Red Hat Enterprise Linux 8

セキュリティーの強化

Red Hat Enterprise Linux 8 の保護
Red Hat Enterprise Linux 8 セキュリティーの強化

Red Hat Enterprise Linux 8 の保護
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ます。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計画および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 (英語のみ) ................................. 4

第1章 RHEL におけるセキュリティーの強化の概要 ............................................... 5
  1.1. コンピューターセキュリティーとは .......................................................... 5
  1.2. セキュリティーの標準化 ................................................................. 5
  1.3. 暗号化ソフトウェアおよび認定 ......................................................... 5
  1.4. セキュリティーコントロール .......................................................... 6
  1.5. 脆弱性のアセスメント ............................................................... 7
  1.6. セキュリティーへの脅威 ............................................................... 9
  1.7. 一般的な不正使用と攻撃 ........................................................... 12

第2章 インストール時の RHEL の保護 ............................................................ 17
  2.1. BIOS および UEFI のセキュリティー .................................................. 17
  2.2. ディスクのパーティション設定 ..................................................... 17
  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 ............................. 18
  2.4. 必要なパッケージの最小限のインストール ....................................... 18
  2.5. インストール後の手順 ............................................................... 18

第3章 システム全体の暗号化ポリシーの使用 .................................................. 20
  3.1. システム全体の暗号化ポリシー ..................................................... 20
  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 22
  3.3. FIPS モードへのシステムの切り替え ............................................. 23
  3.4. システム全体の暗号化ポリシーに従わないようにアプリケーションを除外 23
  3.5. 関連情報 ............................................................... 24

第4章 PKCS #11 で暗号化ハードウェアを使用するようにアプリケーションを設定 .......... 26
  4.1. PKCS #11 による暗号化ハードウェアへの対応 ..................................... 26
  4.2. スマートカードに保存された SSH 鍵の使用 ....................................... 26
  4.3. APACHE および NGINX で秘密鍵を保護する HSM の使用 ..................... 28
  4.4. スマートカードから証明書を使用して認証するアプリケーションの設定 28
  4.5. 関連情報 ............................................................... 29

第5章 共通システム証明書の使用 .............................................................. 30
  5.1. システム全体でトラストストアの使用 ............................................. 30
  5.2. 新しい証明書の追加 ............................................................... 30
  5.3. 信頼されているシステム証明書の管理 ............................................. 31
  5.4. 関連情報 ............................................................... 32

第6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 ......................... 33
  6.1. RHEL におけるセキュリティーコンプライアンスツール ......................... 33
  6.2. RED HAT SECURITY ADVISORIES OVAL フィード ......................... 34
  6.3. システムの脆弱性スキャン ......................................................... 34
  6.4. リモートシステムの脆弱性のスキャン ............................................. 35
  6.5. セキュリティーコンプライアンスのプロファイルの表示 ........................... 36
  6.6. 特定のベースラインによるセキュリティーコンプライアンスの評価 ............. 37
  6.7. OSPP に合わせてシステムの修復 .................................................. 38
  6.8. SCAP WORKBENCH を使用したカスタムプロファイルでシステムのスキャン 39
  6.9. インストール後にセキュリティーコンプライアンスを統一するシステムのデプロイメント 43
  6.10. 関連情報 .............................................................. 45

第7章 AIDE で整合性の確認 ................................................................. 47
  7.1. AIDE のインストール .............................................................. 47
  7.2. AIDE を使用した整合性チェックの実行 ........................................ 47
第8章 LUKSを使用したブロックデバイスの暗号化
8.1. LUKSディスクの暗号化
8.2. 暗号化されていないデバイスのデータの暗号化
8.3. 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

第9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
9.1. NBDE (NETWORK-BOUND DISK ENCRYPTION)
9.2. 暗号化クライアント(CLEVIS)のインストール
9.3. SELINUXをENFORCINGモードで有効にしたTANGサーバーのデプロイメント
9.4. TANG鍵の変更
9.5. 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
9.6. 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
9.7. LUKSで暗号化したROOTボリュームの手動登録の設定
9.8. 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
9.9. 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
9.10. 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
9.11. NBDEネットワークで仮想マシンのデプロイメント
9.12. 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
9.13. 関連情報

第10章 システムの監査
10.1. LINUXのAUDIT
10.2. AUDITシステムのアーキテクチャー
10.3. セキュアな環境へのAUDITDの設定
10.4. AUDITDの開始および制御
10.5. AUDITログファイルについて
10.6. AUDITCTLでAUDITルールを定義および実行
10.7. 永続的なAUDITルールの定義
10.8. 事前に設定されたルールファイルの使用
10.9. 永続ルールを定義するAUGENRULESの使用
10.10. 関連情報
RED HAT ドキュメントへのフィードバック (英語のみ)

ご意見ご要望をお聞かせください。ドキュメントの改善点はございませんか。改善点を報告する場合は、以下のように行います。

- 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 RHELにおけるセキュリティーの強化の概要

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

にも関わらず、多くの組織（個々のユーザーも含む）は、機能性、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、セキュリティーを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。したがって、適切なセキュリティーの確保は、無許可の侵入が発生してはじめて徹底されることも少なくありません。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)、投資利益率 (Return on Investment: ROI)、サービスの品質 (Quality of Service: QoS) などの用語や評価指標は日常的なビジネス用語として用いられるようになっています。各種の業界が、計画およびプロセス管理コストの一環として、これらの評価指標を用いてデータ保全性や可用性などを算出しています。電子商取引などの業界では、データの可用性と信頼性は、成功と失敗の違いを意味します。

1.2. 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)、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モデルを説明します。

- 機密性 - 機密情報は、事前に定義された個人だけが利用できるようにする必要があります。許可されていない情報の送信や使用は、制限する必要があります。たとえば、情報に機密性があれば、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時にいつでもアクセスできるようにする必要があります。可用性は、合意した頻度とタイミングで情報を入手できることを保証します。これは、パーセンテージで表されることが多い、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か選択されているか」、「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にどのように対応しているか」、および「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とおりです。

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

1.4.2. 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の例は以下のとおりです。

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

- 技術の設定、監視、および保守を行うスタッフの専門知識
- サービスとカーネルのバッチ、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精通している人材を確保することはできますが、多くの分野を専門とするスタッフを確保することは容易ではありません。これは、情報セキュリティーの各専門分野で、継続的な注意と重点が必要となるためです。情報セキュリティーは、常に変化しています。

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。基本的にはこのフェーズでは、対象を絞り、すべての既知の脆弱性を調べます。調査フェーズが終わると報告フェーズになります。ここでは、調査結果が高中低のカテゴリーに分類され、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、外部からシステムに攻撃を試みます。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。プリンターサーバー、ファイルサーバー、データベースなどのリソースを見ることができます。

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

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

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

セキュリティー管理者の力量は、使用するツールとその管理者が有する知識で決まります。現在使用できるアセスメントツールのいずれかを選び、それらをシステムに対して実行すると、ほぼ間違いなく誤検出がいくつか見つかります。プログラム障害でもユーザーエラーでも、結果は同じです。ツールは、誤検出することもあるので、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」とは何を指していますか？1台のサーバー、またはネットワーク全体およびネットワーク内にあるすべてのサーバーを確認しますか？会社外ですか？それとも内部ですか？この質問に対する回答は、選択したツールだけでなく、そのツールの使用方法を決定する際に重要です。

方法論の確立の詳細は、以下の Web サイトを参照してください。

- 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「AIDE で整合性のチェック」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な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ています。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、ハブやルーターがデータパケットのブロードキャストを送信します。この方式は、外部侵入者やローカルホストの未認証ユーザーが仕掛けるアドレス解決プロトコル（ARP）およびメディアアクセスコントロール（MAC）アドレスの偽装に対して最も脆弱です。

集中化サーバー

ネットワーキングのもう一つの落とし穴は、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一台の強力なマシンに統合しています。集中化は、複数サーバーを設定するよりも管理が簡単で、コストを大幅に削減できるので便利です。ただし、集中化されたサーバーはネットワークにおける単一障害点となります。中央のサーバーが攻撃されると、ネットワークが完全に使用できなくなるか、データの不正操作や盗難が起きやすくなる可能性があります。このような場合は、中央サーバーがネットワーク全体へのアクセスを許可することになります。

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じ様に重要です。サーバーが攻撃されると、クラッカーが意のままにすべてのコンテンツを盗んだり、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、主要な問題の一部を詳述します。

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パッケージが1000個以上含まれます。ただし、サーバー管理者が、ディストリビューションに含まれるすべての個別パッケージをインストールすることはありません。代わりに、複数のサーバーアプリケーションを含むパッケージのベースインストールを行います。

システム管理者は、インストールに含まれるプログラムに注意を向けずにオペレーティングシステムをインストールしてしまうことがよくあります。これにより、不要なサービスがインストールされ、デフォルト設定でオンになっていることで、問題が発生する場合があります。つまり、管理者が気附かないところで、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、クラッカーがシステムのパスを悪用できてしまう可能性があります。

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、多数のバグが発見され、修正されてきました。

しかし、完璧なソフトウェアというものはありません、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少なくありません。

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）などで、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が公表されています。このような情報発信は、コミュニティにセキュリティ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、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えると、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を行うには、警戒を怠らず、パッチ追跡を絶えず行い、適切なシステム保守を実行して、よりセキュアなコンピューティング環境を維持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。これには、管理者の経験の少なさだけでなく、管理者の過信やモチベーションの低さなども原因となります。

管理者が、サーバーやワークステーションにパッチを当てるのを忘れたり、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こともあります。その他にも、よく起こるケースとして、サービスのデフォルトパスワードや鍵を変更しないまま放置しておくことが挙げられます。たとえば、データベースにはデフォルトの管理パスワードが設定されているものがありますが、ここでは、システム管理者がインストール後すぐにデフォルトパスワードを変更することを、データベース開発者は想定しています。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、管理者の不注意によりサーバーが危険にさらされる場合もあります。

本質的に安全ではないサービス

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

安全ではないネットワークサービスの例として、暗号化されていないユーザー名とパスワードを認証時に要求するサービスが挙げられます。具体的例としては、TelnetやFTPの2つがあげられます。パケット盗聴ソフトウェアがリモートユーザーとこのようなサービスの間のトラフィックを監視している場合、ユーザーネ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、クラッカーが、ネットワーク上でクラッキングしたネットワークを操作して、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、攻撃者のマシンで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む。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例としては、他にもNFS、NISなどのネットワークファイルシステムおよび情報サービスが挙げられます。このサービスは、LAN利用を目的として開発されましたが、リモートユーザーのWANも対象として開発されました。(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す。NISも、ブレーンテキストのASCIIまたはDBM (ASCIIから派生)データベースに、パスワードやファイルーパミッションなど、ネットワーク上の全コンピュータへの周知が必要とな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ります。

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、管理者は、このようなサービスを使用しないといけない場合があるため、注意して設定することが重要となります。

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かなかありませんが、クレジットカード情報のような機密データが含まれるため、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選択され、一連の攻撃で「スレーブ」マシンとして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、バ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ないかもしれません。たとえば、サーバーが公開ネットワーク上でTelnetやFTPのサービスを提供している場合、攻撃者はネットワークを通じてブ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プロトコルのバージョン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プロトコルのバージョン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。この一般的な不正使用では、それがどのように実行され、管理者がその攻撃からネットワークをどのように適切に保護できるかを理解していることが重要になります。

<table>
<thead>
<tr>
<th>不正使用</th>
<th>説明</th>
<th>備考</th>
</tr>
</thead>
<tbody>
<tr>
<td>Red Hat Enterprise Linux 8</td>
<td>セキュリティーの強化</td>
<td></td>
</tr>
<tr>
<td>不正使用</td>
<td>説明</td>
<td>備考</td>
</tr>
<tr>
<td>----------</td>
<td>------</td>
<td>------</td>
</tr>
<tr>
<td>空またはデフォルトのパスワード</td>
<td>管理パスワードを空白のままにしたり、製品ベンダーが設定したデフォルトのパスワードをそのまま使用します。これは、ルーターやファイアウォールなどのハードウェアで最もよく見られますが、Linuxで実行するサービスにはデフォルトの管理者パスワードが設定され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）。</td>
<td>一般的に、ルーター、ファイアウォール、VPN、ネットワーク接続ストレージ（NAS）の機器など、ネットワークハードウェアに関連するものです。多数のレガシーオペレーティングシステム、特にサービスをバンドルしたオペレーティングシステム（UNIXやWindowsなど）でよく見られます。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。このような空白のパスワードは、このアカウントを発見した悪意のあるユーザーが利用できる絶好のエントリーポイントとなります。</td>
</tr>
<tr>
<td>デフォルトの共有鍵</td>
<td>セキュアなサービスでは、開発や評価テスト向けにデフォルトのセキュリティー鍵がパッケージ化されていることがあります。この鍵を変更せずにインターネットの実稼働環境に置いた場合は、同じデフォルトの鍵を持つすべてのユーザーがその共有鍵のリソースや、そこにあたるすべての機密情報にアクセスできるようになります。</td>
<td>無線アクセスポイントや、事前設定済みでセキュアなサーバー機器に最も多く見られます。</td>
</tr>
<tr>
<td>IPスプーフィング</td>
<td>リモートマシンがローカルネットワークのノードのように動作し、サーバーに脆弱性を見つけるとバックドアプログラムまたはトロイの木馬をインストールして、ネットワークリソース全体へのコントロールを得ようとします。</td>
<td>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している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。</td>
</tr>
</tbody>
</table>
盗聴

2つのノード間の接続を盗聴することにより、ネットワーク上のアクティブなノード間を行き交うデータを収集します。

この種類の攻撃には大抵、Telnet、FTP、HTTP転送などのプレーンテキストの転送プロトコルが使用されます。

リモートの攻撃者がこのような攻撃を仕掛けるには、LAN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に対する防護策としては、暗号化鍵交換、ワンタイムパスワード、または暗号化された認証によるサービス使用が挙げられます。通信中は強力な暗号化を実施することをお勧めします。

<table>
<thead>
<tr>
<th>不正使用</th>
<th>説明</th>
<th>備考</th>
</tr>
</thead>
<tbody>
<tr>
<td>盗聴</td>
<td>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。</td>
<td>この種類の攻撃には大抵、Telnet、FTP、HTTP転送などのプレーンテキストの転送プロトコルが使用されます。リモートの攻撃者がこのような攻撃を仕掛けるには、LAN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に対する防護策としては、暗号化鍵交換、ワンタイムパスワード、または暗号化された認証によるサービス使用が挙げられます。通信中は強力な暗号化を実施することをお勧めします。</td>
</tr>
<tr>
<td>不正使用</td>
<td>説明</td>
<td>備考</td>
</tr>
<tr>
<td>----------</td>
<td>-------</td>
<td>------</td>
</tr>
<tr>
<td>サービスの脆弱性</td>
<td>攻撃者はインターネットで実行しているサービスの欠陥や抜け穴を見つけています。攻撃者がこの脆弱性を利用する場合は、システム全体と格納されているデータを攻撃するだけでなく、ネットワーク上の他のシステムも攻撃する可能性があります。</td>
<td>CGIなどのHTTPベースのサービスは、リモートのコマンド実行やインタラクティブなシェルアクセスに対しても脆弱です。HTTPサービスが「nobody」などの権限のないユーザーとして実行している場合でも、設定ファイルやネットワークマップなどの情報が読み取られる可能性があります。または、攻撃者がサービス拒否攻撃を開始して、システムのリソースを浪費させたり、他のユーザーが利用できないようにする可能性もあります。開発時およびテスト時には気が付かない脆弱性がサービスに含まれることがあります。(アプリケーションのメモリーバッファー領域をあふれさせ、任意のコマンドを実行できるようなインタラクティブなコマンドプロンプトを攻撃者に提供するように、攻撃者が任意の値を使用してサービスをクラッシュさせるバッファーオーバーフローなどの)脆弱性は、完全な管理コントロールを攻撃者に与えるものとなる可能性があります。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。</td>
</tr>
<tr>
<td>不正使用</td>
<td>説明</td>
<td>備考</td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>アプリケーションの脆弱性</td>
<td>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。</td>
<td>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。</td>
</tr>
<tr>
<td>サービス拒否攻撃 (DoS: Denial of Service)</td>
<td>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る。これにより、正当なユーザーがリソースを使用できなくなります。</td>
<td>米国で最も多く報告されたDOSの問題は、2000年に発生しました。この時、通信量が非常に多い民間および政府のサイトが一部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。</td>
</tr>
<tr>
<td></td>
<td></td>
<td>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入侵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。</td>
</tr>
</tbody>
</table>
第2章 インストール時の RHEL の保護

セキュリティーへの対応は、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 権限を取得することを防ぐことができます。このような攻撃に対するセキュリ
ティー対策は、ワークステーションの情報の機密性とマシンの場所によって異なります。

たとえば、見本市で使用されていて機密情報を含んでいないマシンでは、このような攻撃を防ぐことが
重要ではないかもしれません。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、誰の監視下にもかかわらず置かれている場合は、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 2 つの理由を以下に示します。

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う可能性があります。

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

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

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

2.1.1.1. 非 BIOS ベースシステムのセキュリティー

その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 システムの BIOS とほぼ
同等の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、メーカーにお問い合わせください。

2.2. ディスッ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推奨します。各パーティションを作成する理由は以下のようになります。

/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す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満杯になり、オペレーティングシステムが不安定になる可能性があります。また、システムを、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ため、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、データが完全に失われます。したがって、パーティションを分けることが、データ損失に対する保護につながります。また、このパーティションを、頻繁にバックアップを作成する対象に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使用されてしまう可能性があります。このディレクトリーは / に置かれているため、こうした状態が発生すると、システムが不安定になり、クラッシュする可能性があります。そのため、このディレクトリーは個別のパーティションに移動することが推奨されます。

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あります。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のに使用されるパ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 8 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ている可能性がある場合があります。

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時に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、一方で LAN またはインフ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の慣行に従い、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
コンピューターの各ソフトウェアには脆弱性が潜在して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 から行う場合には、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。その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、キックスタート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 で次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い場合は、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、次のコマンドを実行します。
   
   ```
   $ systemctl list-units | grep service
   ```

[[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と、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。

<table>
<thead>
<tr>
<th>レベル</th>
<th>説明</th>
</tr>
</thead>
<tbody>
<tr>
<td>DEFAULT</td>
<td>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、TLSプロトコルの1.2と1.3、IKEv2プロトコル、およびSSH2プロトコルが使用できます。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。</td>
</tr>
<tr>
<td>LEGACY</td>
<td>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限しますが、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。</td>
</tr>
<tr>
<td>FUTURE</td>
<td>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。</td>
</tr>
<tr>
<td>FIPS</td>
<td>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FIPSモードに切り替えます。</td>
</tr>
</tbody>
</table>

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に対応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
```

暗号化ポリシーの変更を確実に適用するには、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した、強力な暗号デフォルト
以下の一覧には、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が含まれます。このアプリケーションはソースには存在しないか、またはビルド時にサポートを無効にしているため、アプリケーションは使用できません。

- 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)
- 224 ビットより小さいすべての ECC 曲線 (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
以下の暗号スイートおよびプロトコルは、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に限り利用可能にできます。

- パラメーターが 1024 ビットより小さい DH
- 鍵のサイズが 1024 ビットより小さい RSA
- 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)

暗号ポリシーの各レベルで有効な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スイートおよびプロトコルを示しています。

<table>
<thead>
<tr>
<th></th>
<th>LEGACY</th>
<th>DEFAULT</th>
<th>FIPS</th>
<th>FUTURE</th>
</tr>
</thead>
<tbody>
<tr>
<td>IKEv1</td>
<td>いいえ</td>
<td>いいえ</td>
<td>いいえ</td>
<td>いいえ</td>
</tr>
<tr>
<td>3DES</td>
<td>はい</td>
<td>いいえ</td>
<td>いいえ</td>
<td>いいえ</td>
</tr>
<tr>
<td>RC4</td>
<td>はい</td>
<td>いいえ</td>
<td>いいえ</td>
<td>いいえ</td>
</tr>
<tr>
<td></td>
<td>LEGACY</td>
<td>DEFAULT</td>
<td>FIPS</td>
<td>FUTURE</td>
</tr>
<tr>
<td>------------------</td>
<td>--------</td>
<td>---------</td>
<td>------</td>
<td>--------</td>
</tr>
<tr>
<td>DH</td>
<td>最低1024ビット</td>
<td>最低2048ビット</td>
<td>最低2048ビット</td>
<td>最低3072ビット</td>
</tr>
<tr>
<td>RSA</td>
<td>最低1024ビット</td>
<td>最低2048ビット</td>
<td>最低2048ビット</td>
<td>最低3072ビット</td>
</tr>
<tr>
<td>DSA</td>
<td>いいえ</td>
<td>いいえ</td>
<td>いいえ</td>
<td>いいえ</td>
</tr>
<tr>
<td>TLS v1.0</td>
<td>いいえ</td>
<td>いいえ</td>
<td>いいえ</td>
<td>いいえ</td>
</tr>
<tr>
<td>TLS v1.1</td>
<td>いいえ</td>
<td>いいえ</td>
<td>いいえ</td>
<td>いいえ</td>
</tr>
<tr>
<td>デジタル署名におけ るSHA-1</td>
<td>いいえ</td>
<td>いいえ</td>
<td>いいえ</td>
<td>いいえ</td>
</tr>
<tr>
<td>CBCモード暗号</td>
<td>いいえ</td>
<td>いいえ</td>
<td>いいえ</td>
<td>いいえ</td>
</tr>
<tr>
<td>256ビットより小 さい鍵を持つ対称 暗号</td>
<td>いいえ</td>
<td>いいえ</td>
<td>いいえ</td>
<td>いいえ</td>
</tr>
<tr>
<td>証明書における SHA-1および SHA-224の署名</td>
<td>いいえ</td>
<td>いいえ</td>
<td>いいえ</td>
<td>いいえ</td>
</tr>
</tbody>
</table>

関連情報

- 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。

関連情報

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

関連情報

- 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. FIPS モードへのシステムの切り替え

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) 公開文書 140-2 の要件に準拠した暗号化モジュールのセルフチェックを有効にするポリシーレベルが含まれます。FIPS 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 の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 FIPS 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. システムが再起動したら、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.
```

関連情報

- 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. システム全体の暗号化ポリシーに従わないようにアプリケーションを除外

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。

3.4.1. システム全体の暗号化ポリシーを除外する例

wget

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。

```bash
# wget --secure-protocol=TLSv1_1 --ciphers="SECURE128"
```

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

curl ソ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。

```bash
# curl --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。

Firefox

Web ブラウザーの Firefox でシステム全体の暗号化ポリシーをオプトアウトできない場合でも、Firefox の設定エディターで、対応している暗号と TLS バージョンをさらに詳細に制限できます。アド
レスバーに `about:config` と入力し、必要に応じて `security.tls.version.min` の値を変更します。たと
えば、`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、`security.tls.version.min 2` が TLS 1.1 になります。

OpenSSH

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、KexAlgorithms 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

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。

関連情報

- 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 PKCS #11 で暗号化ハードウェアを使用するようにアプリケーションを設定

スマートカードや、エンドユーザー認証用の暗号化トークン、サーバーアプリケーション用のハードウェアセキュリティーモジュール (HSM) など、専用の暗号化デバイスで秘密情報の一部を分離することで、セキュリティー層が追加されます。Red Hat Enterprise Linux 8 では、PKCS #11 API を使用した暗号化ハードウェアへの対応がアプリケーション間で統一され、暗号ハードウェアでの秘密の分離が複雑なタスクではなくなりました。

4.1. PKCS #11 による暗号化ハードウェアへの対応

PKCS #11 (Public-Key Cryptography Standard) は、暗号化情報を保持する暗号化デバイスに、アプリケーションプログラミングインタフェース (API) を定義し、暗号化機能を実行します。デバイスはトークンと呼ばれ、ハードウェアまたはソフトウェアに実装できます。

PKCS #11 トークンに保存されるストレージオブジェクトタイプには、証明書、データオブジェクト、公開鍵、秘密鍵、またはシークレットキーが含まれます。このオブジェクトは、PKCS #11 の URI スキームにより一意に識別できます。

PKCS #11 の URI では、属性に従って、PKCS #11 モジュールで特定のオブジェクトを識別する標準的な方法が提供されます。これにより、URI の形式で、すべてのライブラリーやアプリケーションを同じ設定文字列で設定できます。

Red Hat Enterprise Linux 8 では、デフォルトでスマートカード用に OpenSC PKCS #11 ドライバーが提供されています。ただし、Red Hat Enterprise Linux では、ハードウェアトークンと HSM に独自のPKCS #11 モジュールがあります。この PKCS #11 モジュールは p11-kit ツールで登録できます。これは、システムの登録済みスマートカードドライバーにおけるラッパーとして機能します。

システムで独自の PKCS #11 モジュールを有効にするには、新しいテキストファイルを /etc/pkcs11/modules/ ディレクトリーに追加します。

システムに独自の PKCS #11 モジュールを追加すると、/etc/pkcs11/modules/ ディレクトリーに新しいテキストファイルのみを作成する必要があります。たとえば、p11-kit の OpenSC 設定ファイルは、以下のようになります。

```
$ cat /usr/share/p11-kit/modules(opensc.module
module: opensc-pkcs11.so
```

関連情報

- PKCS #11 の URI スキーム
- Controlling access to smart cards

4.2. スマートカードに保存された SSH 鍵の使用

Red Hat Enterprise Linux 8 では、OpenSSH クライアントでスマートカードに保存されている RSA 鍵および ECDSA 鍵を使用できるようになりました。

前提条件

- クライアントで、opensc パッケージをインストールして、pcscd サービスを実行している。
1. PKCS#11 の URI を含む OpenSC PKCS#11 モジュールが提供する鍵の一覧を表示し、この出力を keys.pub ファイルに保存します。

```bash
$ ssh-keygen -D pkcs11: > keys.pub
$ ssh-keygen -D pkcs11:
pkcs11:id=%02;object=SIGN%20pubkey;token=SSH%20key;manufacturer=piv_II?
module-path=/usr/lib64/pkcs11/opensc-pkcs11.so
ecdsa-sha2-nistp256 AAA...J0hkYnnsM=
pkcs11:id=%01;object=PIV%20AUTH%20pubkey;token=SSH%20key;manufacturer=piv_II?
module-path=/usr/lib64/pkcs11/opensc-pkcs11.so
```

2. リモートサーバー (example.com) でスマートカードを使用した認証を有効にするには、公開鍵をリモートサーバーに転送します。前の手順で作成された keys.pub で ssh-copy-id コマンドを使用します。

```bash
ssh-copy-id -f -i keys.pub username@example.com
```

3. 手順 1 の ssh-keygen -D コマンドの出力にある ECDSA 鍵を使用して example.com に接続するには、鍵を一意に参照する URI のサブセットのみを使用できます。以下に例を示します。

```bash
$ ssh -i "pkcs11:id=%01?module-path=/usr/lib64/pkcs11/opensc-pkcs11.so" example.com
```

4. ~/.ssh/config ファイルで同じ URI 文字列を使用して、設定を永続化できます。

```bash
$ cat ~/.ssh/config
IdentityFile "pkcs11:id=%01?module-path=/usr/lib64/pkcs11/opensc-pkcs11.so"
$ ssh example.com
```

OpenSSH は p11-kit-proxy ラッパーを使用し、OpenSC PKCS #11 モジュールが PKCS#11 キットに登録されているため、以前のコマンドを簡素化できます。

```bash
$ ssh -i "pkcs11:id=%01" example.com
```

PKCS #11 の URI の id= の部分を飛ばすと、OpenSSH が、プロキシーモジュールで利用可能な鍵をすべて読み込みます。これにより、必要な入力の量を減らすことができます。

```bash
$ ssh -i pkcs11: example.com
```

関連情報
- Fedora 28: Better smart card support in OpenSSH
- man ページの p11-kit(8)
4.3. APACHE および NGINX で秘密鍵を保護する HSM の使用

HTTP サーバーの Apache および Nginx は、ハードウェアセキュリティーモジュール (HSM) に保存されている秘密鍵と連携できます。これにより、鍵の漏えいや中間者攻撃を防ぐことができます。通常、これを行うには、ビジーなサーバーに高パフォーマンスの HSM が必要になります。

Apache HTTP サーバー

HTTPS プロトコルの形式でセキュアな通信を行うために、Apache HTTP サーバー (httpd) は OpenSSL ライブラリを使用します。OpenSSL は、PKCS #11 にネイティブに対応しません。HSM を使用するには、エンジンインターフェースを介して PKCS #11 モジュールへのアクセスを提供する openssl-pkcs11 パッケージをインストールする必要があります。通常のファイル名ではなく PKCS #11 の URI を使用すると、/etc/httpd/conf.d/ssl.conf 設定ファイルでサーバーの鍵と証明書を指定できます。以下に例を示します。

```
SSLCertificateFile  "pkcs11:id=%01;token=softhsm;type=cert"
SSLCertificateKeyFile "pkcs11:id=%01;token=softhsm;type=private?pin-value=111111"
```

httpd-manual パッケージをインストールして、TLS 設定を含む Apache HTTP サーバーの完全ドキュメントを取得します。/etc/httpd/conf.d/ssl.conf 設定ファイルで利用可能なディレクティブの詳細は、/usr/share/httpd/manual/mod/mod_ssl.html を参照してください。

Nginx HTTP およびプロキシーサーバー

Nginx は暗号化操作に OpenSSL を使用するため、PKCS #11 への対応は openssl-pkcs11 エンジンを介して行う必要があります。Nginx は現在、HSM からの秘密鍵の読み込みのみに対応します。また、証明書は通常のファイルとして個別に提供する必要があります。/etc/nginx/nginx.conf 設定ファイルの server セクションで ssl_certificate オプションおよび ssl_certificate_key オプションを変更します。

```
ssl_certificate /path/to/cert.pem
ssl_certificate_key "engine:pkcs11:pkcs11:token=softhsm;id=%01;type=private?pin-value=111111";
```

Nginx 設定ファイルの PKCS #11 URI にプレフィックス engine:pkcs11: が必要なことに注意してください。これは、他の pkcs11 プレフィックスがエンジン名を参照するためです。

4.4. スマートカードから証明書を使用して認証するアプリケーションの設定

- wget ネットワークダウンローダーでは、ローカルに保存された秘密鍵へのパスの代わりに PKCS #11 の URI を指定できるため、安全に保存された秘密鍵と証明書を必要とするタスク用のスクリプトの作成が容易になります。以下に例を示します。

```
$ wget --private-key 'pkcs11:token=softhsm;id=%01;type=private?pin-value=111111' --certificate 'pkcs11:token=softhsm;id=%01;type=cert' https://example.com/
```

詳細は、man ページの wget(1) を参照してくださいます。

- curl ツールで使用する PKCS #11 の URI は、以下のように指定します。
$ curl --key 'pkcs11:token=softhsm;id=%01;type=private?pin-value=111111' --cert 'pkcs11:token=softhsm;id=%01;type=cert' https://example.com/

詳細は、man ページの curl(1) を参照してください。

- Web ブラ우ザーの Firefox は、p11-kit-proxy モジュールを自動的に読み込みます。つまり、システムで対応しているすべてのスマートカードが自動的に検出されます。TLS クライアント認証を使用した場合、その他に必要な設定はありません。また、サーバーがスマートカードを要求する際に、スマートカードの鍵が自動的に使用されます。

カスタムアプリケーションで PKCS #11 の URI の使用

アプリケーションが GnuTLS ライブラリーまたは NSS ライブラリーを使用する場合、PKCS #11 の URI は PKCS #11 の組み込みサポートで保証されます。また、OpenSSL ライブラリーに依存するアプリケーションは、openssl-pkcs11 エンジンが生成する暗号化ハードウェアモジュールにアクセスできます。

アプリケーションでスマートカードの秘密鍵を使用する必要があり、NSS、GnuTLS、または OpenSSL は使用しない場合は、p11-kit を使用して PKCS #11 モジュールの登録を実装します。

詳細は、man ページの p11-kit (8) を参照してください。

4.5. 関連情報

- man ページの pkcs11.conf(5)
第5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一覧を選択したり、証明書一覧を作成したりできます。

5.1. 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。

5.2. 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。

   ```bash
   # 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設定を更新するには、update-ca-trustコマンドを実行します。

   ```bash
   # update-ca-trust 
   ```
Firefox ブラウザでは、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。

### 5.3. 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。

```bash
$ 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。

```bash
$ 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
...
[trimmed for clarity]
```

- 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。

```bash
# 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。
関連情報

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
cia-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

5.4. 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6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。ポリシーは、システムの目的や、組織におけるシステム重要性により異なります。カスタマイズしたソフトウェア設定や導入の特徴によっても、カスタマイズしたポリシーのチェックリストが必要になっています。

6.1. 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) 規格に基づいており、コンプライアンスポリシーの自動化に合わせるように設計されています。

- **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ツントを検証し、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、セキュリティ強化に関する実用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求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照査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。

関連情報

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

6.2. RED HAT SECURITY ADVISORIES OVAL ディテール

Red Hat Enterprise Linux のセキュリティ監査機能は、標準規格「セキュリティ設定共通化手順 (Security Content Automation Protocol (SCAP))」を基にしています。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 の仕様は、セキュリティーコンテンツの形式により、既知で標準化されたエコシステムが作られますが、一方で、スキャナーやポリシーエディターの導入は義務化されていません。このような状態では、企業がいくつか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。その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、宣言型でリソースが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 標準規格は、いくつかのドキュメント形式を定義します。この形式にはそれぞれ異なる種類の情報が記載され、異なる目的に使用されます。

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問題をすべて追跡して調査します。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します。Red Hat は OVAL パッチ定義を作成してサポートし、マシンが判読可能なセキュリティーアドバイザリーを提供します。

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 パッケージが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。

関連情報
- 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

6.3. 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、セキュリティーロンプライアンスコンテンツの確認、ならびにスキャンおよび評価を基にしたレポートとガイドの生成が可能です。このユーティリティーは、OpenSCAP ライブラリーのフロントエンドとしてサービスを提供し、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) にグルーブ化します。

前提条件

- AppStream リポジトリ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/com.redhat.rhsa-RHEL8.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 com.redhat.rhsa-RHEL8.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 &
```

関連情報

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

6.4. 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/com.redhat.rhsa-RHEL8.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 com.redhat.rhsa-RHEL8.xml
```

関連情報

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

6.5. セキュリティーコンプライアンスのプロファイルの表示

RHEL 8 は、セキュリティーポリシーを扱う複数のプロファイルを提供します。スキャンまたは修復にそれらを使用することを決定する前に、その一覧を表示して oscap info サブコマンドで詳しい説明を確認できます。

前提条件

- `openscap-scanner` パッケージおよび `scap-security-guide` パッケージがインストールされている。

手順

1. SCAP Security Guide プロジェクトが提供するセキュリティーコンプライアンスプロファイルで利用可能なファイルをすべて表示します。

```
$ ls /usr/share/xml/scap/ssg/content/
  ssg-firefox-cpe-dictionary.xml  ssg-rhel6-ocil.xml
  ssg-firefox-cpe-oval.xml       ssg-rhel6-oval.xml
  ...
  ssg-rhel6-ds-1.2.xml           ssg-rhel8-oval.xml
  ssg-rhel8-ds.xml              ssg-rhel8-xccdf.xml
  ...
[trimmed for clarity]
```

2. oscap info サブコマンドを使用して、選択したデータストリームに関する詳細情報を表示します。データストリームを含む XML ファイルは、名前に -ds 文字列で示されます。Profiles セクションでは、利用可能なプロファイルと、その ID の一覧を確認できます。

```
$ oscap info /usr/share/xml/scap/ssg/content/ssg-rhel8-ds.xml
...
Profiles:
  Title: PCI-DSS v3 Control Baseline for Red Hat Enterprise Linux 8
  Id: xccdf_org.ssgproject.content_profile_pci-dss
  Title: OSPP - Protection Profile for General Purpose Operating Systems
```
3. データストリームファイルからプロファイルを選択し、選択したプロファイルに関する追加情報表示します。そのためには、oscap info に --profile オプションを指定した後に、直前のコマンドの出力で表示された ID の最後のセクションを指定します。たとえば、PCI-DSS プロファイルの ID は xccdf_org.ssgproject.content_profile_pci-dss で、--profile オプションの値は pci-dss です。

$ oscap info --profile pci-dss /usr/share/xml/scap/ssg/content/ssg-rhel8-ds.xml
...
Title: PCI-DSS v3.2.1 Control Baseline for Red Hat Enterprise Linux 8
Id: xccdf_org.ssgproject.content_profile_pci-dss
Description: Ensures PCI-DSS v3.2.1 security configuration settings are applied.
...
[trimmed for clarity]

関連情報
- man ページの scap-security-guide (8)

6.6. 特定のベースラインによるセキュリティーコンプライアンスの評価

SCAP セキュリティーガイド スイートは、XCCDF、OVAL、およびデータストリームのドキュメント形式で、複数のプラットフォームのプロファイルを提供します。このプロファイルは、OSPP (Operating System Protection Profile)、PCI-DSS (Payment Card Industry Data Security Standard) などのセキュリティーポリシーに基づく一連のルールです。これにより、セキュリティー規格に関しては、自動化された方法でシステムを監査できます。

前提条件
- openscap-scanner パッケージがインストールされている。

手順
1. scap-security-guide パッケージをインストールします。

   # yum install scap-security-guide

2. oscap info サブコマンドを使用して、選択したデータストリームに関する詳細情報を表示します。Profiles セクションでは、利用可能なプロファイルと、その ID の一覧を確認できます。

   $ oscap info /usr/share/xml/scap/ssg/content/ssg-rhel8-ds.xml
   ...
   Profiles:
   Title: PCI-DSS v3 Control Baseline for Red Hat Enterprise Linux 8
   Id: xccdf_org.ssgproject.content_profile_pci-dss
   Title: OSPP - Protection Profile for General Purpose Operating Systems
   Id: xccdf_org.ssgproject.content_profile_ospp
   ...
   [trimmed for clarity]
データストリームファイルからプロファイルを選択し、前のコマンドの出力で特定されたIDの最後の部分を oscap info の --profile オプションに提供し、選択したプロファイルの詳細を表示します。たとえば、OSPP プロファイルには、ID xccdf_org.ssgproject.content_profile_ospp があり、--profile オプションの値は ospp です。

```bash
$ oscap info --profile ospp /usr/share/xml/scap/ssg/content/ssg-rhel8-ds.xml
```

PCI-DSS v3 Control Baseline プロファイルは、xccdf_org.ssgproject.content_profile_pci-dss によって識別されます。

```bash
$ oscap info --profile pci-dss /usr/share/xml/scap/ssg/content/ssg-rhel8-ds.xml
... Description: Ensures PCI-DSS v3 related security configuration settings are applied. [trimmed for clarity]
```


3. 選択したプロファイルでそのシステムがどのように複雑であるかを評価し、スキャン内容を保存すると、以下のように HTML ファイル (report.html) に結果が表示されます。

```bash
$ oscap xccdf eval --report report.html --profile ospp /usr/share/xml/scap/ssg/content/ssg-rhel8-ds.xml
```

システムにインストールされている openscap-utils パッケージと、リモートシステムにインストールされている openscap-scanner パッケージを使用して、ホスト名 machine1、ポート 22で実行している SSH、およびユーザー名 joesec でリモートシステムで脆弱性をスキャンし、結果を remote-report.html ファイルに保存します。

```bash
$ oscap-ssh joesec@machine1 22 xccdf eval --report remote_report.html --profile ospp /usr/share/xml/scap/ssg/content/ssg-rhel8-ds.xml
```

### 関連情報

- man ページの scap-security-guide (8)
- file:///usr/share/doc/scap-security-guide/ ディレクトリーにインストールされている SCAP セキュリティーガイドに関するドキュメント
- scap-security-guide-doc パッケージでインストールされている「Guide to the Secure Configuration of Red Hat Enterprise Linux 8」

### 6.7. OSPP に合わせてシステムの修復

この手順を使用して、OSPP (Protection Profile for General Purpose Operating Systems) に合わせて RHEL 8 システムを修復します。
重要

Red Hatは、セキュリティを強化した修正で加えられた変更を元に戻す自動手段は提供していません。修正は、デフォルト設定のRHELシステムで対応しています。インストール後にシステムが変更した場合は、修正を実行しても、必要なセキュリティープロファイルに準拠しない場合があります。

前提条件

- RHEL 8システムに、scap-security-guideパッケージがインストールされている。

手順

1. oscapコマンドに--remediateオプションを指定して使用します。
   
   ```bash
   # oscap xccdf eval --profile ospp --remediate /usr/share/xml/scap/ssg/content/ssg-rhel8-ds.xml
   ```

2. システムを再起動します。

検証手順

1. そのシステムがOSPPプロファイルにどのように準拠しているかを評価し、結果をospp_report.htmlファイルに保存します。
   
   ```bash
   $ oscap xccdf eval --report ospp_report.html --profile ospp
   /usr/share/xml/scap/ssg/content/ssg-rhel8-ds.xml
   ```

関連情報

- manページのscap-security-guide(8)およびoscap(8)

6.8. SCAP WORKBENCHを使用したカスタムプロファイルでシステムのスキャン

SCAP Workbench(scap-workbench)はグラフィカルユーティリティで、1台のローカルシステムまたはリモートシステムで構成スキャンと脆弱性スキャンを実行し、システムの修復を実行して、スキャン評価に基づくレポートを生成します。oscapコマンドラインユーティリティとの比較は、SCAP Workbenchには限定的な機能しかないことに注意してください。また、SCAP Workbenchは、XCCDFおよびデータストリームファイルの形式でのみセキュリティーコンテンツを処理できます。

前提条件

- SCAP Workbenchが、yum install scap-workbenchコマンドで、システムにインストールされている。

6.8.1. SCAP Workbenchを使用したシステムのスキャンおよび修復

選択したセキュリティーポリシーに対してシステムを評価するには、以下の手順に従います。

手順

1. GNOMEClassicデスクトップ環境をSCAP Workbenchで有効化するには、オペレーティングシステムのカスタム設定を実行する必要があります。
1. **GNOME Classic** デスクトップ環境から **SCAP Workbench** を実行するには、**Super** キーを押してアクティビティーの概要を開き、scap-workbenchと入力して**Enter**を押します。または、次のコマンドを実行します。

```
$ scap-workbench &
```

2. 以下のオプションのいずれかを使用してセキュリティーポリシーを選択します。

- 開始ウィンドウの **Load Content** ボタン
- **Open content from SCAP Security Guide**
- **File** メニューの **Open Other Content** で、XCCDF、SCAP RPM、またはデータストリームファイルの各ファイルを検索します。

3. **Remediate** チェックボックスを選択して、システム設定の自動修正を行うことができます。このオプションを有効にすると、**SCAP Workbench** は、ポリシーにより適用されるセキュリティールールに従ってシステム設定の変更を試みます。このプロセスは、システムスキャン時に失敗した関連チェックを修正する必要があります。

警告
修正オプションが有効な状態でのシステム評価は、慎重に行わないとシステムが機能不全に陥る場合があります。

4. **Scan** ボタンをクリックし、選択したプロファイルでシステムをスキャンします。
5. スキャン結果を XCCDF ファイル、ARF ファイル、または HTML ファイルの形式で保存するには、Save Results コンボボックスをクリックします。HTML Report オプションを選択して、スキャンレポートを、人間が判読できる形式で生成します。XCCDF 形式および ARF (データストリーム) 形式は、追加の自動処理に適しています。3つのオプションはすべて繰り返し選択できます。

6. 結果ベースの修復をファイルにエクスポートするには、ポップアップメニューの Generate remediation role を使用します。

6.8.2. SCAP Workbench を使用したセキュリティープロファイルのカスタマイズ

以下の手順では、SCAP Workbench を使用してプロファイルをカスタマイズする方法を示します。oscap コマンドラインユーティリティで使用するようにカスタマイズしたプロファイルを保存することもできます。

手順

1. SCAP Workbench を実行し、Open content from SCAP Security Guide または File メニューの Open Other Content を使用してカスタマイズするプロファイルを選択します。

2. 選択したセキュリティープロファイルをさらに調整して、組織のニーズに応じてプロファイルを厳格または緩やかにするには、Customize ボタンをクリックします。これにより、各 XCCDF ファイルを変更せずに、現在選択している XCCDF プロファイルを編集できる新しいカスタマイズウィンドウが開きます。新しいプロファイル ID を選択します。
3. 論理グループに分けられたルールを持つツリー構造を使用するか、Search フィールドを使用して変更するルールを検索します。

4. ツリー構造のチェックボックスを使用した include ルールまたは exclude ルール、または必要に応じてルールの値を変更します。

5. OK ボタンをクリックして変更を確認します。

6. 変更内容を永続的に保存するには、以下のいずれかのオプションを使用します。
   - File メニューの Save Customization Only を使用して、カスタマイズファイルを別途保存します。
   - File メニュー Save All を選択して、すべてのセキュリティーコンテンツを一度に保存します。
Into a directory オプションを選択すると、SCAP Workbench は、XCCDF ファイルまたはデータストリームファイル、ならびにカスタマイズファイルの両方を、指定した場所に保存します。これは、バックアップソリューションとして役に立ちます。

As RPM オプションを選択すると、SCAP Workbench に、XCCDF またはデータストリームファイル、ならびにカスタマイズファイルを含む RPM パッケージの作成を指示できます。これは、リモートでスキャンできないシステムにセキュリティーコンテンツを配布したり、詳細な処理のためにコンテンツを配信するのに便利です。

注記

SCAP Workbench は、カスタマイズしたプロファイル向けの結果ベースの修正に対応していないため、oscap コマンドラインユーティリティーでエクスポートした修正を使用します。

6.8.3. 関連情報

- man ページの scap-workbench (8)
- SCAP Workbench User Manual

6.9. インストール後にセキュリティープロファイルに準拠するシステムのデプロイメント

管理者は、OpenSCAP スイートを使用して、インストールプロセスの直後に、OSPP や PCI-DSS などのセキュリティープロファイルに準拠する RHEL システムをデプロイできます。このデプロイメント方法を使用した場合、管理者は、後で修正スクリプトを使用して適用することはできないパスワード強度のルールなどの特定ルールを適用できます。

6.9.1. グラフィカルインストールを使用した OSPP 準拠の RHEL システムのデプロイメント

OSPP (Protection Profile for General Purpose Operating System) に合わせて調整された RHEL システムをデプロイする場合は、この手順を使用します。

前提条件

- グラフィカルインストールプログラムでシステムを起動している。
- インストール概要 画面を開いている。

手順

1. インストール概要 画面で、ソフトウェアの選択 をクリックします。ソフトウェアの選択 画面が開きます。
2. ベース環境 ペインで、サーバー 環境を選択します。ベース環境は、1つだけ選択できます。
警告

GUI を使用するサーバー は、デフォルトのベース環境です。GUI を使用するサーバー オプションでインストールした GNOME パッケージには nfs-utils パッケージが必要で、このパッケージは OSPP に準拠していません。デフォルトのベース環境を Server に変更しないで OSPP を選択すると、インストールプロセスが停止します。

3. 完了 をクリックして設定を適用し、インストール概要 画面に戻ります。
4. セキュリティーポリシー をクリックします。セキュリティーポリシー 画面が開きます。
5. システムでセキュリティーポリシーを有効にするには、セキュリティーポリシーの適用 を ON に切り替えます。
6. プロファイルペインで Protection Profile for General Purpose Operating Systems プロファイルを選択します。
7. プロファイルの選択 をクリックして選択を確定します。
8. 画面下部に表示される Protection Profile for General Purpose Operating Systems の変更を確定します。残りの手動変更を完了します。
9. OSPP には、準拠する必要がある厳密なパーティション分割要件があるため、/boot, /home, /var, /var/log, /var/tmp, および /var/log/audit にそれぞれパーティションを作成します。
10. グラフィカルインストールプロセスを完了します。

注記

グラフィカルインストールプログラムは、インストールに成功すると、対応するキックスタートファイルを自動的に作成します。/root/anaconda-ks.cfg ファイルを使用して、OSPP 準拠のシステムを自動的にインストールできます。

検証手順

1. 強化プロセスのレポートは、/root/openscap_data/eval_remediate_report.html ファイルにあります。oscap は chroot 環境でレポートを作成するため、誤検出も含まれます。たとえばサービス関連のルールはすべてエラーとして示されます。
2. システムの現在のステータスを正確に確認するには、インストール完了後に再起動してスキャンします。

```bash
# oscap xccdf eval --profile ospp --report eval_postinstall_report.html
/usr/share/xml/scap/ssg/content/ssg-rhel8-ds.xml
```

関連情報

- パーティション設定の詳細は、「グラフィカルユーザーアンターフェースを使用した RHEL のインストール」を参照してください。
6.9.2. キックスタートを使用したOSPP準拠のRHELシステムのデプロイメント

この手順を使用して、OSPP (Protection Profile for General Purpose Operating System) に合わせて調整されたRHELシステムをデプロイします。

前提条件

- RHEL 8システムに、scap-security-guideパッケージがインストールされている。

手順

1. キックスタートファイル /usr/share/scap-security-guide/kickstarts/ssg-rhel8-ospp-ks.cfgを、選択したエディターで開きます。
2. 設定要件を満たすように、パーティション設定スキームを更新します。OSPPコンプライアンスでは、/boot、/home、/var、/var/log、/var/tmp、および/var/log/auditにそれぞれ設定したパーティションを維持して、パーティションのサイズのみを変更できます。

警告

OSCAP Anaconda Addon プラグインはテキストのみのインストールには対応していないため、キックスタートファイルのtextオプションは使用しないでください。詳細はRHBZ#1674001を参照してください。

3. キックスタートインストールを開始する方法は、「キックスタートインストールの開始」を参照してください。

重要

OSPP要件では、ハッシュ形式のパスワードは確認できません。

検証手順

1. 強化プロセスのレポートは、/root/openscap_data/eval_remediate_report.htmlファイルにあります。oscapはchroot環境でレポートを作成するため、誤検出も含まれます。たとえばサービス関連のルールはすべてエラーとして示されます。
2. システムの現在のステータスを正確に確認するには、インストール完了後に再起動してスキャンします。

# oscap xccdf eval --profile ospp --report eval_postinstall_report.html
/usr/share/xml/scap/ssg/content/ssg-rhel8-ds.xml

関連情報

- 詳細は、OSCAP Anaconda Addonプロジェクトページを参照してください。

6.10. 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 に関連するその他のコンポーネントおよびプロジェクトの詳細情報が提供されています。

SCAP Workbench プロジェクトページ - SCAP Workbench プロジェクトのホームページでは、scap-workbench アプリケーションの詳細情報が提供されています。

SCAP Security Guide (SSG) プロジェクトページ - SSG プロジェクトのホームページでは、Red Hat Enterprise Linux 向けの最新セキュリティーコンテンツが提供されています。

National Institute of Standards and Technology (NIST) SCAP ページ - このページでは、SCAP の出版物、仕様、SCAP 検出プログラムなどの SCAP 関連の資料が多数提供されます。

National Vulnerability Database (NVD) - このページは、SCAP コンテンツおよびその他の SCAP 規格ベースの脆弱性管理データに関する最大のリポジトリです。

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を維持する方法などが説明されています。
第7章 AIDE で整合性の確認

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。

7.1. 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、次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、次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のデータベース、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。

7.2. 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. 手動でチェックを開始するには、以下を行います。

   # aide --check
2. **AIDE** は、最低でも、スキャンを毎週実行するように設定する必要があります。**AIDE** は毎日実行する必要があります。たとえば、**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。

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

### 7.3. 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。

    ```bash
    # 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、ファイル名から末尾の .new を削除します。

### 7.4. 関連情報

**AIDE** の詳細は、man ページの aide (1) を参照してください。
第8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。デバイスで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形式は、Red Hat Enterprise Linuxにおけるブロックデバイスの暗号化のデフォルト実装です。

8.1. LUKSディスクの暗号化

LUKS (Linux Unified Key Setup-on-disk-format) は、ブロックディ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、複数のユーザー鍵が、パーティションのバルク暗号化に使用されるマスターキーを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を保護するのに適しています。
- 暗号化されたブロックデバイスの基本的な内容は任意であり、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、既存のディバイスマッパーのカーネルサブシステムを使用します。
- LUKSは、パラフレーズの強化を提供し、辞書攻撃から保護します。
- LUKSディバイスには複数のキーが含まれ、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。

8.1.1. RHEL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期せられていません。ハードドライブを暗号化するオプションを選択すると、コンピューターや起動する際にはパスフレーズを入力する必要があります。このパスフレーズは、パーティションの複号に用いられるバルク暗号化キューのロックに使用されます。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テーブルの変更を選択できます。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におけるデフォルトの形式はLUKS2です。従来のLUKS (LUKSI) は完全にサポートされ、後方互換性のある形式として提供されます。LUKS2形式はLUKSIに比べて、特定の状況でLUKSIから変換できます。具体的には、以下のシナリオでは変換ができます。
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、今後も、バイナリー構造を変更することなく、さまざまな要素を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 にのみ対応する以前のシステムとの互換性を必要とす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 は、バージョン 7.6 以降の LUKS2 形式に対応しています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

関連情報

- 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

8.2. 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

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。

```bash
# 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。

```bash
# 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。

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

関連情報

- 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。

8.3. 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

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、別の LUKS ヘッダーを使用してファイル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
```

関連情報

- 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9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) などのさまざまなロックの解除方法、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ロックの解除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 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。

9.1. 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技術により実装されます。

図9.1 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 (luksmeta パッケージは、LUKS2 ボリュームには使用されません)

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。エスクローベースのソリューション (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) 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コンポーネントで、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 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し、NBDE を実現します。

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します。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、使いやすい場面に、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 のデフォルト形式であるため、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 のプロビジョニング状態は、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、規格を必要とせずに 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 と呼ばれています。

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。

9.2. 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) 形式で暗号文を提供します。

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

関連情報

- 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]
```

54
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
Decrpts 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。

9.3. SELINUX を ENFORCING モードで有効にした TANG サーバーのデプロイメント

RHEL 8 では、SELinux タイプ tangd_port_t が追加され、Tang サーバーを、SELinux Enforcing モードで制限のあるサービスとしてデプロイできます。

前提条件
- policycoreutils-python-utils パッケージおよび依存関係がインストールされている。

手順
1. 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
2. 7500/tcp などの不要なポートを選択し、tangd サービスがそのポートにバインドできるようにします。

```
# semanage port -a -t tangd_port_t -p tcp 7500
```

ポートを1つのサービスのみで一度に使用できるため、すでに使用しているポートを使用しようとすると、ValueError: Port already defined エラーが発生します。

3. ファイアウォールのポートを開きます。

```
# firewall-cmd --add-port=7500/tcp
# firewall-cmd --runtime-to-permanent
```

4. tangd サービスを有効にします。

```
# systemctl enable tangd.socket
```

5. オーバーライドファイルを作成します。

```
# systemctl edit tangd.socket
```

6. 以下のエディター画面で、/etc/systemd/system/tangd.socket.d/ ディレクトリーにある空の override.conf ファイルを開き、次の行を追加して、Tang サーバーのデフォルトのポートを、80 から、以前取得した番号に変更します。

```
[Socket]
ListenStream=
ListenStream=7500
```

ファイルを保存して、エディターを終了します。

7. 変更した設定を再読み込みします。

```
# systemctl daemon-reload
```

8. 設定が機能していることを確認します。

```
# systemctl show tangd.socket -p Listen
Listen=[::]:7500 (Stream)
```

9. tangd サービスを開始します。

```
# systemctl start 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 ユーティリティを使用します。

関連情報
- man ページの semanage (8)
- man ページの firewall-cmd (1)
9.4. 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。

手順

1. 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。

   ```bash
   # 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
   ```

2. 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。

   ```bash
   # 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。

3. 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。

9.5. 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。  
- Tangサーバーが利用できる。
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:port"}' < input-plain.txt > secret.jwe

The advertisement contains the following signing keys:

_OsIk0T-E2l6qfdDwVmidoZjA

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

この例のURL(http://tang.srv:port)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

関連情報

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

9.6. 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 設定オブジェクト形式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にも対応します。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

関連情報
9.7. 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:
OtE0T-E2l6qfdDwVmidoZjA

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 ヘッダークー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次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、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。以下に例を示します。

```
# 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 `drac.cmdline(7)` を参照してください。

**関連情報**

詳細は、以下の `man` ページを参照してください。

- `clevis-luks-bind(1)`
9.8. 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、キックスタートと統合して、登録プロセスを完全に自動化にできます。

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定します。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-E2l6qjfdDiwVmidZjA"}' \ <<< "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。

9.9. 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。

関連情報
詳細は、以下の man ページを参照してください。

- clevis-luks-unlockers(7)

9.10. 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

NBDE を使用して、LUKS で暗号化した非 root ボリュームをアンロックするには、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. clevis luks bind コマンドを使用したバインド手順を実行します。詳細は、「LUKS で暗号化した root ボリュームの手動登録の設定」を参照してください。

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。この設定ファイルに _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。

関連情報
詳細は、以下の man ページを参照してください。

- clevis-luks-unlockers(7)

9.11. 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 マスター鍵を変更しません。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 マスター鍵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 ボリュームを自動アンロックする」)
リュームの自動登録の設定」と参照)、または暗号化した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。

関連情報
詳細は、以下の man ページを参照してください。
- clevis-luks-bind(1)

9.12. 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。他の仮想化環境と同様に、LUKS マスター鍵を共有しないように、1 つのイメージから起動するインスタンス数を減らすことが推奨されます。

したがって、ベストプラクティスは、どのパブリックリポジトリでも共有されず、限られたインスタンスのデプロイメントのベースを提供するように、イメージをカスタマイズすることです。作成するインスタンスの数は、デプロイメントのセキュリティーポリシーで定義する必要があります。また、LUKS マスター鍵の攻撃ベクトルに関連するリスク許容度に基づいて決定する必要があります。

LUKS に対応する自動デプロイメントを構築するには、Lorax、virt-install などのシステムとキックスタートファイルを一緒に使用し、イメージ構築プロセス中にマスター鍵の一意性を確保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。

9.13. 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)
第9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

- clevis-luks-unlockers(1)
第10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するのではありません。システムで使用されるセキュリティーポリシーの違反を発見するために使用できます。このような違反は、SELinuxなどの別のセキュリティー対策で防ぐことができます。

10.1. LINUX の AUDIT

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

以下は、Auditがログファイルに記録できる情報の概要です。

- イベントの日時、タイプ、結果
- サブジェクトとオブジェクトの機密性のラベル
- イベントを開始したユーザーのIDとイベントの関連性
- Audit設定の全修正およびAuditログファイルへのアクセス試行
- SSH、Kerberos、およびその他の認証メカニズムの全使用
- 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などの属性に基づくincludeまたはexcludeイベント

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

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

Auditは以下でも認定されています。

- National Information Assurance Partnership (NIAP) および Best Security Industries (BSI) による評価
- Red Hat Enterprise Linux 5におけるLSPP/CAPP/RSBAC/EAL4以降の認定
Red Hat Enterprise Linux 6 における OSPP/EAL4 以降 (Operating System Protection Profile / Evaluation Assurance Level 4 以降) 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がアクセス、修正、または実行されたか、もしく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、これららのファイルが破損した場合に監査証跡を入手可能とする際に役立ちます。

システムコールの監視

Audit は、一部のシステムコールが使用されるたびにログエントリーを生成するように設定できます。これを使用すると、settimeofday または 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。これにより作成されるログエントリーをユーザーセッション検索すると、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。

システムのパス名の実行の記録

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、ルールの呼び出し後にファイルが置き換えられた場合でも、Audit は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にかけ、いくつか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 ユーティリティおよび ebtables ユーティリティは、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。

10.2. AUDIT システムのアーキテクチャー

Audit システムは、ユーザー空間アプリケーションおよびユーティリティーと、カーネル側のシステムコール処理という 2 つの主要部分で構成されます。カーネルコンポーネントは、ユーザー空間アプリケーションからシステムコールを受け、これを user、task、fstype、または exit のいずれかのフィルタで振り分けます。
システムコールが exclude フィルターを通過すると、前述のフィルターのいずれかに送られます。このフィルターにより、Audit ルール設定に基づいてシステムコールが Audit デーモンに送信され、さらに処理されます。

ユーザー空間の Audit デーモンは、カーネルから情報を収集し、ログファイルのエントリーを作成します。他のユーザー空間ユーティリティーは、Audit デーモン、カーネルの Audit コンポーネント、または Audit ログファイルと相互作用します。

- auditctl - Audit 制御ユーティリティーはカーネル Audit コンポーネントと相互作用し、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を設定し、コンテキストも制御します。
- 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。

RHEL 8 では、Audit dispatcher デーモン (audisp) 機能は、Audit デーモン (auditd) に統合されています。監査イベントと、リアルタイムの分析プログラムの相互作用に使用される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。

10.3. セキュアな環境への AUDITD の設定

デフォルトの auditd 設定は、ほとんどの環境に適しています。ただし、厳格なセキュリティーポリシーに対応する必要がある場合は、/etc/audit/auditd.conf ファイルの Audit デーモン設定に以下の設定が推奨されます。

log_file
Audit ログファイル (通常は /var/log/audit/) を保持するディレクトリーは、別のマウントポイントにマウントされている必須があります。これにより、その他のプロセスがこのディレクトリー内の領域を使用しないようにし、Audit デーモンの残りの領域を正確に検出します。

max_log_file
1つの Audit ログファイルの最大サイズを指定します。Audit ログファイルを保持するパーティションで利用可能な領域をすべて使用するように設定する必要があります。

max_log_file_action
max_log_file に設定した制限に達すると実行するアクションを指定します。Audit ログファイルが上書きされないように keep_logs に設定する必要があります。

space_left
space_left_action パラメーターに設定したアクションが発生するディスクの空き容量を指定します。管理者は、ディスクの領域を反映して解放するのに十分な時間を設定する必要があります。space_left の値は、Audit ログファイルが生成される速度によって異なります。

space_left_action
空間が不足した場合に発生するアクションを指定します。管理者が実行するアクションのログを記録するために十分なサイズを残す必要があります。

admin_space_left_action
空間が不足した場合に発生するアクションを指定します。管理者が実行するアクションのログを記録するために十分なサイズを残す必要があります。

admin_space_left_action
space_left_action パラメーターの設定が不備な場合に発生するアクションを指定します。

disk_full_action
Audit ログファイルが含まれるパーティションに空き領域がない場合に発生するアクションを指定します。

Red Hat Enterprise Linux 8 セキュリティの強化
Auditログファイルが含まれるパーティションに空き領域がない場合に発生するアクションを指定します（haltまたはsingleに設定する必要があります）。これにより、Auditがイベントをログに記録できなくなると、システムは、シングルユーザーモードでシャットダウンまたは動作します。

disk_error_action
Auditログファイルが含まれるパーティションでエラーが検出された場合に発生するアクションを指定します。このパラメーターは、ハードウェアの機能不全処理に関するローカルのセキュリティーポリシーに基づいて、syslog、single、haltのいずれかに設定する必要があります。

flush
incremental_asyncに設定する必要があります。これはfreqパラメーターと組み合わせて機能します。これは、ハードドライブとのハード同期を強制する前にディスクに送信できるレコードの数を指定します。freqパラメーターは100に設定する必要があります。このパラメーターにより、アクティビティが集中した際に高いパフォーマンスを保ちつつ、Auditイベントデータがディスクのログファイルと確実に同期されるようになります。

10.4. AUDITDの開始および制御
auditdが設定されると、サービスを起動してAudit情報を収集し、ログファイルに保存します。rootユーザーで次のコマンドを実行し、auditdを起動します。

```
~# service auditd start
```
システムの起動時にauditdが開始するように設定するには、次のコマンドを実行します。

```
~# systemctl enable auditd
```
service auditd actionコマンドを使用すると、auditdでさまざまなアクションを実行できます。ここでのアクションは以下のいずれかになります。

stop
auditdを停止します。
restart
auditdを再起動します。
reloadまたはforce-reload
/etc/audit/auditd.confファイルからauditdの設定を再ロードします。
rotate
/var/log/audit/ディレクトリーログファイルをローテーションします。
resume
Auditイベントのログが一旦停止した後、再開します。たとえば、Auditログファイルが含まれるディスクパーティションの未使用領域が不足している場合などです。
condrestartまたはtry-restart
auditdがすでに起動している場合にのみ、これを再起動します。
status
auditdの稼働状況を表示します。

残りの設定オプションは、ローカルのセキュリティーポリシーに合わせて設定します。
service コマンドは、auditd デーモンと正しく相互作用する唯一の方法です。auid 値が適切に記録されるように、service コマンドを使用する必要があります。systemctl コマンドは、2 つのアクション (enable および status) にのみ使用できます。

10.5. AUDIT ログファイルについて

デフォルトでは、Audit システムはログエントリーを /var/log/audit/audit.log ファイルに保存します。ログローテーションが有効になっていれば、ローテーションされた audit.log ファイルは同じディレクトリに保存されます。

下記の Audit ルールを追加して、/etc/ssh/sshd_config ファイルの読み取りまたは修正の試行をすべてログに記録します。

```
# auditctl -w /etc/ssh/sshd_config -p warx -k sshd_config
```

auditd デーモンが実行している場合は、たとえば次のコマンドを使用して、Audit ログファイルに新しいイベントを作成します。

```
$ cat /etc/ssh/sshd_config
```

このイベントは、audit.log ファイルでは以下のようになります。

```
type=SYSCALL msg=audit(1364481363.243:24287): arch=c000003e syscall=2 success=no exit=-13 a0=7fffd19c5592 a1=0 a2=7fffd19c4b50 a3=a items=1 ppid=2686 pid=3538 auid=1000 uid=1000 gid=1000 euid=1000 suid=1000 fsuid=1000 sgid=1000 egid=1000 tty=pts0 ses=1 comm="cat" exe="/bin/cat" subj=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023 key="sshd_config"
type=CWD msg=audit(1364481363.243:24287): cwd="/home/shadowman"
type=PATH msg=audit(1364481363.243:24287): item=0 name="/etc/ssh/sshd_config" inode=409248 dev=fd:00 mode=0100600 ouid=0 ogid=0 rdev=0:00 obj=system_u:object_r:etc_t:s0 nametype=NORMAL cap_fp=none cap_fi=none cap_fe=0 cap_fver=0 type=PROCTITLE msg=audit(1364481363.243:24287) : proctitle=636174002F6574632F7373682F737368645F636F6E666967
```

上記のイベントは 4 つのレコードで構成されており、タイムスタンプとシリアル番号を共有します。レコードは、常に type=で始まります。各レコードには、スペースまたはコンマで区切られた名前と値のペア (name=value) が複数使用されています。上記のイベントの詳細な分析は以下のようにになります。

1つの目のレコード

```
type=SYSCALL
```

type フィールドには、レコードのタイプが記載されます。この例の SYSCALL 値は、カーネルへのシステムコールによりこれが記録されたことを示しています。

```
msg=audit(1364481363.243:24287):
```

msg フィールドには以下が記録されます。

- audit(time_stamp:ID) 形式のレコードのタイムスタンプおよび一意の ID。複数のレコードが同じ Audit イベントの一部として生成されている場合は、同じタイムスタンプおよび ID を共有できます。タイムスタンプは Unix の時間形式です (1970年1月1日 00:00:00 UTC からの秒数)。
カーネル空間およびユーザー空間のアプリケーションが提供するさまざまなイベント固有のname=valueペア。

arch=c000003e

archフィールドには、システムのCPUアーキテクチャーに関する情報が含まれます。c000003eの値は16進数表記で記録されます。ausearchコマンドでAuditレコードを検索する場合は、-iオプションまたは--interpretオプションを使用して、16進数の値を人間が判読できる値に自動的に変換します。c000003e値はx86_64として解釈されます。

syscall=2

syscallフィールドは、カーネルに送信されたシステムコールのタイプを記録します。値が2の場合は、/usr/include/asm/unistd_64.hファイルに、人間が判読できる値を指定できます。この場合の2はオープンなシステムコールです。ausyscallユーティリティーでは、システムコール番号を、人間が判読できる値に変換できます。ausyscall--dumpコマンドを使用して、システムコールの一覧とその数字を表示します。詳細は、manページのausyscall(8)を参照してください。

success=no

successフィールドは、その特定のイベントで記録されたシステムコールが成功したかどうかを記録します。この例では、呼び出しが成功していませんでした。

exit=-13

exitフィールドには、システムコールが返した終了コードを指定する値が含まれます。この値は、システムコールにより異なります。次のコマンドを実行すると、この値を人間が判読可能なものに変換できます。

~# ausearch --interpret --exit -13

この例では、監査ログに、終了コード-13で失敗したイベントが含まれていることが前提となります。

a0=7fffd19c5592, a1=0, a2=7fffd19c5592, a3=a

a0からa3までのフィールドは、このイベントにおけるシステムコールの最初の4つの引数を、16進法で記録します。この引数は、使用されるシステムコールにより異なります。ausearchユーティリティーで解釈できます。

items=1

itemsフィールドには、システムコールのレコードに続くPATH補助レコードの数が含まれます。

ppid=2686

ppidフィールドは、親プロセスID(PPID)を記録します。この例では、2686は、bashなどの親プロセスのPPIDです。

pid=3538

pidフィールドは、プロセスID(PID)を記録します。この例の3538はcatプロセスのPIDです。

auid=1000

auidフィールドには、loginuidであるAuditユーザーIDが記録されます。このIDは、ログイン時にユーザーに割り当てられ、ユーザーのIDが変更した後でもすべてのプロセスに引き継がれます(たとえば、su-johnコマンドでユーザーアカウントを切り替えた場合)。

uid=1000

uidフィールドは、解析しているプロセスを開始したユーザーのユーザーIDを記録します。ユーザーIDは、ausearch-i--uidUIDのコマンドを使用するとユーザー名に変換されます。

gid=1000

gidフィールドは、解析しているプロセスを開始したユーザーのグループIDを記録します。
euid=1000
euid フィールドは、解析しているプロセスを開始したユーザーの実効ユーザー ID を記録します。
suid=1000

suid フィールドは、解析しているプロセスを開始したユーザーのセットユーザー ID を記録します。
fsuid=1000

fsuid フィールドは、解析しているプロセスを開始したユーザーのファイルシステムユーザー ID を記録します。
egid=1000

egid フィールドは、解析しているプロセスを開始したユーザーの実効グループ ID を記録します。
sgid=1000

sgid フィールドは、解析しているプロセスを開始したユーザーのセットグループ ID を記録します。
fsgid=1000

fsgid フィールドは、解析しているプロセスを開始したユーザーのファイルシステムグループ ID を記録します。
tty=pts0

tty フィールドは、解析しているプロセスが開始したターミナルを記録します。

ses=1

ses フィールドは、解析しているプロセスが開始したセッションのセッション ID を記録します。

comm="cat"

comm フィールドは、解析しているプロセスを開始するために使用したコマンドのコマンドライン名を記録します。この例では、この Audit イベントを発生するのに、cat コマンドが使用されました。

exe="/bin/cat"

exe フィールドは、解析しているプロセスを開始するために使用した実行可能ファイルへのパスを記録します。

subj=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023

subj フィールドは、解析しているプロセスの実行時にラベル付けされた SELinux コンテンツを記録します。

key="sshd_config"

key フィールドは、Audit ログでこのイベントを生成したルールに関連付けられている管理者による定義の文字列を記録します。

2つ目のレコード

type=CWD

2つ目のレコードの type フィールドの値は、CWD(現在の作業ディレクトリー) です。このタイプは、最初のレコードで指定されたシステムコールを開始したプロセスの作業ディレクトリーを記録するために使用されます。

この記録の目的は、相対パスが関連する PATH 記録に保存された場合に、現行プロセスの位置を記録することにあります。これにより、絶対パスを再構築できます。

msg=audit(1364481363.243:24287)

msg フィールドは、最初のレコードと同じタイムスタンプと ID の値を持ちます。タイムスタンプは Unix の時間形式です (1970 年 1 月 1 日 00:00:00 UTC からの秒数)。

cwd="/home/user_name"

cwd フィールドは、システムコールが開始したディレクトリーのパスになります。
第10章 システムの監査

３つのレコード

type=PATH

３つのレコードでは、type フィールドの値は PATH です。Audit イベントには、システムコールに引数として渡されたすべてのパスに PATH タイプのレコードが含まれます。この Audit イベントでは、1つのパス (/etc/ssh/sshd_config) のみが引数として使用されます。

msg=audit(1364481363.243:24287):

msg フィールドは、1つ目と2つ目のレコードと同じタイムスタンプと ID になります。

item=0

item フィールドは、SYSCALL タイプレコードで参照されているアイテムの合計数のうち、現在のレコードがどのアイテムであるかを示します。この数はゼロベースで、0 は最初の項目であることを示します。

name="/etc/ssh/sshd_config"

name フィールドは、システムコールに引数として渡されたファイルまたはディレクトリーのパスを記録します。この場合、これは /etc/ssh/sshd_config ファイルです。

inode=409248

inode フィールドには、このイベントで記録されたファイルまたはディレクトリーに関連する inode番号が含まれます。以下のコマンドは、inode 番号 409248 に関連するファイルまたはディレクトリーを表示します。

```
~]$ find / -inum 409248 -print
/etc/ssh/sshd_config
```

dev=fd:00

dev フィールドは、このイベントで記録されたファイルまたはディレクトリーを含むデバイスのマイナーおよびメジャーの ID を指定します。ここでは、値が /dev/fd/0 のデバイスを示しています。

mode=0100600

mode フィールドは、ファイルまたはディレクトリーのパーミッションを、st_mode フィールドのstat コマンドが返す数値表記で記録します。詳細は、man ページの stat(2) を参照してください。この場合、0100600 は -rw------- として解釈されます。つまり、root ユーザーのみ、/etc/ssh/sshd_config ファイルに読み取りおよび書き込みのパーミッションが付与されます。

ouid=0

ouid フィールドは、オブジェクトの所有者のユーザー ID を記録します。

ogid=0

ogid フィールドは、オブジェクトの所有者のグループ ID を記録します。

rdev=00:00

rdev フィールドには、特定ファイルにのみ記録されたデバイス識別子が含まれます。ここでは、記録されたファイルは通常のファイルであるため、このフィールドは使用されません。

obj=system_u:object_r:etc_t:s0

obj フィールドは、実行時に、記録されているファイルまたはディレクトリーにラベル付けするSELinux コンテキストを記録します。

nametype=NORMAL

nametype フィールドは、指定したシステムコールのコンテキストで各パスのレコード操作の目的を記録します。

cap_fp=none

cap_fp フィールドは、ファイルまたはディレクトリーオブジェクトで許可されたファイルシステムベースの機能の設定に関連するデータを記録します。
cap_fi=none

cap_fi フィールドは、ファイルまたはディレクトリーオブジェクトの親を継承されたファイルシステムベースの機能の設定に関するデータを記録します。

cap_fe=0

cap_fe フィールドは、ファイルまたはディレクトリーオブジェクトのファイルシステムベースの機能の有効ビットの設定を記録します。

cap_fver=0

cap_fver フィールドは、ファイルまたはディレクトリーオブジェクトのファイルシステムベースの機能のバージョンを記録します。

4 つ目のレコード

type=PROCTITLE

type フィールドには、レコードのタイプが記載されます。この例の PROCTITLE 値は、このレコードにより、カーネルへのシステムコールにより発生するこの監査イベントを発生させた完全なコマンドラインを提供することが指定されることを示しています。

proctitle=636174002F6574632F7373682F737368645F636F6E666967

proctitle フィールドは、解析しているプロセスを開始するために使用したコマンドのコマンドラインを記録します。このフィールドは 16 进数の表記で記録され、Audit ログパーサーに影響が及ばないようにします。このテキストは、この Audit イベントを開始したコマンドに復号します。ausearch コマンドで Audit レコードを検索する場合は、-i オプションまたは --interpret オプションを使用して、16 进数の値を人間が判読できる値に自動的に変換します。636174002F6574632F7373682F737368645F636F6E666967 値は、cat /etc/ssh/sshd_config として解釈されます。

10.6. AUDITCTL で AUDIT ルールを定義および実行

Audit システムは、ログファイルで取得するものを定義する一連のルールで動作します。Audit ルールは、auditctl ユーティリティを使用してコマンドラインで設定するか、/etc/audit/rules.d/ ディレクトリで設定できます。

auditctl コマンドを使用すると、Audit システムの基本的な機能を制御し、どの Audit イベントをログに記録するかを指定するルールを定義できます。

ファイルシステムのルールの例

1. すべての書き込みアクセスと /etc/passwd ファイルのすべての属性変更をログに記録するルールを定義するには、次のコマンドを実行します。

   # auditctl -w /etc/passwd -p wa -k passwd_changes

2. すべての書き込みアクセスと、/etc/selinux/ ディレクトリー内の全ファイルへのアクセスと、その属性変更をすべてログに記録するルールを定義するには、次のコマンドを実行します。

   # auditctl -w /etc/selinux/ -p wa -k selinux_changes

システムロールのルールの例

1. システムで 64 ビットアーキテクチャーが使用され、システムコールの adjtimex または settimeofday がプログラムにより使用されるたびにログエントリーを作成するルールを定義するには、次のコマンドを実行します。

   # auditctl -w / -p wa -k adjtimex
実行可能なファイルルール

/bin/id プログラムのすべての実行をログに取得するルールを定義するには、次のコマンドを実行します。

```bash
# auditctl -F exe=/bin/id -S execve -k execution_bin_id
```

関連情報

パフォーマンスに関するヒントや、使用例などは、man ページの auditctl(8) を参照してください。

10.7. 永続的な AUDIT ルールの定義

再起動後も持続するように Audit ルールを定義するには、/etc/audit/rules.d/audit.rules ファイルに直接追加するか、/etc/audit/rules.d/ディレクトリーにあるルールを読み込む augenrules プログラムを使用する必要があります。

auditd サービスを開始すると、/etc/audit/audit.rules ファイルが生成されることに注意してください。/etc/audit/rules.d/のファイルは、同じ auditctl コマンドライン構文を使用してルールを指定します。ハッシュ記号 (#) に続く空の行とテキストは無視されます。

また、auditctl コマンドは、以下のように -Rオプションを使用して指定したファイルからルールを読み込むのに使用することもできます。

```bash
# auditctl -R /usr/share/doc/audit/rules/30-stig.rules
```

10.8. 事前に設定されたルールファイルの使用

/usr/share/doc/audit/rules/ディレクトリーには、audit パッケージが各種の証明書規格に従って、事前設定ルールのファイル形式が提供されています。

30-nispom.rules

NISPOM (National Industrial Security Program Operating Manual) の「Information System Security」の章で指定している要件を満たす Audit ルール設定

30-ospp-v42.rules

OSPP (Protection Profile for General Purpose Operating Systems) プロファイルバージョン 4.2 に定義されている要件を満たす監査ルール設定

30-pci-dss-v31.rules

PCI DSS (Payment Card Industry Data Security Standard) v3.1 に設定されている要件を満たす監査ルール設定

30-stig.rules
セキュリティー技術実装ガイド (STIG: Security Technical Implementation Guide) で設定されている要件を満たす Audit ルール設定

上記の設定ファイルを使用するには、/etc/audit/rules.d/ ディレクトリーにコピーして、以下のように augenrules --load コマンドを使用します。

```
# cp /usr/share/doc/audit/rules/10-base-config.rules /usr/share/doc/audit/rules/30-stig.rules
/etc/audit/rules.d/
# augenrules --load
```

関連情報

Audit ルールには、順序付けが可能な番号指定スキームがあります。命名スキームの詳細は、/usr/share/doc/audit/rules/README-rules ファイルを参照してください。

詳細、トラブルシューティング、およびその他の使用例は、man ページの audit.rules(7) を参照してください。

10.9. 永続ルールを定義する AUGENRULES の使用

augenrules スクリプトは、/etc/audit/rules.d/ ディレクトリーにあるルールを読み込み、audit.rules ファイルにコンパイルします。このスクリプトは、自然なソート順序の特定の順番で、.rules で終わるすべてのファイルを処理します。このディレクトリーのファイルは、以下の意味を持つグループに分類されます。

- 10 - カーネルおよび auditctl の設定
- 20 - 一般的なルールに該当してしま可能性もあるが、ユーザー側で独自ルールを作成することも可能
- 30 - 主なルール
- 40 - 任意のルール
- 50 - サーバー固有のルール
- 70 - システムのローカルルール
- 90 - ファイナライズ (不変)

ルールは、すべてを一度に使用することは意図されていません。ルールは考慮すべきポリシーの一部であり、個々のファイルは /etc/audit/rules.d/ にコピーされます。たとえば、STIG 設定でシステムを設定し、10-base-config, 30-stig, 31-privileged, 99-finalize の各ルールをコピーします。

/etc/audit/rules.d/ ディレクトリーにルールを置いたら、--load ディレクティブで augenrules スクリプトを実行することでそれを読み込みます。

```
# augenrules --load
/sbin/augenrules: No change
No rules
enabled 1
failure 1
pid 742
rate_limit 0
[trimmed for clarity]
```
関連情報
Audit ルールおよび augenrules スクリプトの詳細は、man ページの audit.rules(8) および augenrules(8) を参照してください。

10.10. 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)