*

ます。相互認証は、TLS のクライアント側も証明書を持つ必要があります（一般にエンドユーザー/ブラウザのシナリオの場合ではありません）。TLS-PSK、Secure Remote Password (SRP) プロトコルまたはいくつかの他のプロトコルが使われている場合を除き、証明書なしで強力な相互認証を提供できます。

一般的に、TLS に対して不可欠な鍵情報と証明書は X.509 証明書（必要なフィールドとデータのフォーマットを定義します）の形式で取り扱われます。

SSL は近代的な流儀で機能します。上位・下位互換およびピア間のネゴシエーションに対するサポートとともに、設計により拡張可能です。¥n[18]

A.2.5. Cramer-Shoup 暗号システム

Cramer–Shoup システムは非対称暗号アルゴリズムです。そして、標準的な暗号推測を用いた適応的選択暗号文攻撃に対して安全であると証明された、初めての効果的なスキーマでした。そのセキュリティは、決定的 Diffie–Hellman 仮定の計算的な難しさ（広く考えられていますが、証明されていません）に基づいています。1998年に Ronald Cramer と Victor Shoup により開発された、Elgamal 暗号の拡張です。極めて柔軟である Elgamal と比べて、Cramer–Shoup は資源の豊富な攻撃者に対してさえも柔軟ではないことを確定する追加の要素を追加しました。この非柔軟性は、衝突耐性のあるハッシュ機能と追加の計算の使用により達成されました。結果として Elgamal の2倍の暗号文になりました。[19]

A.2.6. ElGamal 暗号

暗号学において、ElGamal 暗号システムは Diffie-Hellman 鍵合意に基づいた公開鍵暗号に対する非対称鍵暗号アルゴリズムです。1985年に Taher Elgamal により説明されました。[1] Elgamal 暗号は、フリーの GNU Privacy Guard ソフトウェア、最近のバージョンの PGP および他の暗号システムにおいて使用されています。Digital Signature Algorithm は ElGamal 署名スキーマ（ElGamal 暗号と混同してはいけません）の変種です。[20]

[18] "Transport Layer Security" *Wikipedia* 14 April 2010 *http://en.wikipedia.org/wiki/Transport_Layer_Security*
[19] "Cramer–Shoup cryptosystem" *Wikipedia* 14 April 2010 *http://en.wikipedia.org/wiki/Cramer-Shoup_cryptosystem*
[20] "ElGamal encryption" *Wikipedia* 14 April 2010 *http://en.wikipedia.org/wiki/ElGamal_encryption*

付録B 改訂履歴

改訂 19.2-1 Sat Mar 22 2014 Christensen Eric [FAMILY Given]
sparks@fedoraproject.org

Minor changes.

改訂 19.1-1 Mon June 03 2013 Christensen Eric [FAMILY Given]
sparks@fedoraproject.org

Fedora 19 向けブランチ。
ファイアウォールの章を追加しました。

改訂 18.2-1 Wed April 17 2013 Christensen Eric [FAMILY Given]
sparks@fedoraproject.org

'制限されたサービスの管理' ガイドのセクションを追加しました。

改訂 18.1-1 Wed April 10 2013 Christensen Eric [FAMILY Given]
sparks@fedoraproject.org

SELinux ガイドのセクションを追加しました。

改訂 18.0-1 Sat October 6 2012 Christensen Eric [FAMILY Given]
sparks@fedoraproject.org

基本的な強化の章を修正しました (BZ 841825 および 693620)。
LUKS のリンク切れを修正しました (BZ 846299)。
7 Zip の章に GUI のセクションを追加しました (BZ 854781)。
yum-plugin-security の章を修正しました (BZ 723282)。
GPG CLI コマンド画面を修正しました (BZ 590493)。
Yubikey のセクションを改善しました (BZ 644238)。
誤字を修正しました (BZ 863636)。
いくつかの章において wiki のマークアップを削除しました。
Seahorse の説明を更新しました。

改訂 17.0-1 Tue January 24 2012 Christensen Eric [FAMILY Given]
sparks@fedoraproject.org

Fedora 17 向けブランチ。

改訂 16.0-1 Fri September 09 2011 Christensen Eric [FAMILY Given]
sparks@fedoraproject.org

Fedora 16 用に分岐しました。

改訂 14.3-1 Sat Apr 02 2011 Christensen Eric [FAMILY Given]
sparks@fedoraproject.org

VPN テキストを暗号の章に移動しておよび再フォーマットしました。

改訂 14.2-1 Wed Oct 20 2010 Oglesby Zach [FAMILY Given]
zoglesby@fedoraproject.org

ローカル認証とともに Fedora において Yubikey を使用するためにテキストを追加しました。(BZ 644999)

改訂 14.2-0 Fri Oct 6 2010 Christensen Eric [FAMILY Given]
sparks@fedoraproject.org

ドキュメントのソースにあるすべての変数を削除しました。

改訂 14.1-2 Fri Oct 1 2010 Christensen Eric [FAMILY Given]
sparks@fedoraproject.org

DISA Unix Checklist へのリンクと更新されたリンクを訂正しました。

改訂 14.1-1 Wed Jul 8 2010 Christensen Eric [FAMILY Given]
sparks@fedoraproject.org

CVE の章を追加しました。

改訂 14.0-1 Fri May 28 2010 Christensen Eric [FAMILY Given]
sparks@fedoraproject.org

Fedora 14 用に分岐しました。

改訂 13.0-7 Fri May 14 2010 Christensen Eric [FAMILY Given]
sparks@fedoraproject.org

bug 591980 により 7-zip の章から "バグ" のあるテキストを削除しました。

改訂 13.0-6 Wed Apr 14 2010 Christensen Eric [FAMILY Given]
sparks@fedoraproject.org

暗号標準の付録を完成させました。

改訂 13.0-5 Fri Apr 09 2010 Christensen Eric [FAMILY Given]
sparks@fedoraproject.org

"Alpine での GPG の使用" を追加しました。
"Evolution での GPG の使用" を追加しました。

改訂 13.0-4 Tue Apr 06 2010 Christensen Eric [FAMILY Given]
sparks@fedoraproject.org

パラグラフにおいて翻訳できないテキストに関する問題を修復しました。

改訂 13.0-3 Tue Apr 06 2010 Christensen Eric [FAMILY Given]
sparks@fedoraproject.org

Fedora 12 に見られる PackageKit の脆弱性のテキストを削除しました。

改訂 13.0-2 Fri Nov 20 2009 Christensen Eric [FAMILY Given]
sparks@fedoraproject.org

ドキュメントの最後に改訂履歴を追加しました。
暗号標準の付録を追加しました。

改訂 13.0-1 Fri Nov 20 2009 Christensen Eric [FAMILY Given]
sparks@fedoraproject.org

Fedora 13 の分岐をしました。

改訂 1.0-23 Thu Nov 19 2009 Christensen Eric [FAMILY Given]
sparks@fedoraproject.org

再びセクション "ローカルユーザーが信頼されるパッケージをインストールするかもしれません" を最新の修正へと更新しました。

改訂 1.0-22 Thu Nov 19 2009 Christensen Eric [FAMILY Given]
sparks@fedoraproject.org

セクション "ローカルユーザーが信頼されるパッケージをインストールするかもしれません" を最新の修正へと更新しました。

改訂 1.0-21 Wed Nov 18 2009 Christensen Eric [FAMILY Given]
sparks@fedoraproject.org

セクション "ローカルユーザーが信頼されるパッケージをインストールするかもしれません" を追加しました。

改訂 1.0-20 Sat Nov 14 2009 Christensen Eric [FAMILY Given]
sparks@fedoraproject.org

Wikipedia から暗号標準の付録へと情報を追加しました。
7-zip 部分の開発における役割に対して執筆者ページに Adam Ligas を追加しました。

改訂 1.0-19 Mon Oct 26 2009 Christensen Eric [FAMILY Given]
sparks@fedoraproject.org

ライセンスを CC-BY-SA に更新しました。

改訂 1.0-18 Wed Aug 05 2009 Christensen Eric [FAMILY Given]
sparks@fedoraproject.org

Bug 515043 に関連した問題を修正しました。

改訂 1.0-17 Mon Jul 27 2009 Christensen Eric [FAMILY Given]
sparks@fedoraproject.org

SPEC におけるベンダ情報を修復しました。

改訂 1.0-16 Fri Jul 24 2009 Release Engineering Fedora [FAMILY Given]
rel-eng@lists.fedoraproject.org

https://fedoraproject.org/wiki/Fedora_12_Mass_Rebuild のために再構築しました。

改訂 1.0-15 Tue Jul 14 2009 Christensen Eric [FAMILY Given]
sparks@fedoraproject.org

spec における BUILDREQUIRES へと "desktop-file-utils" を追加しました。

改訂 1.0-14 Tue Mar 10 2009 Radvan Scott [FAMILY Given]
sradvan@redhat.com

rhel 固有事項を削除して、ドラフトの大まかなレビューと削除をして、プッシュの準備ができました。

改訂 1.0-13 Mon Mar 2 2009 Radvan Scott [FAMILY Given]
sradvan@redhat.com

多くの軽微な修正。

改訂 1.0-12 Wed Feb 11 2009 Radvan Scott [FAMILY Given]
sradvan@redhat.com

既存/古いスクリーンショットを F11 の新しいものに置き換えました。

改訂 1.0-11 Tue Feb 03 2009 Radvan Scott [FAMILY Given]
sradvan@redhat.com

Fedora 9 の LUKS 固有事項を以降のリリースも同様に含めるよう修正しました。
参考資料セクションにおける 404 を修正しました、おもに無効な NSA リンクです。
フォーマットの軽微な変更をしました。

改訂 1.0-10 Wed Jan 27 2009 Christensen Eric [FAMILY Given]
sparks@fedoraproject.org

失われたファイアウォールのスクリーンショットを修正しました。

改訂 1.0-9 Wed Jan 27 2009 Christensen Eric [FAMILY Given]
sparks@fedoraproject.org

検証の間に不適切であることがわかった項目を修正しました。多くの Red Hat の参考資料は Fedora の参考資料に変更されました。