Red Hat Enterprise Linux 8

セキュリティーの強化

Red Hat Enterprise Linux 8 のセキュリティーに関するガイド

Last Updated: 2019-05-07

Red Hat Enterprise Linux 8 セキュリティーの強化

Red Hat Enterprise Linux 8 のセキュリティーに関するガイド

法律上の通知

概要

本書は、ユーザーおよび管理者が、ローカルおよびリモートの侵入、悪用、および悪意のある行為からワークステーションおよびサーバーを保護するプロセスおよびプラクティスを学ぶのに利用できます。本書は Red Hat Enterprise Linux を対象としていますが、概念および手法はすべての Linux システムに適用できるものです。データセンター、勤務先および自宅で安全なコンピューター環境を構築するのに必要なプランニングとツールを詳細に説明します。管理上の適切な知識、警戒体制、およびツールを備えることで、Linux を実行しているシステムの機能をフルに活用して、大概の一般的な侵入や悪用の手法からシステムを保護できます。

目次

RED HAT ドキュメントへのフィードバック

ドキュメントの改善に関するご意見やご要望をお聞かせください。

- 特定の文章に簡単なコメントを記入する場合は、ドキュメントが Multi-page HTML 形式になっているのを確認してください。コメントを追加する部分を強調表示し、そのテキストの下に表示される **Add Feedback** ポップアップをクリックし、表示された手順に従ってください。
- より詳細なフィードバックを行う場合は、Bugzilla のチケットを作成します。
    1. Bugzilla の Web サイトにアクセスします。
    2. Component で **Documentation** を選択します。
    3. **Description** フィールドに、ドキュメントの改善に関するご意見を記入してください。ドキュメントの該当部分へのリンクも記入してください。
    4. **Submit Bug** をクリックします。

第1章 RED HAT ENTERPRISE LINUX のセキュリティー強化の概要

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

しかし残念なことに、多くの組織 (個々のユーザーも含む) が、機能性や生産性、便利さ、使いやすさおよび予算面の懸念事項にばかり目を向け、セキュリティーはその結果論と見なし、セキュリティーのプロセスが見過ごされています。さらに、セキュリティーの適切な実施については、無許可の侵入が発生してはじめて徹底されることも多くあります。多くの侵入の試みを阻止する効果的な方法は、インターネットなどの信頼できないネットワークにサイトを接続する前に、適切な措置を講じることです。

1.1. コンピューターセキュリティーとは

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

1.2. セキュリティーの標準化

企業はどの業界でも、米国医師会 (AMA: American Medical Association) や米国電気電子学会 (IEEE: Institute of Electrical and Electronics Engineers) などの標準化推進団体が作成する規制やルールに従っています。情報セキュリティーにも同じことが言えます。多くのセキュリティーコンサルタントやベンダーが **機密性 (Confidentiality)、保全性 (Integrity)、可用性 (Availability)** の頭文字をとった CIA として知られる標準セキュリティーモデルを採用しています。この 3 階層モデルは、機密情報のリスク評価やセキュリティー方針の確立において一般的に採用されているモデルです。以下でこの CIA モデルについて説明します。

- 機密性 - 機密情報は、事前に定義された個人だけが利用できるようにする必要があります。許可されていない情報の送信や使用は、制限する必要があります。たとえば、情報に機密性があれば、権限のない個人が顧客情報や財務情報を悪意のある目的 (ID 盗難やクレジット詐欺など) で入手できません。

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

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

1.3. 暗号化ソフトウェアおよび認定

Red Hat Enterprise Linux は、業界のベストプラクティスに従い、**FIPS 140-2**、**Common Criteria** (CC) などのセキュリティー認証を受けています。

ナレッジベースの記事「RHEL 8 core crypto components」では、Red Hat Enterprise Linux 8 コア暗号化コンポーネントの概要 (どのコンポーネントが選択されているか、どのように選択されているか、オペレーティングシステムにどのように統合されているかどうか、ハードウェアセキュリティーモジュールおよびスマートカードがどのようにサポートされているか、これらに暗号化認証がどのように適用されているか) を説明します。

1.4. セキュリティーコントロール

多くの場合、コンピューターセキュリティーは、一般に **コントロール** と呼ばれる以下の 3 つのマスターカテゴリーに分類されます。

- 物理的
- 技術的
- 管理的

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

1.4.1. 物理的コントロール

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

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

1.4.2. 技術的コントロール

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

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

1.4.3. 管理的コントロール

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

- トレーニングおよび認識の向上
- 災害準備および復旧計画
- 人員採用と分離の戦略
- 人員登録とアカウンティング

1.5. 脆弱性のアセスメント

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

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

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

脆弱性アセスメントは、お使いのネットワークとシステムのセキュリティーに関する内部監査です。このアセスメントの結果により、ネットワークの機密性、完全性、および可用性の状態が明らかになります。通常、脆弱性アセスメントは、対象システムとリソースに関する重要なデータを収集する調査フェーズから開始します。その後システム準備フェーズとなります。基本的にこのフェーズでは、対象を絞り、それが持つすべての既知の脆弱性を調べます。準備フェーズが終わると報告フェーズになります。ここでは、調査結果が高中低のカテゴリに分類され、対象のセキュリティーを向上させる (または脆弱性のリスクを軽減する) 方法が話し合われます。

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

1.5.1. アセスメントとテストの定義

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

外部からの視点で脆弱性アセスメントを実施する場合は、システムに外部から攻撃を試みます。会社を

外から見ることで、クラッカーの視点に立つことができます。一般にルーティング可能な 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ーフェースなど、クラッカーが目を付けるものに着目します。DMZ は「非武装地帯 (demilitarized zone)」を表し、企業のプライベート LAN などの信頼できる内部ネットワークと、公的なインターネットなどの信頼できない外部ネットワークの間にあるコンピューターまたは小さなサブネットワークに相当します。通常、DMZ には Web (HTTP) サーバー、FTP サーバー、SMTP (e-mail) サーバー、DNS サーバーなど、インターネットのトラフィックにアクセスできるデバイスが含まれます。

内部からの視点で脆弱性アセスメントを実施する場合、実行者は内部関係者であり、信頼されるステータスにあることから、有利な立場になります。内部からの視点は、実行者やその同僚がシステムにログオンした時点で得られるものです。プリントサーバー、ファイルサーバー、データベースなどのリソースを見ることができます。

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

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

ネットワークインフラストラクチャーのアセスメントは動的なプロセスです。セキュリティー (情報セキュリティーおよび物理的なセキュリティー) は動的なものです。アセスメントを実施することで概要が明らかになり、誤検出 (False positives) および検出漏れ (False negatives) が示される場合があります。誤検出では、実際には存在しない脆弱性をツールが検出します。検出漏れでは、実際の脆弱性が除外されてしまいます。

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

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

警告

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

脆弱性アセスメントの実施には、以下のような利点があります。

- 情報セキュリティーに事前にフォーカスできる
- クラッカーが発見する前に潜在的な不正使用を発見できる

- システムを最新の状態に維持し、パッチを適用できる
- スタッフの成長と専門知識の開発を促す
- 経済的な損失や否定的な評判を減らす

1.5.2. 脆弱性評価に関する方法論の確立

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

ターゲットは何か? 1台のサーバー、またはネットワーク全体、そしてネットワーク内にあるすべてのものが含まれるのか? 会社の外部にいるのか、それとも内部にいるのか? これらの質問に対する答えは、ツールを選択する際だけでなく、ツールの使用方法を決定する際にも役立ちます。

方法論の確立の詳細は、以下の Web サイトを参照してください。

- https://www.owasp.org/ - **The Open Web Application Security Project**

1.5.3. 脆弱性アセスメントのツール

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

以下で、利用可能なツールを一部紹介します。

- **Nmap** は、ホストシステムを見つけて、そのシステムでポートを開くことができる一般的なツールです。**AppStream** リポジトリーから **Nmap** をインストールするには、**root** で **yum install nmap** コマンドを実行します。詳細は man ページの **nmap(1)** を参照してください。
- **oscap** コマンドラインユーティリティー、**scap-workbench** グラフィカルユーティリティーなどの **OpenSCAP** スイートのツールは、完全に自動化されたコンプライアンス監査を提供します。詳細は「セキュリティーコンプライアンスおよび脆弱性スキャンの開始」を参照してください。
- **AIDE** (Advanced Intrusion Detection Environment) は、システムのファイルのデータベースを作成し、そのデータベースを使用してファイルの整合性を確保し、システムの侵入を検出します。詳細は「Checking integrity with AIDE」を参照してください。

1.6. セキュリティーへの脅威

1.6.1. ネットワークセキュリティーへの脅威

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

セキュリティーが十分ではないアーキテクチャー

間違った構成のネットワークは、未承認ユーザーの主要のエントリーポイントになります。信頼に基づいたオープンなローカルネットワークを、安全性が非常に低いインターネットに対して無防備な状態にしておくことは、犯罪の多発地区でドアを半開きにしておくようなものです。すぐに何かが起きること

はないかもしれませんが、**いずれ**、誰かが、このチャンスを悪用するでしょう。

ブロードキャストネットワーク

システム管理者は、セキュリティー計画においてネットワーキングハードウェアの重要性を見落としがちです。ハブやルーターなどの単純なハードウェアは、ブロードキャストやノンスイッチの仕組みに基づいています。つまり、あるノードがネットワークを介して受信ノードにデータを送信するときは常に、受信ノードがデータを受信して処理するまで、ハブやルーターはデータパケットのブロードキャストを送信します。この方式は、外部侵入者やローカルホスト上の認証されていないユーザーが仕掛けるアドレス解決プロトコル (**ARP**) およびメディアアクセスコントロール (**MAC**) アドレスの偽装に対して最も脆弱です。

集中化サーバー

ネットワーキングのもうひとつの落とし穴は、集中化されたコンピューティングの使用にあります。多くの企業では、一般的なコスト削減手段として、すべてのサービスを 1 台の強力なマシンに統合しています。集中化は、複数サーバーの設定よりも管理がより簡単な上、コストを大幅に削減できるので便利な手段と言えます。しかし、集中化されたサーバーはネットワークにおける単一障害点となるため、集中化サーバーが攻撃されると、ネットワークは完全に使えなくなるか、またはデータの不正操作や盗難が起きやすくなる可能性があります。こうした状況では、1 つの集中化サーバーが侵入口となり、ネットワーク全体へのアクセスを許してしまうことになります。

1.6.2. サーバーセキュリティーへの脅威

サーバーには組織の重要情報が数多く含まれることが多いため、サーバーのセキュリティーはネットワークのセキュリティーと同様に重要です。サーバーが攻撃されると、クラッカーがすべてのコンテンツを意のままに盗んだり、不正に操作したりできるようになる可能性があります。以下のセクションでは、主要な問題のいくつかを詳述します。

未使用のサービスと開かれたポート

Red Hat Enterprise Linux 8 のフルインストールを行うと、アプリケーションとライブラリーののパッケージが 1000 個以上含まれますが、サーバー管理者が、ディストリビューションに含まれるすべての個別パッケージをインストールすることはほとんどありません。その代わりに、複数のサーバーアプリケーションを含むパッケージのベースインストールを行います。

システム管理者は、どのアプリケーションがインストールに含まれるかを気にせずにオペレーティングシステムをインストールしてしまうことがよくありますが、必要のないパッケージがインストールされ、デフォルト設定でオンになっていると、問題が発生する場合があります。したがって、管理者の気付かないところで、Telnet、DHCP、DNS などの不要なサービスがサーバーやワークステーションで実行し、その結果、サーバーへの不要なトラフィックが発生したり、システムへのパスがクラッカーに提供される可能性があります。

パッチが適用されないサービス

デフォルトのインストールに含まれるほとんどのサーバーアプリケーションは、ソフトウェアの細部まで徹底的にテストされており、堅牢な作りになっています。何年も実稼働環境で使用される中で、そのコードは入念に改良され、数多くのバグの発見や修正が行われてきました。

しかし、完璧なソフトウェアというものはなく、改良の余地は常にあります。または、比較的新しいソフトウェアは、実稼働環境に導入されてから日が浅く、他のサーバーソフトウェアほど普及していないこともあるため、厳密なテストが期待通りに行われていない状況も多く見受けられます。

開発者やシステム管理者が、サーバーアプリケーションで悪用される可能性のあるバグを発見することも多々あり、Bugtraq メーリングリスト (http://www.securityfocus.com)、Computer Emergency Response Team (CERT) Web サイト (http://www.cert.org) などの、バグ追跡やセキュリティー関連の Web サイトに関連する情報を公開します。これらは、コミュニティーにセキュリティーの脆弱性を警告する効果的な方法ではありますが、システムに速やかにパッチを当てるかどうかは個々のシステム管理

者が決定します。クラッカーも、パッチが適用されていないシステムがあればクラッキングできるように、同じように脆弱性トラッキングサービスにアクセスし、関連情報を利用できることを考慮すると、速やかな対応がとりわけ重要になります。優れたシステム管理には、よりセキュアなコンピューティング環境を維持するために、警戒を怠らず、バグ追跡を絶えず行い、適切なシステム保守を実行することが求められます。

管理における不注意

管理者がシステムにパッチを当てないことが、サーバーのセキュリティーに対する最大の脅威の 1 つになります。これは、管理者の経験の少なさだけでなく、管理者の過信やモチベーションの低さなども原因となります。

管理者はサーバーやワークステーションにパッチを当てることを忘れたり、システムのカーネルやネットワーク通信のログメッセージを見落とす場合もあります。その他にも、よく起こるケースとして、サービスのデフォルトパスワードやキーを変更しないまま放置しておくことが挙げられます。たとえば、データベースにはデフォルトの管理パスワードが設定されているものがありますが、この設定では、データベース開発者は、システム管理者がインストール後すぐにデフォルトパスワードを変更することを想定しています。しかし、データベース管理者がパスワードを変更することを忘れると、クラッカーの経験が浅くても、周知のデフォルトパスワードを使用してデータベースの管理者権限を得ることができます。この他にも、管理者の不注意によりサーバーが危険にさらされるケースが多数存在します。

本質的に安全ではないサービス

どんなに注意深い組織であっても、選択するネットワークサービスが本質的に安全でない限り、攻撃を受けやすくなります。たとえば、多くのサービスは、信頼できるネットワークで使用されるのを想定されて開発されますが、これらのサービスが (本質的に信頼できない) インターネット上で利用可能になる時点で、この仮定は成立しなくなります。

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

また、基本的にこのようなサービスはセキュリティー業界で**中間者**攻撃と呼ばれる攻撃の被害者になりやすくなります。この種の攻撃では、クラッカーは、ネットワーク上のクラッキングされたネームサーバーをだまし、目標のサーバーではなくクラッカーのマシンを指定して、ネットワークトラフィックをリダイレクトします。誰かがサーバーへのリモートセッションを開くと、攻撃者のマシンがリモートサービスと無防備なユーザーとの間に存在する目に見えないパイプとして機能し、この間を流れる情報を取り込みます。このようにして、クラッカーはサーバーやユーザーに気付かれることなく、管理パスワードや生データを収集できるようになります。

安全ではないサービスの別のカテゴリーは、NFS、NIS などのネットワークファイルシステムと情報サービスです。これらは、LAN 利用を目的として開発されましたが、(リモートユーザー用の) WAN も対象に含まれるように拡張されました。NFS では、クラッカーによる NFS 共有のマウントやそこに格納されているものへのアクセスを防ぐ認証やセキュリティーの仕組みがデフォルトで設定されていません。NIS も、プレーンテキスト ASCII または DBM (ASCII から派生) 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などの、ネットワーク上のすべてのコンピューターに知らせる必要のある重要な情報を保持しています。クラッカーがこのデータベースのアクセス権を取得すると、管理者のアカウントを含む、ネットワーク上のすべてのユーザーアカウントにアクセスできるようになってしまいます。

Red Hat Enterprise Linux 8 では、デフォルトでは、上記のサービスがすべて無効になっています。ただし、監理者は、このようなサービスを使用しないといけない場合があるため、注意して設定することが重要となります。

1.6.3. ワークステーションおよび家庭用 PC のセキュリティーに対する脅威

ワークステーションや家庭用 PC はネットワークやサーバーほど攻撃にさらされることはないかもしれませんが、クレジットカード情報のような機密データが含まれるのでシステムクラッカーの標的になります。ワークステーションは知らぬ間に攻撃者によって「スレーブ」マシンとして引き入れられ、一連の攻撃で攻撃者に使用される可能性もあります。このため、ユーザーはワークステーションの脆弱性を理解しておくと、オペレーティングシステムの再インストールや、深刻な場合はデータ盗難からの回復といった問題から免れることができます。

不適切なパスワード

攻撃者が最も簡単にシステムへのアクセスを得る方法の 1 つとして、パスワードが適切でないことが挙げられます。

脆弱なクライアントアプリケーション

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

SSH などのセキュアなプロトコルを使用している場合であっても、クライアントアプリケーションを定期的に更新していないと、リモートユーザーは特定の攻撃を受けやすくなる可能性があります。たとえば、SSH プロトコルバージョン 1 のクライアントは、悪意のある SSH サーバーからの X 転送攻撃に対して脆弱です。クライアントがサーバーに接続すると、攻撃者はネットワーク上でクライアントによるキー入力やマウス操作をひそかに収集できます。この問題は SSH プロトコルバージョン 2 で修正されましたが、ユーザーはどのアプリケーションにこのような脆弱性があるかを追跡し、必要に応じてアプリケーションを更新する必要があります。

1.7. 一般的な不正使用と攻撃

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

表1.1 一般的な不正使用

不正使用	説明	備考

不正使用	説明	備考
空またはデフォルトのパスワード	管理パスワードを空白のままにしたり、製品ベンダーが設定したデフォルトパスワードをそのまま使用します。これは、ルーターやファイアウォールなどのハードウェアで最もよく見られますが、Linux で実行するサービスにはデフォルトの管理者パスワードが入っているものがあります（ただし Red Hat Enterprise Linux 8 には含まれません)。	ルーター、ファイアウォール、VPN、Network Attached Storage (NAS) アプライアンスなどのネットワークハードウェアに一般的に関連するものです。 多数のレガシーオペレーティングシステム、特にサービスをバンドルしたオペレーティングシステム (UNIX や Windows など) によくあります。 管理者が急いで特権ユーザーアカウントを作成したためにパスワードが空白のままになっていることがありますが、これは、このアカウントを発見した悪意のあるユーザーにとっては、絶好のエントリーポイントとなります。
デフォルトの共有鍵	セキュアなサービスでは、開発や評価テスト用にデフォルトのセキュリティー鍵がパッケージ化されていることがあります。この鍵を変更せずにインターネット上の実稼働環境に置いた場合は、同じデフォルトの鍵を持つ**すべてのユーザー**がその共有鍵のリソースや、そこにあるすべての機密情報にアクセスできるようになります。	無線アクセスポイントや事前設定済みのセキュアなサーバー機器に最も多く見られます。
IP スプーフィング	リモートマシンがローカルネットワーク上のノードのように動作し、サーバーに脆弱性を見つけるとバックドアプログラムまたはトロイの木馬をインストールして、ネットワークリソース全体へのコントロールを得ようとします。	スプーフィングは、攻撃者が標的となるシステムへの接続を調整するのに、TCP/IP シーケンス番号を予測しなければならないため、かなり難しくなりますが、クラッカーの脆弱性の攻撃を支援する利用可能なツールがいくつかあります。 標的となるシステムで実行される、**source-based** 認証技術を使用するサービス (**rsh**、**telnet**、FTP など) により異なりますが、このようなサービスは、**ssh**、または SSL/TLS で使用される PKI などの形式の暗号化認証と比較すると推奨されません。

不正使用	説明	備考
盗聴	2 つのノード間の接続を盗聴することにより、ネットワーク上のアクティブなノード間を行き交うデータを収集します。	この種類の攻撃には大抵、Telnet、FTP、HTTP 転送などのプレーンテキストの転送プロトコルが使われます。 リモートの攻撃者がこのような攻撃を仕掛けるには、LAN 上で、攻撃するシステムへのアクセス権が必要になります。通常、クラッカーは、LAN 上にあるシステムを危険にさらすためにアクティブ攻撃（IP スプーフィングや中間者攻撃など) を仕掛けます。 パスワードのなりすましを防ぐ予防策としては、暗号化鍵交換、ワンタイムパスワード、または暗号化された認証によるサービス使用が挙げられます。通信中は強力な暗号化を実施することをお勧めします。

不正使用	説明	備考
サービスの脆弱性	攻撃者はインターネット上で実行されているサービスの欠陥や抜け穴を見つけます。この脆弱性を利用する攻撃者は、システム全体と格納されているデータを攻撃するだけでなく、ネットワーク上の他のシステムも攻撃する可能性があります。	CGI などの HTTP ベースのサービスは、リモートのコマンド実行やインタラクティブなシェルアクセスに対しても脆弱です。HTTP サービスが「nobody」などの権限のないユーザーとして実行される場合でも、設定ファイルやネットワークマップなどの情報が読み取られる可能性があります。または、攻撃者がサービス拒否攻撃を開始して、システムのリソースを浪費させたり、他のユーザーが利用できないようにする可能性もあります。 開発時およびテスト時には気付かない脆弱性がサービスに含まれることがあります。(攻撃者が任意の値を使用してアプリケーションのメモリーバッファー領域をあふれさせ、任意のコマンドを実行できるようなインタラクティブなコマンドプロンプトを攻撃者に提供する、サービスをクラッシュさせる **バッファーオーバーフロー**などの) 脆弱性は完全な管理コントロールを攻撃者に与えるものとなる可能性があります。 管理者は、root 権限でサービスが実行されないようにし、ベンダー、または CERT、CVE などのセキュリティー組織がアプリケーション用のパッチやエラータ更新を提供していないかを常に注意する必要があります。

不正使用	説明	備考
アプリケーションの脆弱性	攻撃者はデスクトップやワークステーションのアプリケーション（電子メールクライアントなど）に欠陥を見つけ出し、任意のコードを実行したり、将来のシステム侵害のためにトロイの木馬を移植したり、システムを破壊したりします。攻撃を受けたワークステーションがネットワークの残りの部分に対して管理特権を持っている場合は、さらなる不正使用が起こる可能性があります。	ワークステーションとデスクトップは、ユーザーが侵害を防いだり検知するための専門知識や経験を持たないため、不正使用の対象になりやすくなります。認証されていないソフトウェアをインストールしたり、要求していないメールの添付ファイルを開く際には、それに伴うリスクについて個々に通知することが必須です。 電子メールクライアントソフトウェアが添付ファイルを自動的に開いたり、実行したりしないようにするといった、予防手段を取ることが可能です。さらに、Red Hat Network や他のシステム管理サービスなどから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ーデプロイメントの負担を軽減できます。
サービス拒否攻撃 (DoS: Denial of Service)	単独の攻撃者または攻撃者のグループは、目標のホスト (サーバー、ルーター、ワークステーションのいずれか) に認証されていないパケットを送り、組織のネットワークまたはサーバーのリソースに対して攻撃を仕掛けます。これにより、正当なユーザーはリソースを使用できなくなります。	2000 年に発生した米国内での DoS で最も多く報告されたケースとして、通信量が非常に多い民間および政府サイトのいくつかが利用できなくなりました。**ゾンビ (zombie)** や、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幅接続を有し、セキュリティー侵害された複数のシステムを使用して、調整された ping フラッド攻撃が行われたためです。 通常ソースパケットは、真の攻撃元を調査するのが難しくなるよう、偽装 (または再ブロードキャスト）されています。 **iptables** を使用したイングレスフィルタリング (IETF rfc2267) や、**snort** などのネットワーク侵入検知システムにおける進歩は、管理者が分散型サービス拒否攻撃を追跡し、これを防止するのに役立っています。

第2章 インストール時の RHEL の保護

セキュリティーは、Red Hat Enterprise Linux をインストールする前にすでに始まっています。最初からシステムのセキュリティーを設定することで、追加のセキュリティー設定を実装することがより簡単になります。

2.1. BIOS および UEFI のセキュリティー

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

たとえば、見本市で使われていて機密情報を含んでいないマシンでは、このような攻撃を防ぐことが重要ではないかもしれません。しかし、同じ見本市で、企業ネットワークに対して暗号化されていない SSH 秘密鍵のある従業員のノートパソコンが、誰の監視下にもなく置かれていた場合は、重大なセキュリティー侵害につながり、その影響は企業全体に及ぶ可能性があります。

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

2.1.1. BIOS パスワード

コンピューターの BIOS をパスワードで保護する主な理由は、以下の 2 つです[1]。

1. **BIOS 設定の変更を防止する** - 侵入者が BIOS にアクセスした場合は、CD-ROM やフラッシュドライブから起動するように設定できます。このようにすると、侵入者が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可能になり、システム上で任意のプロセスを開始したり、機密性の高いデータをコピーできるようになってしまいます。

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

BIOS パスワードの設定方法はコンピューターメーカーで異なるため、具体的な方法についてはコンピューターのマニュアルを参照してください。

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

2.1.1.1. 非 BIOS ベースシステムのセキュリティー

他のシステムやアーキテクチャーでは、異なるプログラムを使用して x86 システム上の BIOS とほぼ同等の低レベルのタスクを実行します。**UEFI** (**Unified Extensible Firmware Interface**) シェルなどがこの例になります。

BIOS と同様のプログラムをパスワード保護する方法は、メーカーの指示を参照してください。
:experimental:

2.2. ディスクのパーティション設定

Red Hat は、**/boot**、/、**/home**/**tmp**、および **/var/tmp/** の各ディレクトリーに別々のパーティションを作成することをお勧めします。それぞれの理由は異なり、各パーティションに取り組みます。

/boot

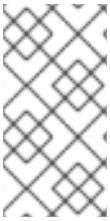
このパーティションは、システムの起動時にシステムが最初に読み込むパーティションです。Red Hat Enterprise Linux 8 でシステムを起動するのに使用されるブートローダーとカーネルイメージはこのパーティションに保存されます。このパーティションは暗号化しないでください。このパーティションが / に含まれており、そのパーティションが暗号化されているなどの理由で利用できなくなると、システムを起動できなくなります。

/home

ユーザーデータ (**/home**) が別のパーティションではなく / に保存されていると、このパーティションが一杯になり、オペレーティングシステムが不安定になる可能性があります。また、システムを、Red Hat Enterprise Linux 8 の次のバージョンにアップグレードする際に、**/home** パーティションにデータを保存できると、このデータはインストール時に上書きされないので、アップグレードが非常に簡単になります。root パーティション (/) が破損すると、データは完全に失われてしまいます。個別のパーティションを使用することで、データ損失に対する保護が少しは高まります。また、このパーティションを頻繁にバックアップする対象とすることも可能です。

/tmp および /var/tmp/

/tmp ディレクトリーおよび **/var/tmp/** ディレクトリーは、どちらも長期保存の必要がないデータを保存するために使用されます。しかし、このいずれかのディレクトリーでデータがあふれると、ストレージ領域がすべて消費されてしまう可能性があります。このディレクトリーが / に保管されていて、こうした状態が発生すると、システムは不安定になり、クラッシュする可能性があります。そのため、このディレクトリーは個別のパーティションに移動することが推奨されます。



注記

インストールプロセス時に、パーティションを暗号化するオプションがユーザーに示されます。ユーザーは、パスフレーズを入力する必要があります。これは、パーティションのデータを保護するために使用されるバルク暗号鍵を解除する鍵として使用されます。

2.3. インストールプロセス時のネットワーク接続の制限

Red Hat Enterprise Linux をインストールする際に使用するインストールメディアは、特定のタイミングで作成されたスナップショットです。そのため、セキュリティー修正が最新のものではなく、このインストールメディアで設定するシステムが公開されてから修正された特定の問題に対して安全性に欠ける場合があります。

脆弱性が含まれる可能性のあるオペレーティングシステムをインストールする場合には、必ず、公開レベルを、必要最小限のネットワークゾーンに限定してください。最も安全な選択肢は、インストールプロセス中はマシンをネットワークから切断した状態にする「ネットワークなし」のゾーンです。インターネット接続からのリスクが最も高くなっていますが、LAN またはイントラネット接続で十分な場合もあります。ベストなセキュリティーの慣習に従い、ネットワークからRed Hat Enterprise Linux 8 をインストールする場合は、お使いのリポジトリーに最も近いゾーンを選択するようにしてください。

2.4. 必要なパッケージの最小限のインストール

コンピューター上の各ソフトウェアには脆弱性が潜んでいる可能性があるため、実際に使用するパッケージのみをインストールすることがベストプラクティスになります。インストールを DVD から行う場合は、インストールしたいパッケージのみを選択するようにします。他のパッケージが必要になる場合は、後でいつでもシステムに追加できます。

2.5. インストール後の手順

以下は、Red Hat Enterprise Linux のインストール直後に実行する必要があるセキュリティー関連の手順です。

1. システムを更新します。root で以下のコマンドを実行します。

```
# dnf update
```

2. ファイアウォールサービスの **firewalld** は Red Hat Enterprise Linux のインストールで自動的に有効になっていますが、kickstart 設定などで明示的に無効となっている場合もあります。このような場合は、ファイアウォールを再度有効にすることが推奨されます。
firewalld を開始するには、root で以下のコマンドを実行します。

```
# systemctl start firewalld
# systemctl enable firewalld
```

3. セキュリティーを強化するために、不要なサービスは無効にしてください。たとえば、使用中のコンピューターにプリンターがインストールされていなければ、以下のコマンドを使用して **cups** サービスを無効にします。

```
# systemctl disable cups
```

アクティブなサービスを確認するには、以下のコマンドを実行します。

```
$ systemctl list-units | grep service
```

---

[1] システム BIOS はメーカーによって異なるため、いずれかのタイプのパスワード保護のみをサポートするものもあれば、いずれのタイプのパスワード保護もサポートしないものもあります。

第3章 システム全体の暗号化ポリシーの使用

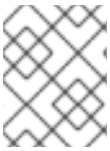
暗号ポリシーは、コア暗号化サブシステムを構成するシステムコンポーネントで、TLS、IPSec、SSH、DNSSec、および Kerberos の各プロトコルに対応します。これにより、管理者が選択できる小規模セットのポリシーを提供します。

3.1. システム全体の暗号化ポリシー

システム全体のポリシーを設定すると、RHEL のアプリケーションはそのポリシーに従い、ポリシーを満たしていないアルゴリズムやプロトコルを使用するように明示的に要求されない限り、その使用を拒否します。つまり、システムが提供した設定で実行する際にデフォルトのアプリケーションの挙動にポリシーを適用しますが、必要な場合は上書きできます。

Red Hat Enterprise Linux 8 には、以下のポリシーレベルが含まれます。

DEFAULT	デフォルトのシステム全体の暗号化ポリシーレベルで、現在の脅威モデルに対して安全なものです。TLS プロトコル 1.2 と 1.3、IKEv2、SSH2 プロトコルを許可します。RSA 鍵と Diffie-Hellman パラメーターは長さが 2048 ビット以上であれば許容されます。
LEGACY	このポリシーは、Red Hat Enterprise Linux 5 以前のリリースとの互換性を最大化しますが、攻撃領域が大きくなるため脆弱になります。**DEFAULT** レベルでのアルゴリズムとプロトコルに加えて、TLS プロトコル 1.0 および 1.1 を許可します。アルゴリズム DSA、3DES、および RC4 が許可され、RSA 鍵と Diffie-Hellman パラメーターの長さが 1023 ビット以上であれば許容されます。
FUTURE	近い将来の攻撃に耐えられると考えられている保守的なセキュリティーレベルです。このレベルは、署名アルゴリズムに SHA-1 の使用を許可しません。RSA 鍵と Diffie-Hellman パラメーターは、ビット長が 3072 以上だと許可されます。
FIPS	FIPS140-2 要件に準拠するポリシールールです。これは、**fips-mode-setup** ツールの内部で使用され、RHEL システムを FIPS140-2 準拠モードに切り替えます。



注記

ポリシーレベルで許可されていると記載されている特定のアルゴリズムと暗号は、アプリケーションがそれをサポートしている場合に限り使用できます。

暗号化ポリシーを管理するツール

現在のシステム全体の暗号化ポリシーを表示または変更するには、**update-crypto-policies** ツールを使用します。以下に例を示します。

```
$ update-crypto-policies --show
DEFAULT
```

```
# update-crypto-policies --set FUTURE
Setting system policy to FUTURE
```

確実に暗号化ポリシーの変更を適用するには、システムを再起動します。

安全ではない暗号スイートおよびプロトコルを削除することによる強力な暗号デフォルト

次の一覧は、RHEL 8 のコア暗号化ライブラリーから削除された暗号スイートおよびプロトコルです。これは、ソースコード内に存在しないか、ビルド時に無効化されているため、アプリケーションが利用することはできません。

- DES (RHEL 7 以降)
- すべてのエクスポートグレードの暗号化スイート (RHEL 7 以降)
- 署名内の MD5 (RHEL 7 以降)
- SSLv2 (RHEL 7 以降)
- SSLv3 (RHEL 8 以降)
- すべての ECC 曲線 < 224 ビット (RHEL 6 以降)
- すべてのバイナリーフィールドの ECC 曲線 (RHEL 6 以降)

すべてのポリシーレベルで無効になっている暗号スイートおよびプロトコル
以下の一覧は、すべての暗号化ポリシーレベルで無効になっています。これは、各アプリケーションで明示的に有効にした場合のみ利用可能にできます。

- パラメーターを持つ DH < 1024 ビット
- 鍵サイズを持つ RSA < 1024 ビット
- Camellia
- ARIA
- SEED
- IDEA
- 完全性のみの暗号スイート
- SHA-384 HMAC を使用した TLS CBC モード暗号化スイート
- AES-CCM8
- TLS 1.3 と互換性がないすべてのECC曲線 (secp256k1 を含む)
- IKEv1 (RHEL 8 以降)

暗号ポリシーレベルで有効にされる暗号スイートおよびプロトコル
次の表は、暗号ポリシーの各レベルで有効な暗号化ポリシーとプロトコルを示しています。

	LEGACY	DEFAULT	FIPS	FUTURE
IKEv1	no	no	no	no
3DES	yes	no	no	no
RC4	yes	no	no	no

	LEGACY	DEFAULT	FIPS	FUTURE
DH	min. 1024-bit	min. 2048-bit	min. 2048-bit	min. 3072-bit
RSA	min. 1024-bit	min. 2048-bit	min. 2048-bit	min. 3072-bit
DSA	yes	no	no	no
TLS v1.0	yes	no	no	no
TLS v1.1	yes	no	no	no
デジタル署名における SHA-1	yes	yes	no	no
CBC モード暗号	yes	yes	yes	no
鍵を持つ対称暗号 < 256 ビット	yes	yes	yes	no
証明書における SHA-1 および SHA-224 の署名	yes	yes	yes	no

関連資料

- 詳細は、man ページの **update-crypto-policies(8)** を参照してください。

3.2. システム全体の暗号化ポリシーを、以前のリリースと互換性のあるモードに切り替え

Red Hat Enterprise Linux 8 におけるデフォルトのシステム全体の暗号化ポリシーでは、現在は古くて安全ではないプロトコルは許可されません。Red Hat Enterprise Linux 5 およびそれ以前のリリースとの互換性が必要な場合には、安全でない **LEGACY** ポリシーレベルを利用できます。

警告

LEGACY ポリシーレベルに設定するとシステムおよびアプリケーションの安全性が低下します。

手順

1. システム全体の暗号化ポリシーを **LEGACY** レベルに切り替えるには、**root** で以下のコマンドを実行します。

```
# update-crypto-policies --set LEGACY
Setting system policy to LEGACY
```

関連資料

- 利用可能な暗号化ポリシーのレベルは、man ページの **update-crypto-policies(8)** を参照してください。

3.3. システムを FIPS140-2 準拠モードに切り替え

システム全体の暗号化ポリシーには、連邦情報処理規格 (FIPS) 140-2 に準拠したポリシーレベルが含まれます。FIPS 140-2 モードを有効または無効にする **fips-mode-setup** ツールは、内部的に **FIPS** システム全体の暗号化ポリシーレベルを使用します。

手順

1. RHEL 8 で FIPS140-2 コンプライアンスモードにシステムを切り替えるには、以下のコマンドを実行します。

```
# fips-mode-setup --enable
Setting system policy to FIPS
FIPS mode will be enabled.
Please reboot the system for the setting to take effect.
```

2. システムを再起動して、カーネルを FIPS モードに切り替えます。

```
# reboot
```

3. 再起動後、FIPS モードの現在の状態を確認できます。

```
# fips-mode-setup --check
FIPS mode is enabled.
```

関連資料

- man ページの **fips-mode-setup(8)**
- FIPS 140-2 の詳細は、National Institute of Standards and Technology (NIST) Web サイトの「Security Requirements for Cryptographic Modules」 を参照してください。

3.4. アプリケーションをシステム全体の暗号化ポリシーに従わないように除外

アプリケーションで使用される暗号化関連の設定をカスタマイズする必要がある場合は、サポートされる暗号スイートとプロトコルをアプリケーションで直接設定することが推奨されます。

/etc/crypto-policies/back-ends ディレクトリーからアプリケーション関連のシンボリックリンクを削除することもできます。カスタマイズした暗号化設定に置き換えることもできます。この設定により、除外されたバックエンドを使用するアプリケーションに対するシステム全体の暗号化ポリシーが使用できなくなります。この修正は、Red Hat ではサポートされていません。

システム全体の暗号化ポリシーを除外する例

- **wget** ネットワークダウンローダーで使用される暗号化設定をカスタマイズするには、**--secure-protocol** オプションおよび **--ciphers** オプションを使用します。以下に例を示します。

```
# wget --secure-protocol=TLSv1_1 --ciphers="SECURE128" https://example.com
```

  詳細は、man ページの **wget(1)** の HTTPS (SSL/TLS) Options のセクションを参照してください。

- **curl** ツールで使用する暗号を指定するには、**--ciphers** オプションを使用して、その値に、コロンで区切った暗号化のリストを指定します。以下に例を示します。

```
# curl https://example.com --ciphers DES-CBC3-SHA:RSA-DES-CBC3-SHA
```

  詳細は、man ページの **curl(1)** を参照してください。

- アドレスバーに **about:config** と入力し、必要に応じて **security.tls.version.min** オプションの値を変更します。**security.tls.version.min** を **1** に設定すると、最低でも TLS 1.0 が必要になります。**security.tls.version.min 2** は TLS 1.1 に一致します。

- **OpenSSH** サーバーに対するシステム全体の暗号化ポリシーを除外するには、**/etc/sysconfig/sshd** ファイルの **CRYPTO_POLICY=** 変数の行のコメントを除外します。この変更後、**/etc/ssh/sshd_config** ファイルの **Ciphers** セクション、**MACs** セクション、**KexAlgoritms** セクション、および **GSSAPIKexAlgorithms** セクションで指定した値は上書きされません。詳細は、man ページの **sshd_config(5)** を参照してください。

- **Libreswan** IPsec スイートで IKEv1 プロトコルを使用できるようにするには、**/etc/ipsec.conf** ファイルの次の行をコメントアウトします。

```
include /etc/crypto-policies/back-ends/libreswan.config
```

  次に、接続設定に **ikev2=never** オプションを追加してください。詳細は man ページの **ipsec.conf(5)** を参照してください。

関連資料

- 詳細は、man ページの **update-crypto-policies(8)** を参照してください。

3.5. 関連情報

- 詳細は、Red Hat カスタマーポータルのナレッジベースの記事「System-wide crypto policies in RHEL 8」 および 「Strong crypto defaults in RHEL 8 and deprecation of weak crypto algorithms」を参照してください。

第4章 共通システム証明書の使用

共有システム証明書ストレージは、NSS、GnuTLS、OpenSSL、および Java が、システムの証明書アンカーと、ブラックリスト情報を取得するデフォルトソースを共有します。トラストストアには、デフォルトで、Mozilla CA の一覧 (信頼される一覧および信頼されない一覧) を含みます。システムは、コア Mozilla CA 一覧を更新したり、証明書一覧を選択したりできます。

4.1. システム全体でトラストストアの使用

Red Hat Enterprise Linux では、統合されたシステム全体のトラストストアが **/etc/pki/ca-trust/** ディレクトリーおよび **/usr/share/pki/ca-trust-source/** ディレクトリーに置かれています。**/usr/share/pki/ca-trust-source/** のトラスト設定は、**/etc/pki/ca-trust/** の設定よりも低い優先順位で処理されます。

証明書ファイルは、以下のディレクトリーにインストールされているサブディレクトリーによって扱われ方が異なります。

- トラストアンカーの場合
  - **/usr/share/pki/ca-trust-source/anchors/** または
  - **/etc/pki/ca-trust/source/anchors/**
- 信頼されない証明書の場合
  - **/usr/share/pki/ca-trust-source/blacklist/** または
  - **/etc/pki/ca-trust/source/blacklist/**
- 拡張された BEGIN TRUSTED ファイル形式の証明書の場合
  - **/usr/share/pki/ca-trust-source/** または
  - **/etc/pki/ca-trust/source/**

注記

階層暗号化システムでは、トラストアンカーは信頼できると想定される、信頼できるエンティティーです。たとえば、X.509 アーキテクチャーでは、ルート証明書がトラストチェーンの元となるトラストアンカーとなっています。トラストアンカーは、パスの検証ができるように、事前に信頼できる団体が所有しておく必要があります。

4.2. 新しい証明書の追加

1. システムで信頼されている CA の一覧に、シンプルな PEM または DER のファイルフォーマットに含まれる証明書を追加するには、**/usr/share/pki/ca-trust-source/anchors/** ディレクトリーまたは **/etc/pki/ca-trust/source/anchors/** ディレクトリーに証明書ファイルをコピーします。以下に例を示します。

```
# cp ~/certificate-trust-examples/Cert-trust-test-ca.pem /usr/share/pki/ca-trust-source/anchors/
```

2. システム全体のトラストストアを更新するには、**update-ca-trust** コマンドを実行します。

```
# update-ca-trust
```

注記

Firefox ブラウザーでは、**update-ca-trust** を実行しなくても、追加した証明書を使用できますが、CA 変更後に **update-ca-trust** を実行することが推奨されます。Firefox、Epiphany、Chromium などのブラウザーはファイルをキャッシュするため、現在のシステム証明書の設定を読み込むために、ブラウザーのキャッシュを削除して、ブラウザーを再起動することが必要になるかもしれません。

4.3. 信頼されているシステム証明書の管理

- トラストアンカーの一覧表示、抽出、追加、削除、または変更を行うには、**trust** コマンドを使用します。このコマンドの組み込みヘルプを表示するには、引数を付けずに、または **--help** ディレクティブを付けて実行します。

```
$ trust
usage: trust command <args>...

Common trust commands are:
  list            List trust or certificates
  extract         Extract certificates and trust
  extract-compat  Extract trust compatibility bundles
  anchor          Add, remove, change trust anchors
  dump            Dump trust objects in internal format

See 'trust <command> --help' for more information
```

- すべてのシステムのトラストアンカーおよび証明書の一覧を表示するには、**trust list** コマンドを実行します。

```
$ trust list
pkcs11:id=%d2%87%b4%e3%df%37%27%93%55%f6%56%ea%81%e5%36%cc%8c%1e%3f%bd;type=cert
    type: certificate
    label: ACCVRAIZ1
    trust: anchor
    category: authority

pkcs11:id=%a6%b3%e1%2b%2b%49%b6%d7%73%a1%aa%94%f5%01%e7%73%65%4c%ac%50;type=cert
    type: certificate
    label: ACEDICOM Root
    trust: anchor
    category: authority
...
[output has been truncated]
```

- トラストアンカーをシステム全体のトラストストアに保存するには、**trust anchor** サブコマンドを使用し、証明書のパスを指定します。**path.to/certificate.crt** を、証明書およびそのファイル名のパスに置き換えます。

```
# trust anchor path.to/certificate.crt
```

- 証明書を削除するには、証明書のパス、または証明書の ID を使用します。

```
# trust anchor --remove path.to/certificate.crt
# trust anchor --remove "pkcs11:id=%AA%BB%CC%DD%EE;type=cert"
```

関連資料
trust コマンドのすべてのサブコマンドは、以下のような詳細な組み込みヘルプを提供します。

```
$ trust list --help
usage: trust list --filter=<what>

  --filter=<what>     filter of what to export
                      ca-anchors        certificate anchors
                      blacklist         blacklisted certificates
                      trust-policy      anchors and blacklist (default)
                      certificates      all certificates
                      pkcs11:object=xx  a PKCS#11 URI
  --purpose=<usage>   limit to certificates usable for the purpose
                      server-auth       for authenticating servers
                      client-auth       for authenticating clients
                      email             for email protection
                      code-signing      for authenticating signed code
                      1.2.3.4.5...      an arbitrary object id
  -v, --verbose       show verbose debug output
  -q, --quiet         suppress command output
```

4.4. 関連情報

詳細は、以下の man ページを参照してください。

- **update-ca-trust(8)**
- **trust(1)**

第5章 セキュリティーコンプライアンスおよび脆弱性スキャンの開始

コンプライアンス監査は、指定したオブジェクトが、コンプライアンスポリシーに記載されているすべてのルールに従っているかどうかを判断するプロセスです。コンプライアンスポリシーは、コンピューティング環境で使用される必要な設定を指定するセキュリティー専門家が定義します。これは多くの場合、チェックリストの形式を取ります。

コンプライアンスポリシーは組織により大幅に異なることがあり、同一組織内でもシステムが異なるとポリシーが異なる可能性があります。システムの目的や組織におけるシステム重要性に基づいて、これらのポリシーは異なります。カスタマイズしたソフトウェア設定や導入の特徴によっても、カスタマイズしたポリシーのチェックリストが必要になってきます。

5.1. RHEL におけるセキュリティーコンプライアンスツール

Red Hat Enterprise Linux は、完全に自動化されたコンプライアンス監査を可能にするツールを提供します。このツールは SCAP (Security Content Automation Protocol) 標準に基づいており、コンプライアンスポリシーの自動化に合わせるように設計されています。

- SCAP Workbench - **scap-workbench** グラフィカルユーティリティーは、1 台のローカルシステムまたはリモートシステムで構成スキャンと脆弱性スキャンを実行するように設計されています。これらのスキャンと評価に基づくセキュリティーレポートの生成にも使用できます。

- OpenSCAP - **oscap** コマンドラインユーティリティーは、ローカルシステムで構成スキャンと脆弱性スキャンを実行するように設計されています。これにより、セキュリティーコンプライアンスのコンテンツを検証し、スキャンおよび評価に基づいてレポートおよびガイドを生成します。

- SCAP Security Guide (SSG) - **scap-security-guide** パッケージは、Linux システム向けの最新のセキュリティーポリシーコレクションを提供します。このガイダンスは、セキュリティー強化に関する実践的なアドバイスのカタログから構成されます (該当する場合は、法規制要件へのリンクが含まれます)。このプロジェクトは、一般的なポリシー要件と特定の実装ガイドラインとの間にあるギャップを埋めることを目的としています。

- Script Check Engine (SCE) - SCE は SCAP プロトコルの拡張機能であり、この機能を使用すると管理者が Bash、Python、Ruby などのスクリプト言語を使用してセキュリティーコンテンツを記述できるようになります。SCE 拡張機能は、**openscap-engine-sce** パッケージで提供されます。

複数のリモートシステムで自動コンプライアンス監査を実行する必要がある場合は、Red Hat Satellite 用の OpenSCAP ソリューションを利用できます。

関連資料

- **oscap(8)** - **oscap** コマンドラインユーティリティーの man ページでは、サポートされるすべてのオプションとその使用方法が説明されます。

- **scap-workbench(8)** - **SCAP Workbench** アプリケーションの man ページでは、このアプリケーションの基本情報と、SCAP コンテンツの潜在的なソースへのリンクが提供されます。

- **scap-security-guide(8)** - **scap-security-guide** プロジェクトの man ページでは、利用可能な SCAP セキュリティープロファイルに関するドキュメントが提供されます。OpenSCAP ユーティリティーを使用して提供されたベンチマークの使用例も提供されてます。

- Red Hat Satellite で OpenSCAP を使用する方法は 『セキュリティーコンプライアンスの管理』 を参照してください。

5.2. RED HAT SECURITY ADVISORIES OVAL フィード

Red Hat Enterprise Linux のセキュリティー監査機能は、セキュリティー設定共通化手順 (Security Content Automation Protocol (SCAP)) 標準仕様に基にしています。SCAP は、自動化された設定、脆弱性およびパッチの確認、技術的な制御コンプライアンスアクティビティー、およびセキュリティーの測定に対応している多目的な仕様のフレームワークです。

SCAP 仕様は、セキュリティーコンテンツの形式が周知かつ標準化されたエコシステムを作り出す一方で、スキャナーやポリシーエディターの導入は義務化されません。このような状態では、企業がいくつものセキュリティーベンダーを用いていても、組織がセキュリティーポリシー (SCAP コンテンツ) を構築するのは一度で済みます。

セキュリティー検査言語 OVAL (Open Vulnerability Assessment Language) は、SCAP に不可欠で最も古いコンポーネントです。他のツールやカスタマイズされたスクリプトとは異なり、OVAL 言語は宣言型でリソースの必要な状態を記述します。OVAL 言語コードは、スキャナーと呼ばれる OVAL インタープリターツールを用いて実行されますが、直接実行されることは決してありません。OVAL が宣言型であるため、評価されるシステムの状態が偶然修正されることはありません。

他のすべての SCAP コンポーネントと同様に、OVAL は XML に基づいています。SCAP 標準は、いくつかのドキュメント形式を定義します。これらはそれぞれ異なる種類の情報を含み、異なる目的に使われます。

Red Hat 製品セキュリティー を使用すると、Red Hat 製品をお使いのお客様に影響を及ぼすセキュリティー問題をすべて追跡して調査し、Red Hat カスタマーポータルで簡潔なパッチやセキュリティーアドバイザリーを適時提供することで、お客様がリスクを評価して管理できます。Red Hat は、OVAL パッチ定義を作成してサポートし、マシンが読み取り可能なセキュリティーアドバイザリーを定義します。

各 RHSA OVAL 定義 は完全なパッケージとして利用でき、新しいセキュリティーアドバイザリーが Red Hat カスタマーポータルで利用可能になってから 1 時間以内に更新されます。

各 OVAL パッチ定義は、Red Hat セキュリティーアドバイザリー (RHSA) と 1 対 1 にマッピングしています。RHSA には複数の脆弱性に対する修正が含まれるため、各脆弱性は、共通脆弱性識別子 (Common Vulnerabilities and Exposures (CVE)) 名ごとに表示され、公開バグデータベースの該当箇所へのリンクが示されます。

RHSA OVAL 定義は、システムにインストールされている RPM パッケージで脆弱なバージョンを確認するように設計されています。この定義は拡張でき、パッケージが脆弱な設定で使用されているかどうかを見つけるなど、さらに確認できるようにすることができます。この定義は、Red Hat が提供するソフトウェアおよび更新に対応するように設計されています。サードパーティーソフトウェアのパッチ状態を検出するには、追加の定義が必要です。

関連資料

- Red Hat and OVAL compatibility
- Red Hat and CVE compatibility
- 製品セキュリティーの概要 の 通知およびアドバイザリー
- Security Data Metrics

5.3. システムの脆弱性のスキャン

oscap コマンドラインユーティリティーを使用すると、ローカルシステムのスキャン、セキュリティーコンプライアンスコンテンツの確認、これらのスキャンおよび評価を基にしたレポートとガイドの生成が可能です。このユーティリティーは OpenSCAP ライブラリーへのフロントエンドとしてサービスを提供し、その機能を処理する SCAP コンテンツのタイプに基づいてモジュール (サブコマンド) にグループ化します。

前提条件

- **AppStream** リポジトリーが有効になっている。

手順

1. **openscap-scanner** パッケージをインストールします。

```
# yum install openscap-scanner
```

2. システムに最新 RHSA OVAL 定義をダウンロードします。

```
# wget https://www.redhat.com/security/data/oval/Red_Hat_Enterprise_Linux_8.xml
```

3. システムの脆弱性をスキャンし、**vulnerability.html** ファイルに結果を保存します。

```
# oscap oval eval --report vulnerability.html Red_Hat_Enterprise_Linux_8.xml
```

その結果をブラウザーで確認します。以下に例を示します。

```
$ firefox vulnerability.html &
```

関連資料

- **oscap(8)** の man ページ
- Red Hat OVAL definitions の一覧

5.4. リモートシステムの脆弱性のスキャン

OpenSCAP スキャナーで、リモートシステムの脆弱性も確認できます。この機能は、**oscap-ssh** ツールにより SSH プロトコルで有効になります。

前提条件

- **AppStream** リポジトリーが有効になっている。
- リモートシステムに **openscap-scanner** パッケージがインストールされている。
- リモートシステムで SSH サーバーが実行している。

手順

1. **openscap-utils** パッケージをインストールします。

```
# yum install openscap-utils
```

2. システムに最新 RHSA OVAL 定義をダウンロードします。

```
# wget https://www.redhat.com/security/data/oval/Red_Hat_Enterprise_Linux_8.xml
```

3. 脆弱性に対して、ホスト名 **machine1**、ポート 22 で実行する SSH、およびユーザー名 **joesec** でリモートシステムをスキャンし、結果を **remote-vulnerability.html** ファイルに保存します。

```
# oscap-ssh joesec@machine1 22 oval eval --report remote-vulnerability.html
Red_Hat_Enterprise_Linux_8.xml
```

関連資料

- **oscap-ssh(8)** の man ページ
- Red Hat OVAL definitions の一覧

5.5. 関連情報

- OpenSCAP プロジェクトページ - OpenSCAP プロジェクトのホームページでは、oscap ユーティリティーと、SCAP に関連する他のコンポーネントおよびプロジェクトの詳細情報が提供されています。
- SCAP Workbench プロジェクトページ - SCAP Workbench プロジェクトのホームページでは、scap-workbench アプリケーションの詳細情報が提供されています。
- SCAP Security Guide (SSG) プロジェクトページ - SSG プロジェクトのホームページでは、Red Hat Enterprise Linux 向けの最新セキュリティーコンテンツが提供されています。
- National Institute of Standards and Technology (NIST) SCAP のページ - このページには、SCAP の出版物、仕様、SCAP 検出プログラムなどの SCAP 関連の資料が大量にあります。
- National Vulnerability Database (NVD) - このページは、SCAP コンテンツおよび他の SCAP 標準ベースの脆弱性管理データの最大のリポジトリーです。
- Red Hat OVAL content repository - Red Hat Enterprise Linux システムの脆弱性に関する OVAL 定義を含むリポジトリーです。このページは、脆弱性の情報を得るために確認が推奨されるページです。
- MITRE CVE - これは、MITRE corporation が提供する既知のセキュリティー脆弱性のデータベースです。RHEL の場合は、Red Hat が提供する OVAL CVE コンテンツを使用することが推奨されます。
- MITRE OVAL - このページでは、MITRE corporation が提供する OVAL 関連のプロジェクトが紹介されています。OVAL の関連情報、たとえば OVAL 言語の最新バージョン、数千にもなる OVAL 定義が用意された OVAL コンテンツのリポジトリーがあります。RHEL のスキャンには、Red Hat が提供する OVAL CVE コンテンツを使用することが推奨されます。
- Red Hat Satellite ドキュメント - このガイドセットでは、OpenSCAP を使用して複数のシステムでシステムセキュリティーを維持する方法などが説明されています。

第6章 AIDE で整合性のチェック

AIDE (Advanced Intrusion Detection Environment) は、システムのファイルのデータベースを作成し、そのデータベースを使用してファイルの整合性を確保し、システムの侵入を検出します。

6.1. AIDE のインストール

以下の手順は、**AIDE** をインストールして、そのデータベースを開始するのに必要です。

前提条件

- **AppStream** リポジトリーが有効になっている。

手順

1. **aide** パッケージをインストールするには、以下のコマンドを実行します。

```
# yum install aide
```

2. 初期データベースを生成するには、以下のコマンドを実行します。

```
# aide --init
```

注記

デフォルト設定では、**aide --init** コマンドは、**/etc/aide.conf** ファイルで定義するディレクトリーとファイルのセットのみを確認します。ディレクトリーまたはファイルを **AIDE** データベースに追加し、監視パラメーターを変更するには、**/etc/aide.conf** を変更します。

3. データベースの使用を開始するには、初期データベースのファイル名から末尾の **.new** を削除します。

```
# mv /var/lib/aide/aide.db.new.gz /var/lib/aide/aide.db.gz
```

4. **AIDE** データベースの場所を変更するには、**/etc/aide.conf** ファイルを編集して、**DBDIR** 値を変更します。追加のセキュリティーについては、データベース、設定、**/usr/sbin/aide** バイナリーファイルを、読み取り専用メディアなどの安全な場所に保存します。

6.2. AIDE を使用した整合性チェックの実行

前提条件

- **AIDE** が適切にインストールされ、そのデータベースが初期化されている。「AIDE のインストール」 を参照してください。

手順

1. 手動チェックを開始するには、以下を行います。

```
# aide --check
Start timestamp: 2018-07-11 12:41:20 +0200 (AIDE 0.16)
AIDE found differences between database and filesystem!!
```

```
...
[output truncated]
```

2. 最低でも、**AIDE** は毎週スキャンを実行するように設定する必要があります。**AIDE** は毎日実行する必要があります。たとえば、**AIDE** を毎日午前 04:05 に実行するようにスケジュールするには、**cron** コマンドを使用して、次の行を **/etc/crontab** ファイルを追加します。

```
05 4 * * * root /usr/sbin/aide --check
```

6.3. AIDE データベースの更新

システムの変更 (パッケージの更新、設定ファイルの修正など) を確認してから、基本となる **AIDE** データベースを更新することが推奨されます。

前提条件

- **AIDE** が適切にインストールされ、そのデータベースが初期化されている。「AIDE のインストール」 を参照してください。

手順

1. 基本となる **AIDE** データベースを更新します。

```
# aide --update
```

aide --update コマンドは、**/var/lib/aide/aide.db.new.gz** データベースファイルを作成します。

2. 整合性チェックで更新したデータベースを使用するには、ファイル名から従属文字列の **.new** を削除します。

6.4. 関連情報

AIDE の詳細は、以下のドキュメントを参照してください。

- **aide(1)** の man ページ
- AIDE マニュアル

第7章 LUKS を使用したブロックデバイスの暗号化

ディスクの暗号化は、それを暗号化することにより、ブロックデバイス上のデータを保護します。デバイスの復号したコンテンツにアクセスするには、パスフレーズまたは鍵を認証として提供する必要があります。これは、モバイルコンピューターや、リムーバブルメディアの場合に特に重要になります。これにより、デバイスをシステムから物理的に削除した場合でも、デバイスのコンテンツを保護するのに役立ちます。LUKS フォーマットは、Red Hat Enterprise Linux でブロックデバイスの暗号化のデフォルト実装です。

7.1. LUKS ディスクの暗号化

LUKS (Linux Unified Key Setup-on-disk-format) は、ブロックデバイスを暗号化でき、暗号化したデバイスの管理を簡素化するツールセットを提供します。LUKS を使用すれば、パーティションのバルク暗号化に使用されるマスター鍵を複数のユーザー鍵が複号できるようになります。

LUKS の機能

- LUKS はブロックデバイス全体を暗号化するため、脱着可能なストレージメディアやノート PC のディスクドライブといった、モバイルデバイスのコンテンツ保護に適しています。
- 暗号化されたブロックデバイスには任意のコンテンツを保持できます。これは、スワップデバイスの暗号化に役立ちます。また、とりわけデータストレージ用にフォーマットしたブロックデバイスを使用する特定のデータベースに関しても有用です。
- LUKS は既存のデバイスマッパーカーネルサブシステムを使用します。
- LUKS はパラフレーズの強化を提供し、辞書攻撃から保護します。
- LUKS デバイスには複数のキースロットが含まれ、ユーザーはこれを使用してバックアップキーやパスフレーズを追加できます。

LUKS が行わない こと

- LUKS は、多くのユーザーが、同じデバイスにアクセスする鍵をそれぞれ所有することが必要となるアプリケーションには適していません。LUKS1 形式は鍵スロットを 8 個提供し、LUKS2 形式は鍵スロットを最大 32 個提供します。
- LUKS は、ファイルレベルの暗号化を必要とするアプリケーションには適していません。

7.1.1. Red Hat Enterprise Linux における LUKS の実装

Red Hat Enterprise Linux は、LUKS を使用してファイルシステムを暗号化します。デフォルトではインストール時に、ファイルシステムを暗号化するオプションが指定されていません。ハードディスクを暗号化するオプションを選択すると、コンピューター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、パーティションの暗号化解読に用いられるバルク暗号化鍵を「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の変更を選択すると、暗号化するパーティションを選択できます。この設定は、パーティションテーブル設定で行われます。

Red Hat Enterprise Linux 8 では、デフォルトの形式は LUKS2 です。レガシーの LUKS (LUKS1) は完全にサポートされたままで、後方互換性の形で提供されます。LUKS2 形式は LUKS1 から着想を得ており、特定の状況で LUKS1 から変換できます。特に以下のシナリオでは、変換することはできません。

- LUKS1 デバイスが、Policy-Based Decryption (PBD - Clevis) ソリューションにより使用されているとマークされている。**cryptsetup** ツールは、**luksmeta** メタデータが検出されると、そのデバイスを変換することを拒否します。

- デバイスがアクティブになっている。デバイスが非アクティブ状態でなければ、変換することはできません。

LUKS2 形式は、バイナリー構造を変更することなく、さまざまな要素を将来更新できるように設計されています。LUKS2 は、内部的にメタデータに JSON テキスト形式を使用し、メタデータの冗長性を提供し、メタデータの破損を検出し、メタデータコピーからの自動修正を可能にします。

重要

LUKS1 だけをサポートする以前のシステムに準拠する必要がある実稼働システムでは、LUKS2 を使用しないでください。Red Hat Enterprise Linux 7 では、バージョン 7.6 以降が LUKS2 形式に対応していることに注意してください。

LUKS に使用されるデフォルトの暗号は **aes-xts-plain64** です。LUKS のデフォルトの鍵サイズは 256 ビットです。**Anaconda** (XTS モード) を使用した LUKS のデフォルトの鍵サイズは 512 ビットです。利用可能な暗号は以下のとおりです。

- AES (Advanced Encryption Standard) FIPS PUB 197
- Twofish (128 ビットブロック暗号)
- Serpent

関連資料

- LUKS プロジェクトのホームページ
- LUKS オンディスクフォーマットの仕様

7.2. 暗号化されていないデバイスのデータの暗号化

以下の手順には、暗号化されていないデバイスのデータを暗号化する手順が含まれます。

前提条件

- **cryptsetup-reencrypt** パッケージがインストールされている。
- データがバックアップされている。
- 暗号化するデバイスのファイルシステムがマウントされていない。

手順

警告

ハードウェア、カーネル、または人的ミスにより、暗号化プロセス時にデータが失われる場合があります。データの暗号化を開始する前に、信頼性の高いバックアップを作成しておいてください。

1. 暗号化するデバイスのデータのバックアップを作成します。

2. 以下のように、そのデバイスのファイルシステムをすべてアンマウントします。

```
# umount /dev/sdb1
```

3. LUKS ヘッダーを保存するための空き容量を確認します。以下のいずれかのオプションを選択します。

   A. 論理ボリュームを暗号化する場合は、以下のように、ファイルシステムのサイズを変更せずに、論理ボリュームを拡張できます。

```
# lvextend -L+8M vg00/lv00
```

   B. **parted** などのパーティション管理ツールを使用してパーティションを拡張します。

   C. このデバイスのファイルシステムを縮小します。ext2、ext3、または ext4 のファイルシステムには **resize2fs** ユーティリティーを使用できます。xfs ファイルシステムは縮小できないことに注意してください。

4. デバイスのヘッドに新しい LUKS ヘッダーを保存しつつ、ファイルシステムを暗号化します。たとえば、以下のコマンドでは、パスワード入力を求めたあと、暗号化処理を開始します。

```
# cryptsetup-reencrypt --new --reduce-device-size 8M /dev/sdb1
```

関連資料

- 詳細は、man ページの **cryptsetup-reencrypt(8)**、**cryptsetup(8)**、**lvextend(8)**, **resize2fs(8)**、および **parted(8)** を参照してください。

7.3. 別のファイルに LUKS ヘッダーを保存し、暗号化していないデバイスのデータの暗号化

以下の手順では、LUKS ヘッダーを保存する空き領域を作成せずにファイルシステムを暗号化する方法を説明します。ヘッダーは、追加のセキュリティー層としても使用できる、独立した場所に保存されます。

前提条件

- **cryptsetup-reencrypt** パッケージがインストールされている。

手順

警告

ハードウェア、カーネル、または人的ミスにより、暗号化プロセス時にデータが失われる場合があります。データの暗号化を開始する前に、信頼性の高いバックアップを作成しておいてください。

1. 暗号化するデバイスのデータのバックアップを作成します。

2. 以下のように、そのデバイスのファイルシステムをすべてアンマウントします。

```
# umount /dev/sdb1
```

3. **--header** パラメーターで取り外した LUKS ヘッダーを使用してファイルにパスを提供する際に **cryptsetup-reencrypt** を使用してファイルシステムを暗号化します。以下のコマンドを実行するとパスフレーズの入力が求められ、暗号化プロセスが開始します。

```
# cryptsetup-reencrypt --new --header /path/to/header /dev/sdb1
```

暗号化したデバイス (この場合は /dev/sdb1) を、たとえば後でロックを解除できるように、取り外した LUKS ヘッダーもアクセスできるようにする必要があります。

```
# cryptsetup open --header /path/to/header /dev/sdb1 my_crypt_device
```

関連資料

- 詳細は、man ページの **cryptsetup-reencrypt(8)** および **cryptsetup(8)** を参照してください。

第8章 ポリシーベースの複号を使用して暗号化ボリュームの自動アンロックの設定

ポリシーベースの複号 (PBD) は、物理マシンおよび仮想マシンにおいて、ハードドライブで暗号化した root ボリュームおよびセカンダリーボリュームのロックを解除できるようにする一連の技術です。PBD は、ユーザーパスワード、TPM (Trusted Platform Module) デバイス、システムに接続する PKCS#11 デバイス (たとえばスマートカード) などのさまざまな方法、もくしは特別なネットワークサーバーを使用します。

PBD を使用すると、ポリシーにさまざまなアンロック方法を組み合わせて、さまざまな方法で同じボリュームのロックを解除できるようにすることができます。Red Hat Enterprise Linux における PBD の現在の実装は、Clevis フレームワークと、**ピン** と呼ばれるプラグインから構成されます。各ピンは、個別のアンロック機能を提供します。現在利用できるピンは以下のとおりです。

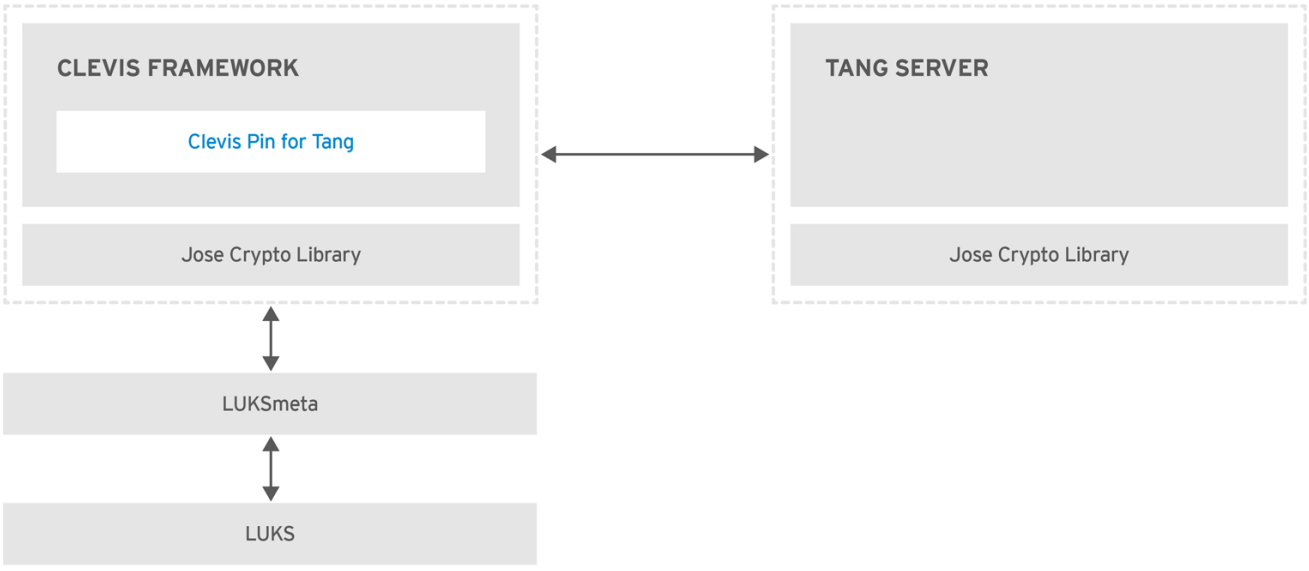
- **tang** - ネットワークサーバーを使用してボリュームのロックを解除
- **tpm2** - TPM2 ポリシーを使用してボリュームのロックを解除

NBDE (Network Bound Disc Encryption) は、特定のネットワークサーバーに暗号化ボリュームをバインドできるようにする PBD テクノロジーのサブカテゴリーです。NBDE の現在の実装には、Tang サーバーと、Tang サーバー用の Clevis ピンが含まれます。

8.1. NBDE (NETWORK-BOUND DISK ENCRYPTION)

Red Hat Enterprise Linux では、NBDE は、以下のコンポーネントおよびテクノロジーにより実装されます。

図8.1 LUKS1 で暗号化したボリュームを使用する場合の NBDE スキーム (LUKS2 ボリュームには luksmeta パッケージが使用されていない)



Tang は、ネットワークのプレゼンスにデータをバインドするためのサーバーです。セキュリティーが保護された特定のネットワークにシステムをバインドする際に利用可能なデータを含めるようにします。Tang はステートレスで、TLS または認証は必要ありません。サーバーが暗号鍵をすべて保存し、使用されたことがあるすべての鍵に関する知識を有するエスクローベースのソリューションとは異なり、Tang はクライアントの鍵と相互作用することはないため、クライアントから識別情報を得ることはありません。

Clevis は、自動化された復号用のプラグイン可能なフレームワークです。NBDE では、Clevis は、LUKS ボリュームの自動アンロックを提供します。**clevis** パッケージは、クライアントで使用される機能を提供します。

Clevis ピン は、Clevis フレームワークへのプラグインです。このようなピンの 1 つは、NBDE サーバー (Tang) との相互作用を実装するプラグインです。

Clevis および Tang は一般的なクライアントおよびサーバーの機能で、ネットワークがバインドされた暗号化を提供します。Clevis および Tang は、Red Hat Enterprise Linux で NBDE を実現するために、LUKS と組み合わせて、root および非 root のストレージボリュームの暗号化および複号に使用されます。

クライアントおよびサーバーのコンポーネントはともに **José** ライブラリーを使用して、暗号化および複号の操作を実行します。

NBDE のプロビジョニングを開始すると、Tang サーバーの Clevis ピンは、Tang サーバーの、アドバタイズされている非対称鍵の一覧を取得します。もしくは、鍵が非対称であるため、Tang の公開鍵の一覧を帯域外に配布して、クライアントが Tang サーバーにアクセスしなくても動作できるようにできます。このモードは **オフラインプロビジョニング** と呼ばれます。

Tang 用の Clevis ピンは、公開鍵のいずれかを使用して、固有で、暗号論的に強力な暗号鍵を生成します。この鍵を使用してデータを暗号化すると、この鍵は破棄されます。Clevis クライアントは、使いやすい場所に、このプロビジョニング操作で生成した状態を保存する必要があります。データを暗号化するこのプロセスは **プロビジョニング手順** と呼ばれています。

LUKS バージョン 2 (LUKS2) は、Red Hat Enterprise Linux 8 のデフォルトフォーマットであるため、NBDE のプロビジョニング状態は、LUKS2 ヘッダーにトークンとして保存されます。**luksmeta** パッケージによる、NBDE に対するプロビジョニング状態の活用は、LUKS1 で暗号化したボリュームにのみ使用されます。Tang 用の Clevis ピンは、仕様を必要とせずに LUKS1 と LUKS2 の両方をサポートします。

クライアントがそのデータにアクセスする準備ができると、プロビジョニング手順で生成したメタデータを読み込み、応答して暗号鍵を戻します。このプロセスは **復元手順** と呼ばれます。

Clevis は、NBDE ではピンを使用して LUKS ボリュームをバインドしているため、自動的にロックが解除されます。バインドプロセスが正常に終了すると、提供されている Dracut アンロックを使用してディスクをアンロックできます。

8.2. 暗号化クライアント (CLEVIS) のインストール

Clevis のプラグイン可能なフレームワークとピンを、暗号化したボリュームを使用するマシン (クライアント) にインストールするには、**root** で以下のコマンドを実行します。

```
# yum install clevis
```

データを複号するには、**clevis decrypt** コマンドを実行して、JWE (JSON Web Encryption) フォーマットで暗号文を提供します。

```
$ clevis decrypt < secret.jwe
```

関連資料

- クリックリファレンスは、組み込みの CLI ヘルプを参照してください。

  ```
  $ clevis
  ```

```
Usage: clevis COMMAND [OPTIONS]

  clevis decrypt      Decrypts using the policy defined at encryption time
  clevis encrypt sss  Encrypts using a Shamir's Secret Sharing policy
  clevis encrypt tang Encrypts using a Tang binding server policy
  clevis encrypt tpm2 Encrypts using a TPM2.0 chip binding policy

$ clevis decrypt
Usage: clevis decrypt < JWE > PLAINTEXT

Decrypts using the policy defined at encryption time

$ clevis encrypt tang
Usage: clevis encrypt tang CONFIG < PLAINTEXT > JWE

Encrypts using a Tang binding server policy

This command uses the following configuration properties:

  url: <string>   The base URL of the Tang server (REQUIRED)

  thp: <string>   The thumbprint of a trusted signing key

  adv: <string>   A filename containing a trusted advertisement
  adv: <object>   A trusted advertisement (raw JSON)

Obtaining the thumbprint of a trusted signing key is easy. If you
have access to the Tang server's database directory, simply do:

    $ jose jwk thp -i $DBDIR/$SIG.jwk

Alternatively, if you have certainty that your network connection
is not compromised (not likely), you can download the advertisement
yourself using:

    $ curl -f $URL/adv > adv.jws
```

- 詳細は、man ページの **clevis(1)** を参照してください。

8.3. TANG サーバーのデプロイメント

tang パッケージとその依存関係をインストールするには、**root** で以下のコマンドを実行します。

```
# yum install tang
```

systemd を使用して、**tangd** サービスを有効にして開始します。

```
# systemctl enable tangd.socket --now
Created symlink from /etc/systemd/system/multi-user.target.wants/tangd.socket to
/usr/lib/systemd/system/tangd.socket.
```

tangd が、**systemd** のソケットアクティベーションメカニズムを使用しているため、最初に接続するとすぐにサーバーが起動します。最初の起動時に、一組の暗号鍵が自動的に生成されます。

鍵の手動生成などの暗号化操作を実行するには、**jose** ユーティリティーを使用します。詳細は **jose -h** コマンドを実行するか、man ページの **jose(1)** を参照してください。

例8.1 Tang 鍵の変更

鍵を定期的に変更することが重要です。鍵を変更するのに適した間隔は、アプリケーション、鍵サイズ、および組織のポリシーにより異なります。一般的な推奨事項は 「Cryptographic Key Length Recommendation」 ページを参照してください。

鍵を変更するには、最初に鍵データベースディレクトリー (通常は **/var/db/tang**) に新しい鍵を生成します。たとえば、以下のコマンドを使用して、新しい署名を作成し、鍵を交換します。

```
# DB=/var/db/tang
# jose jwk gen -i '{"alg":"ES512"}' -o $DB/new_sig.jwk
# jose jwk gen -i '{"alg":"ECMR"}' -o $DB/new_exc.jwk
```

アドバタイズメントから見えなくなるように、古い鍵の名前の先頭に **.** を付けます。以下の例のファイル名は、鍵データベースディレクトリーに実在する固有のファイル名とは異なります。

```
# mv $DB/old_sig.jwk $DB/.old_sig.jwk
# mv $DB/old_exc.jwk $DB/.old_exc.jwk
```

Tang は、直ちにすべての変更を適用します。再起動は必要ありません。

この時点で、新しいクライアントバインディングは新しい鍵を選択し、以前のクライアントは古い鍵を使用し続けます。すべてのクライアントが新しい鍵を使用することを確認すると、古い鍵を削除できます。

警告

クライアントが使用している最中に古い鍵を削除すると、データが失われる場合があります。

Tang は、通信にポート 80 を使用します。このポートは Web サーバーにも広く使用されています。Tang のポート番号を変更するには、標準の **systemd** メカニズムを使用して **tangd.socket** ユニットファイルを上書きします。

8.3.1. 高可用性システムのデプロイメント

Tang は、高可用性デプロイメント構築する方法を 2 つ提供します。

1. **クライアントの冗長性 (推奨)**
   クライアントは、複数の Tang サーバーにバインドする機能を使用して設定する必要があります。この手順では、各 Tang サーバーに独自の鍵があり、クライアントは、このサーバーのサブセットに接続することで複号できるようになります。Clevis では、以前からこのワークフローを **sss** プラグインによりサポートしています。

   この設定の詳細は、以下の man ページを参照してください。

- **tang(8)** の高可用性 (High Availability) セクション
- **clevis(1)** のシャミアの秘密共有 (Shamir's Secret Sharing) のセクション
- **clevis-encrypt-sss(1)**
  Red Hat は、高可用性デプロイメントにこの方法を使用することを推奨します。

2. **鍵の共有**
   冗長性のために、Tang のインスタンスは複数デプロイできます。2 つ目以降のインスタンスを設定するには、**tang** パッケージをインストールし、SHH で **rsync** を使用して、重要なディレクトリーを新しいホストにコピーします。鍵を共有すると、重大な不正アクセスのリスクを増やし、別の自動化インフラストラクチャーが必要となるため、Red Hat はこの方法を推奨しません。

8.4. TANG を使用する NBDE システムへの暗号化クライアントのデプロイメント

以下の手順は、Tang ネットワークサーバーを使用して、暗号化したボリュームの自動ロック解除を設定する手順を説明します。

前提条件

- Clevis フレームワークがインストールされている。「暗号化クライアント (Clevis) のインストール」 を参照してください。
- Tang サーバーが利用できる。「Tang サーバーのデプロイメント」 を参照してください。

手順

1. Clevis 暗号化クライアントを Tang サーバーにバインドするには、**clevis encrypt tang** サブコマンドを使用します。

```
$ clevis encrypt tang '{"url":"http://tang.srv"}' < input-plain.txt > secret.jwe
The advertisement contains the following signing keys:

_OsIk0T-E2l6qjfdDiwVmidoZjA

Do you wish to trust these keys? [ynYN] y
```

   この例の URL (**http://tang.srv**) を、**tang** がインストールされているサーバーの URL に変更します。**secret.jwe** 出力ファイルには、JSON Web の暗号形式で暗号化した暗号文が含まれます。この暗号文は **input-plain.txt** 入力ファイルから読み込まれます。

2. データを複号するには、**clevis decrypt** コマンドを実行して、暗号文 (JWE) を提供します。

```
$ clevis decrypt < secret.jwe > output-plain.txt
```

関連資料

- クイックリファレンスは、man ページの **clevis-encrypt-tang(1)** か、組み込みの CLI ヘルプを使用します。

```
$ clevis
Usage: clevis COMMAND [OPTIONS]
```

```
  clevis decrypt      Decrypts using the policy defined at encryption time
  clevis encrypt http Encrypts using a REST HTTP escrow server policy
  clevis encrypt sss  Encrypts using a Shamir's Secret Sharing policy
  clevis encrypt tang Encrypts using a Tang binding server policy
  clevis encrypt tang Encrypts using a Tang binding server policy
  clevis luks bind    Binds a LUKSv1 device using the specified policy
  clevis luks unlock  Unlocks a LUKSv1 volume

$ clevis decrypt
Usage: clevis decrypt < JWE > PLAINTEXT

Decrypts using the policy defined at encryption time

$ clevis encrypt tang
Usage: clevis encrypt tang CONFIG < PLAINTEXT > JWE

Encrypts using a Tang binding server policy

This command uses the following configuration properties:

  url: <string>   The base URL of the Tang server (REQUIRED)

  thp: <string>   The thumbprint of a trusted signing key

  adv: <string>   A filename containing a trusted advertisement
  adv: <object>   A trusted advertisement (raw JSON)

Obtaining the thumbprint of a trusted signing key is easy. If you
have access to the Tang server's database directory, simply do:

    $ jose jwk thp -i $DBDIR/$SIG.jwk

Alternatively, if you have certainty that your network connection
is not compromised (not likely), you can download the advertisement
yourself using:

    $ curl -f $URL/adv > adv.jws
```

- 詳細は、以下の man ページを参照してください。
- **clevis(1)**
- **clevis-luks-unlockers(7)**

8.5. TPM 2.0 ポリシーを使用した暗号化クライアントのデプロイメント

以下の手順は、Trusted Platform Module 2.0 (TPM 2.0) ポリシーを使用して、暗号化したボリュームの自動ロック解除を設定する手順を説明します。

前提条件

- Clevis フレームワークがインストールされている。「暗号化クライアント (Clevis) のインストール」 を参照してください。
- システムが 64 ビット Intel アーキテクチャー、または 64 ビット AMD アーキテクチャーである。

手順

1. TPM 2.0 チップを使用して暗号化するクライアントをデプロイするには、JSON 設定オブジェクトフォーマットの引数のみが使用されている **clevis encrypt tpm2** サブコマンドを使用します。

```
$ clevis encrypt tpm2 '{}' < input-plain.txt > secret.jwe
```

別の階層、ハッシュ、および鍵アルゴリズムを選択するには、以下のように、設定プロパティーを指定します。

```
$ clevis encrypt tpm2 '{"hash":"sha1","key":"rsa"}' < input-plain.txt > secret.jwe
```

2. データを復号するには、JSON Web Encryption (JWE) 形式の暗号文を提供します。

```
$ clevis decrypt < secret.jwe > output-plain.txt
```

ピンは、PCR (Platform Configuration Registers) 状態へのデータのシーリングもサポートします。このように、PCP ハッシュ値が、シーリング時に使用したポリシーと一致する場合にのみ、データのシーリングを解除できます。

たとえば、SHA-1 バンクに対して、インデックス 0 および 1 の PCR にデータをシールするには、以下を行います。

```
$ clevis encrypt tpm2 '{"pcr_bank":"sha1","pcr_ids":"0,1"}' < input-plain.txt > secret.jwe
```

関連資料

- 詳細と、可能な設定プロパティーの一覧は、man ページの **clevis-encrypt-tpm2(1)** を参照してください。

8.6. LUKS で暗号化した ROOT ボリュームの手動登録の設定

1. LUKS で暗号化した既存の root ボリュームを自動的にアンロックするには、サブパッケージの **clevis-luks** をインストールします。

```
# yum install clevis-luks
```

2. PBD 用 LUKS 暗号化ボリュームを特定します。次の例では、ブロックデバイスは **/dev/sda2** と呼ばれています。

```
# lsblk
NAME                                          MAJ:MIN RM   SIZE RO TYPE  MOUNTPOINT
sda                                             8:0    0    12G  0 disk
├─sda1                                          8:1    0     1G  0 part  /boot
└─sda2                                          8:2    0    11G  0 part
  └─luks-40e20552-2ade-4954-9d56-565aa7994fb6 253:0    0    11G  0 crypt
    ├─rhel-root                               253:0    0   9.8G  0 lvm   /
    └─rhel-swap                               253:1    0   1.2G  0 lvm   [SWAP]
```

3. **clevis luks bind** コマンドを使用して、ボリュームを Tang サーバーにバインドします。

```
# clevis luks bind -d /dev/sda2 tang '{"url":"http://tang.srv"}'
```

```
The advertisement contains the following signing keys:

_OsIk0T-E2l6qjfdDiwVmidoZjA

Do you wish to trust these keys? [ynYN] y
You are about to initialize a LUKS device for metadata storage.
Attempting to initialize it may result in data loss if data was
already written into the LUKS header gap in a different format.
A backup is advised before initialization is performed.

Do you wish to initialize /dev/sda2? [yn] y
Enter existing LUKS password:
```

このコマンドは、以下の 4 つの手順を実行します。

a. LUKS マスター鍵と同じエントロピーを使用して、新しい鍵を作成します。

b. Clevis で新しい鍵を暗号化します。

c. LUKS2 ヘッダートークンに Clevis JWE オブジェクトを保存するか、デフォルト以外の LUKS1 ヘッダーが使用されている場合は LUKSMeta を使用します。

d. LUKS を使用する新しい鍵を有効にします。

注記

バインド手順では、空き LUKS パスワードスロットが少なくとも 1 つあることが前提となっています。そのスロットの 1 つを **clevis luks bind** コマンドが使用します。

4. ボリュームは、現在、既存のパスワードと Clevis ポリシーを使用してロックを解除できます。

5. Clevis JWE オブジェクトが LUKS2 ヘッダートークンに適切に配置されていることを確認するには、**cryptsetup luksDump** コマンドを使用します。

```
# cryptsetup luksDump /dev/sda2
Tokens:
  0: clevis
    Keyslot:  1
```

LUKS1 ヘッダーの場合は、**luksmeta show** コマンドを使用します。

```
# luksmeta show -d /dev/sda2
0   active empty
1   active cb6e8904-81ff-40da-a84a-07ab9ab5715e
2 inactive empty
3 inactive empty
4 inactive empty
5 inactive empty
6 inactive empty
7 inactive empty
```

6. システムの起動プロセスの初期段階でディスクバインディングを処理するようにするには、インストール済みのシステムで以下のコマンドを実行します。

```
# yum install clevis-dracut
# dracut -fv --regenerate-all
```

重要

(DHCP を使用しない) 静的な IP 設定を持つクライアントに NBDE を使用するには、以下のように、手動でネットワーク設定を dracut ツールに渡します。

```
# dracut -fv --regenerate-all --kernel-cmdline
"ip=192.0.2.10::192.0.2.1:255.255.255.0::ens3:none:192.0.2.45"
```

もしくは、以下のように、静的ネットワーク情報を使用して **/etc/dracut.conf.d/** ディレクトリーに .conf ファイルを作成します。

```
# cat /etc/dracut.conf.d/static_ip.conf
kernel_cmdline="ip=192.0.2.10::192.0.2.1:255.255.255.0::ens3:none:192.0.2.45"
```

初期 RAM ディスクイメージを再生成します。

```
# dracut -fv --regenerate-all
```

詳細は、man ページの **dracut.cmdline(7)** を参照してください。

関連資料
詳細は、以下の man ページを参照してください。

- **clevis-luks-bind(1)**

8.7. キックスタートを使用して、LUKS で暗号化した ROOT ボリュームの自動登録の設定

Clevis は、登録プロセスを完全に自動化するために、キックスタートと統合できます。

1. root パーティションが、一時的なパスワードを使用して、LUKS 暗号化を有効にしているディスクを分割するように、キックスタートに指示します。パスワードは、登録プロセスに使用するための一時的なものです。

```
part /boot --fstype="xfs" --ondisk=vda --size=256
part / --fstype="xfs" --ondisk=vda --grow --encrypted --passphrase=temppass
```

2. 関連する Clevis パッケージを **%packages** セクションに追加して、インストールします。

```
%packages
clevis-dracut
%end
```

3. **clevis luks bind** を呼び出して、**%post** セクションのバインディングを実行します。その後、一時パスワードを削除します。

```
%post
clevis luks bind -f -k- -d /dev/vda2 \
tang '{"url":"http://tang.srv","thp":"_OsIk0T-E2l6qjfdDiwVmidoZjA"}' \ <<< "temppass"
```

```
cryptsetup luksRemoveKey /dev/vda2 - <<< "temppass"
%end
```

上記の例では、バインディングの設定で、Tang サーバーで信頼するサムプリントを指定することで、バインディングを完全に非対話にします。

Tang サーバーの代わりに TPM 2.0 ポリシーを使用する場合は、同様の手順を使用できます。

8.8. LUKS で暗号化されたリムーバブルストレージデバイスの自動アンロックの設定

1. USB ドライブなど、LUKS で暗号化したリムーバブルストレージデバイスを自動的にアンロックするには、**clevis-udisks2** パッケージをインストールします。

```
# yum install clevis-udisks2
```

2. システムを再起動し、「LUKS で暗号化した root ボリュームの手動登録の設定」 に従って、**clevis luks bind** コマンドを使用したバインディング手順を実行します。以下に例を示します。

```
# clevis luks bind -d /dev/sdb1 tang '{"url":"http://tang.srv"}'
```

3. LUKS で暗号化したリムーバブルデバイスは、GNOME デスクトップセッションで自動的にアンロックできるようになりました。Clevis ポリシーにバインドするデバイスは、**clevis luks unlock** コマンドでアンロックできます。

```
# clevis luks unlock -d /dev/sdb1
```

Tang サーバーの代わりに TPM 2.0 ポリシーを使用する場合は、同様の手順を使用できます。

関連資料
詳細は、以下の man ページを参照してください。

- **clevis-luks-unlockers(7)**

8.9. システムの起動時に LUKS で暗号化した非 ROOT ボリュームに自動アンロックの設定

LUKS で暗号化した非 root ボリュームをアンロックするのに NBDE を使用するには、以下の手順を行います。

1. clevis-systemd パッケージをインストールします。

```
# yum install clevis-systemd
```

2. Clevis のアンロックサービスを有効にします。

```
# systemctl enable clevis-luks-askpass.path
Created symlink from /etc/systemd/system/remote-fs.target.wants/clevis-luks-askpass.path to
/usr/lib/systemd/system/clevis-luks-askpass.path.
```

3. 「LUKS で暗号化した root ボリュームの手動登録の設定」 の説明通りに、**clevis luks bind** コマンドを使用したバインディング手順を実行します。

4. システムの起動時に暗号化したブロックデバイスを設定するには、**_netdev** オプションに相当する行を **/etc/crypttab** 設定ファイルに追加します。詳細は man ページの **crypttab(5)** を参照してください。

5. **/etc/fstab** ファイルでアクセス可能なファイルシステムの一覧にボリュームを追加します。この設定で **_netdev** オプションを使用します。詳細は、man ページの **fstab(5)** を参照してください。

関連資料
詳細は、以下の man ページを参照してください。

- **clevis-luks-unlockers(7)**

8.10. NBDE ネットワークで仮想マシンのデプロイメント

clevis luks bind コマンドは、LUKS マスターキーを変更しません。これは、仮想マシンまたはクラウド環境で使用する LUKS で暗号化したイメージを作成する場合に、このイメージを実行するすべてのインスタンスがマスター鍵を共有することを意味します。これにはセキュリティーの観点で大きな問題があるため、常に回避する必要があります。

これは、Clevis の制限ではなく、LUKS の設計原理です。クラウドに暗号化された root ボリュームが必要な場合は、クラウドの Red Hat Enterprise Linux の各インスタンスにインストールプロセスを実行できるようにする (通常はキックスタートを使用) 必要があります。このイメージは、LUKS マスターキーを共有しなければ共有できません。

仮想化環境に自動アンロックをデプロイする場合は、キックスタートファイルを使用して lorax、virt-install などのシステムを使用すること (「キックスタートを使用して、LUKS で暗号化した root ボリュームの自動登録の設定」 を参照)、または暗号化した各仮想マシンに固有のマスター鍵があるようにする自動プロビジョニングツールを使用することを Red Hat は強く推奨します。

TPM 2.0 ポリシーを使用した自動ロック解除は、仮想マシンではサポートされていないことに注意してください。

関連資料
詳細は、以下の man ページを参照してください。

- **clevis-luks-bind(1)**

8.11. NBDE を使用してクラウド環境に自動的に登録可能な仮想マシンイメージの構築

自動登録可能な暗号化イメージをクラウド環境にデプロイすると、特有の課題が発生する可能性があります。上で詳述した他の仮想化環境と同様に、LUKS マスター鍵が共有されないように、イメージのインスタンス化は何度も行わないでください。したがって、lorax、virt-install などの自動デプロイメントシステムと Kickstart ファイルをともに使用して、イメージのビルドプロセス時にマスター鍵の一意性を保証する必要があります。

クラウド環境では、ここで検討する 2 つの Tang サーバーデプロイメントオプションが利用できます。まず、クラウド環境そのものに Tang サーバーをデプロイできます。もしくは、2 つのインフラストラクチャー間で VPN リンクを使用した独立したインフラストラクチャーで、クラウドの外に Tang サーバーをデプロイできます。

クラウドに Tang をネイティブにデプロイすると、簡単にデプロイできます。ただし、別のシステムの暗号文のデータ永続化層でインフラストラクチャーを共有します。Tang サーバーの秘密鍵および Clevis メタデータは、同じ物理ディスクに保存できる場合があります。この物理ディスクでは、暗号文データへのいかなる不正アクセスが可能になります。

重要

このため、Red Hat は、データを保存する場所と、Tang が実行しているシステムを、物理的に分離させることを強く推奨します。クラウドと Tang サーバーを分離することで、Tang サーバーの秘密鍵が、Clevis メタデータと誤って結合することがないようにします。さらに、これにより、クラウドインフラストラクチャーが危険にさらされている場合に、Tang サーバーのローカル制御を提供します。

8.12. 関連情報

詳細は、以下の man ページを参照してください。

- **tang(8)**
- **clevis(1)**
- **jose(1)**
- **clevis-luks-unlockers(1)**

第9章 システムの監査

Audit は、追加のセキュリティー機能をシステムに提供しません。システムで使用するセキュリティーポリシーの違反を発見するために使用できます。このような違反は、SELinux などの別のセキュリティー対策で防ぐことができます。

9.1. LINUX AUDIT

Linux Audit システムは、システムのセキュリティー関連情報を追跡する方法を提供します。事前設定されたルールに基づき、Audit は、システムで発生しているイベントに関する情報をできるだけ多く記録するのに使用するログエントリーを生成します。この情報は、ミッションクリティカルな環境でセキュリティーポリシーの違反者と、違反者によるアクションを判断する上で必須のものです。

以下は、Audit がログファイルに記録できる情報の概要です。

- イベントの日時、タイプ、結果
- サブジェクトとオブジェクトの機密性のラベル
- イベントを開始したユーザーの ID とイベントの関連性
- Audit 設定の全修正および Audit ログファイルへのアクセス試行
- SSH、Kerberos、およびその他の認証メカニズムの使用のすべて
- 信頼できるデータベース (**/etc/passwd** など) への変更
- システムからの情報のインポート、およびシステムへの情報のエクスポートの試行
- ユーザー ID、サブジェクトおよびオブジェクトラベル、その他の属性に基づく include または exclude イベント

Audit システムの使用は、多くのセキュリティー関連の認定における要件でもあります。Audit は、以下の認定またはコンプライアンスガイドの要件に合致するか、それを超えるように設計されています。

- Controlled Access Protection Profile (CAPP)
- Labeled Security Protection Profile (LSPP)
- Rule Set Base Access Control (RSBAC)
- NISPOM (National Industrial Security Program Operating Manual)
- Federal Information Security Management Act (FISMA)
- PCI DSS (Payment Card Industry Data Security Standard)
- セキュリティー技術実装ガイド (STIG: Security Technical Implementation Guide)

Audit は以下でも認定されています。

- National Information Assurance Partnership (NIAP) および Best Security Industries (BSI) による評価
- Red Hat Enterprise Linux 5 における LSPP/CAPP/RSBAC/EAL4+ の認定

- Red Hat Enterprise Linux 6 における Operating System Protection Profile / Evaluation Assurance Level 4+ (OSPP/EAL4+) の認定

使用例

ファイルアクセスの監視

Audit は、ファイルやディレクトリーがアクセス、修正、実行されたか、またはファイル属性が変更されたかを追跡できます。これはたとえば、重要なファイルへのアクセスを検出し、これらのファイルが破損した場合に監査証跡を入手可能とする際に便利なものです。

システムコールの監視

Audit は、よく使用される一部のシステムコールが使用されるたびにログエントリーを生成するように設定できます。これを使用すると、**settimeofday** や **clock_adjtime**、その他の時間関連のシステムコールを監視することで、システム時間への変更を追跡できます。

ユーザーが実行したコマンドの記録

Audit はファイルが実行されたかどうかを追跡できるため、特定のコマンドの実行を毎回記録するようにルールを定義できます。たとえば、**/bin** ディレクトリー内のすべての実行可能ファイルにルールを定義できます。その結果により作成されるログエントリーをユーザー ID で検索すると、ユーザーごとに実行されたコマンドの監査証跡を生成できます。

システムのパス名実行の記録

ルールの呼び出し時にパスを inode に変換するファイルアクセスをウォッチする以外に、ルールの呼び出し時に存在しない場合や、ルールの呼び出し後にファイルが置き換えられた場合でも、Audit がパスの実行をウォッチできるようになりました。これにより、ルールは、プログラム実行ファイルをアップグレードした後、またはインストールされる前にも機能を継続できます。

セキュリティーイベントの記録

pam_faillock 認証モジュールは、失敗したログイン試行を記録できます。Audit で失敗したログイン試行も記録するように設定すると、ログインを試みたユーザーに関する追加情報が提供されます。

イベントの検索

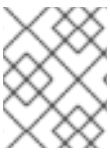
Audit は **ausearch** ユーティリティーを提供します。これを使用すると、ログエントリーをフィルターにかけ、いくつもの条件に基づく完全な監査証跡を提供できます。

サマリーレポートの実行

aureport ユーティリティーを使用すると、記録されたイベントのデイリーレポートを生成できます。システム管理者は、このレポートを分析し、疑わしいアクティビティーをさらに調べることができます。

ネットワークアクセスの監視

iptables ユーティリティーおよび **ebtables** ユーティリティーは、Audit イベントを発生するように設定できるため、システム管理者がネットワークアクセスを監視できるようになります。



注記

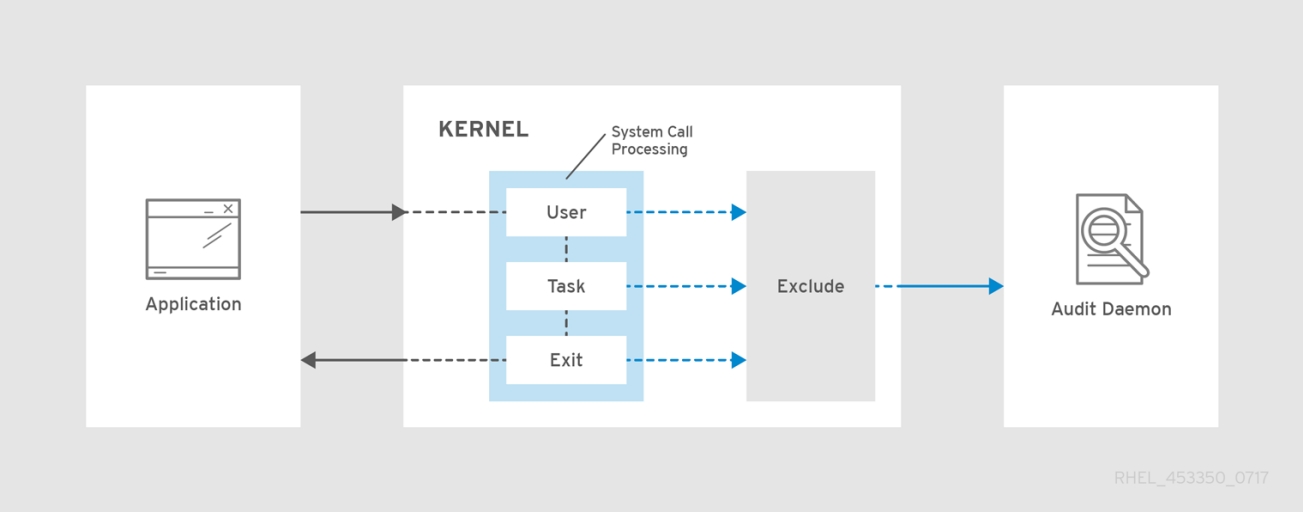
システムのパフォーマンスは、Audit が収集する情報量によって影響される可能性があります。

9.1.1. Audit システムのアーキテクチャー

Audit システムは、ユーザースペースアプリケーションおよびユーティリティーと、カーネル側のシステムコール処理という 2 つの主要パートで構成されます。カーネルコンポーネントは、ユーザースペースアプリケーションからシステムコールを受け、これを **user**、**task**、または **exit** のいずれかのフィル

ターで振り分けます。システムが **exclude** フィルターを通過すると、前述のフィルターの 1 つに送られます。このフィルターは Audit ルール設定に基づいて、システムコールを Audit デーモンに送信してさらに処理します。図9.1「Audit システムのアーキテクチャー」 にこのプロセスを示します。

図9.1 Audit システムのアーキテクチャー



ユーザースペースの Audit デーモンはカーネルから情報を収集し、ログファイルエントリーを作成します。他のユーザースペースユーティリティーは、Audit デーモン、カーネル Audit コンポーネント、または Audit ログファイルと対話します。

- **auditctl** - Audit 制御ユーティリティーはカーネル Audit コンポーネントと対話し、ルールを管理するだけでなくイベント生成プロセスの多くの設定やパラメーターも制御します。
- 残りの Audit ユーティリティーは Audit ログファイルのコンテンツを入力として受け取り、ユーザーの要件に基づいて出力を生成します。たとえば、**aureport** ユーティリティーは記録された全イベントのレポートを生成します。

Audit dispatcher デーモン (**audisp**) 機能は、Audit デーモン (**auditd**) に統合されるようになりました。監査イベントと、リアルタイムの分析プログラムの対話のためのプラグイン設定ファイルは、デフォルトで **/etc/audit/plugins.d/** ディレクトリーに保存されます。

9.2. 関連情報

Audit システムの詳細は、以下の資料を参照してください。

オンラインのリソース

- Linux Audit ドキュメントのプロジェクトページ - https://github.com/linux-audit/audit-documentation/wiki

インストールされているドキュメント

audit パッケージが提供するドキュメンテーションは、**/usr/share/doc/audit/** ディレクトリーにあります。

man ページ

- **audispd.conf**(5)
- **auditd.conf**(5)

- **ausearch-expression**(5)
- **audit.rules**(7)
- **audispd**(8)
- **auditctl**(8)
- **auditd**(8)
- **aulast**(8)
- **aulastlog**(8)
- **aureport**(8)
- **ausearch**(8)
- **ausyscall**(8)
- **autrace**(8)
- **auvirt**(8)