



Red Hat Enterprise Linux 7

仮想化セキュリティガイド

仮想化環境のセキュリティ保護

Red Hat Enterprise Linux 7 仮想化セキュリティーガイド

仮想化環境のセキュリティー保護

Jiri Herrmann

Red Hat Customer Content Services

jherrman@redhat.com

Scott Radvan

Red Hat Customer Content Services

Tahlia Richardson

Red Hat Customer Content Services

Paul Moore

Red Hat エンジニアリング

Kurt Seifried

Red Hat エンジニアリング

David Jorm

Red Hat エンジニアリング

Thanks go to the following people for enabling the creation of this guide:

法律上の通知

Copyright © 2017 Red Hat, Inc.

This document is licensed by Red Hat under the [Creative Commons Attribution-ShareAlike 3.0 Unported License](#). If you distribute this document, or a modified version of it, you must provide attribution to Red Hat, Inc. and provide a link to the original. If the document is modified, all Red Hat trademarks must be removed.

Red Hat, as the licensor of this document, waives the right to enforce, and agrees not to assert, Section 4d of CC-BY-SA to the fullest extent permitted by applicable law.

Red Hat, Red Hat Enterprise Linux, the Shadowman logo, JBoss, OpenShift, Fedora, the Infinity logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries.

Linux ® is the registered trademark of Linus Torvalds in the United States and other countries.

Java ® is a registered trademark of Oracle and/or its affiliates.

XFS ® is a trademark of Silicon Graphics International Corp. or its subsidiaries in the United States and/or other countries.

MySQL ® is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js ® is an official trademark of Joyent. Red Hat Software Collections is not formally related to or endorsed by the official Joyent Node.js open source or commercial project.

The OpenStack ® Word Mark and OpenStack logo are either registered trademarks/service marks or trademarks/service marks of the OpenStack Foundation, in the United States and other countries and are used with the OpenStack Foundation's permission. We are not affiliated with, endorsed or sponsored by the OpenStack Foundation, or the OpenStack community.

All other trademarks are the property of their respective owners.

概要

本ガイドは、Red Hat が提供する仮想化セキュリティーテクノロジーについての概要を説明します。仮想化環境内のホスト、ゲスト、および共有インフラストラクチャー/リソースのセキュリティーを保護するための推奨事項を提供します。

目次

第1章 はじめに	3
1.1. 仮想化環境と非仮想化環境	3
1.2. 仮想化セキュリティが重要である理由	4
1.3. SVIRT を使用した SELINUX の活用	5
第2章 ホストのセキュリティ	6
2.1. ホストのセキュリティが重要である理由	6
2.2. ホスト物理マシンのセキュリティ保護	6
2.3. RED HAT ENTERPRISE LINUX のホストセキュリティ推奨プラクティス	8
第3章 ゲストのセキュリティ	11
3.1. ゲストのセキュリティが重要である理由	11
3.2. ゲストセキュリティの推奨プラクティス	11
第4章 SVIRT	12
4.1. 概要	12
4.2. SELINUX と強制アクセス制御 (MAC)	12
4.3. SVIRT の設定	13
4.4. SVIRT のラベル	14
第5章 仮想化環境におけるネットワークセキュリティ	17
5.1. ネットワークセキュリティの概要	17
5.2. ネットワークセキュリティ推奨プラクティス	17
付録A 追加情報	18
A.1. SELINUX および SVIRT	18
A.2. 仮想化セキュリティ	18
付録B 改訂履歴	19

第1章 はじめに

1.1. 仮想化環境と非仮想化環境

仮想化環境は、攻撃者にとって以前は価値がなかった新たな攻撃ベクトルの発見と既存の 익스プロイトの洗練の両方の機会を与えます。このため、仮想マシンを作成し、これを維持する際には、物理ホストとそのホスト上で実行されるゲストの両方のセキュリティーを確保するための対策を講じることが重要となります。

非仮想化環境

非仮想化環境では、ホストは物理的に相互分離しており、各ホストには Web サーバーや DNS サーバーなどのサービスで構成される自己完結型の環境があります。これらのサービスは、独自のユーザースペース、ホストカーネル、物理ホストと直接通信して、ネットワークにサービスを直接提供します。下図は、非仮想化環境を示しています。

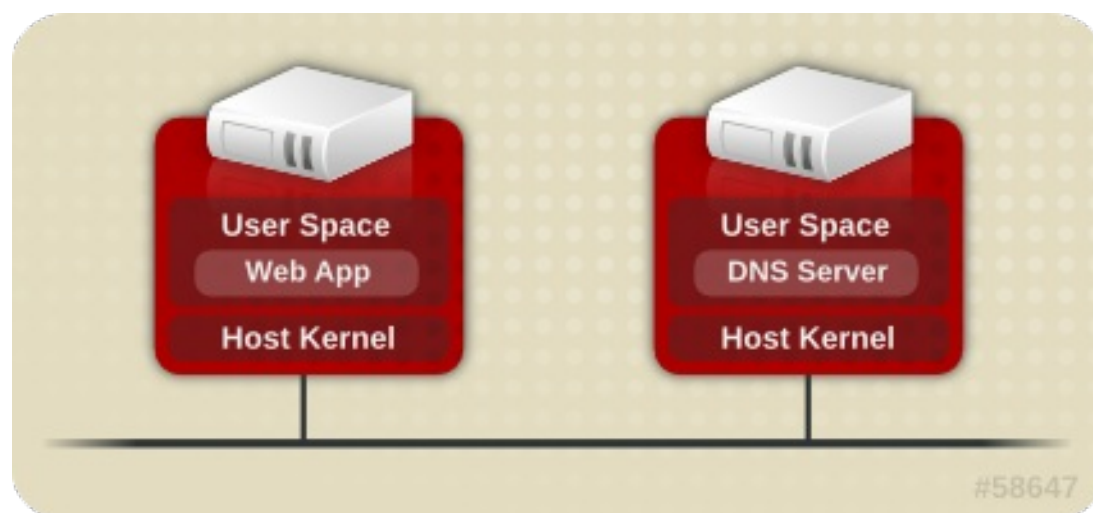


図1.1 非仮想化環境

仮想化環境

仮想化環境では、複数のオペレーティングシステムを(「ゲスト」として)単一のホストカーネルおよび物理ホストに格納することができます。下図は仮想化環境を示しています。

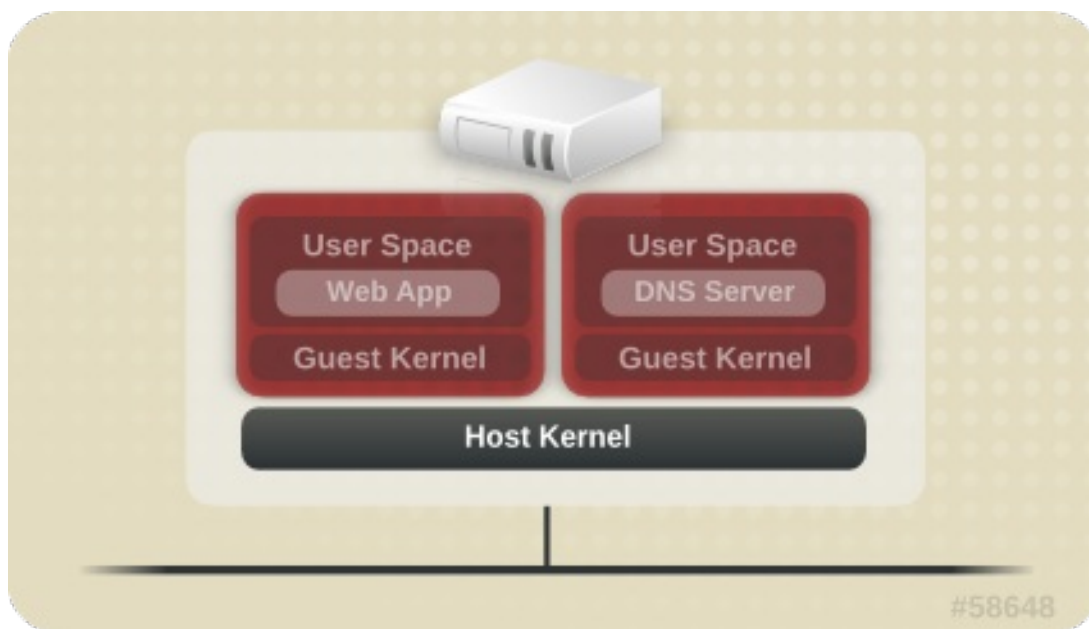


図1.2 仮想化環境

サービスが仮想化されていない場合は、マシンは物理的に分離されています。したがって、エクスプロイトは影響を受けたマシンに抑えられます。ただし、ネットワーク攻撃は明らかな例外となります。仮想化環境内でサービスがグループ化されると、システムの脆弱性が高まります。ハイパーバイザーのセキュリティに不備がある場合、ゲストインスタンスによるエクスプロイトを受ける可能性があり、そのゲストはホストのみならず、そのホスト上で実行されている他のゲストも攻撃できるようになる可能性があります。これは単に理論上の事柄ではなく、攻撃はすでにハイパーバイザー上に存在しています。それらの攻撃がゲストインスタンスを超えて、他のゲストが攻撃にさらされる可能性もあり得ます。

1.2. 仮想化セキュリティが重要である理由

インフラストラクチャーに仮想化をデプロイすると、数多くのメリットがもたらされますが、新たなリスクが生じる可能性もあります。仮想化のリソースとサービスのデプロイにあたっては以下のようなセキュリティに関する考慮事項を検討した上でデプロイを行う必要があります。

- ホスト/ハイパーバイザーは第一のターゲットであり、ゲストとデータの単一障害点となることが多くあります。
- 仮想マシンは望ましくない方法で相互干渉する場合があります。これを防ぐためのアクセス制御が導入されていないとすると、悪意のあるゲストが脆弱なハイパーバイザーをバイパスし、他のゲストのストレージなど、ホストシステム上の他のリソースに直接アクセスする可能性があります。
- 仮想化システムを迅速にデプロイすると、十分なパッチ、モニタリング、メンテナンスなどのリソース管理の必要性が増大するため、リソースとサービスのトラッキングおよび維持管理が難しくなる場合があります。
- 仮想化環境における技術スタッフの知識不足、技能の格差、経験不足などの問題が存在する可能性があります。このような問題は、多くの場合、脆弱性へとつながります。
- ストレージなどのリソースが複数のマシンに散在し、それらのマシンに依存している場合があります。このような場合には環境が過度に複雑化してしまい、システムの管理とメンテナンスが不十分となる可能性があります。

- 仮想化によって、環境内に存在する従来のセキュリティーリスクは排除されません。仮想化レイヤーのみでなく、ソリューションスタック全体のセキュリティーを保護する必要があります。

本ガイドは、仮想化インフラストラクチャーのセキュリティー保護に役立つ **Red Hat Enterprise Linux** の仮想化推奨プラクティスを紹介し、お客様のセキュリティーリスクを軽減することを目的としています。

1.3. SVIRT を使用した SELINUX の活用

sVirt は仮想化を SELinux (Security-Enhanced Linux) によって提供されている既存のセキュリティーフレームワークに組み込むことにより、**強制アクセス制御(MAC)** を仮想マシンに適用します。sVirt の主な目的は、ハイパーバイザーのセキュリティーの脆弱性を利用した攻撃からホストとゲストを保護することです。SELinux は異なるプロセス全体にわたってアクセスポリシーを適用することでシステムを保護します。sVirt は、各ゲストをプロセスとして扱うことによりこの機能をホストとゲストにまで拡張し、悪意のあるゲストが制限付きリソースにアクセスするのを防ぐために設計されたのと同様のポリシーを管理者が適用できるようにします。sVirt についての詳しい情報は「[4章 sVirt](#)」を参照してください。

第2章 ホストのセキュリティ

2.1. ホストのセキュリティが重要である理由

仮想化テクノロジーをデプロイする際、ホストのセキュリティは最優先事項です。Red Hat Enterprise Linux のホストシステムは、物理デバイス、ストレージ、ネットワークへのアクセスに加えて、全仮想化ゲスト自体を管理および制御します。ホストシステムのセキュリティが侵害されると、ホストシステムのみでなくゲストとそのデータまでもが攻撃を受ける可能性があります。

仮想化ゲストのセキュリティはホストシステムにかかっています。Red Hat Enterprise Linux ホストシステムのセキュリティ保護は、セキュアな仮想化プラットフォームの確立に向けた第一歩です。

2.2. ホスト物理マシンのセキュリティ保護

Red Hat Enterprise Linux ホストのパフォーマンスを強化するには次のようなタスクやヒントが役立ちます。

- 強制 (enforcing) モードで SELinux を実行します。setenforce コマンドを使って SELinux を強制 (enforcing) モードで実行するように設定します。

```
#setenforce 1
```

- すべての不必要なサービスを削除するか、または無効にします (AutoFS、NFS、FTP、HTTP、NIS、telnetd、sendmail など)。
- サーバー上にはプラットフォームの管理に必要な最低限のユーザーアカウントのみを追加します。不必要なユーザーアカウントは削除してください。
- ホストでは不必要なアプリケーションは実行しないようにしてください。ホストでアプリケーションを実行すると仮想マシンのパフォーマンスに影響を与えるため、その影響がサーバーの安定性に及ぶ可能性があります。サーバーをクラッシュさせる可能性のあるアプリケーションは、サーバー上のすべての仮想マシンをダウンさせてしまう原因ともなります。
- 仮想マシンのインストールおよびイメージには集中管理できる場所を使用します。仮想マシンのイメージは `/var/lib/libvirt/images/` に格納してください。仮想マシンのイメージをこれ以外のディレクトリに格納する場合は、そのディレクトリを SELinux ポリシーに追加し、インストールを開始する前にラベルの再設定を必ず行ってください。集中管理ができる共有可能なネットワークストレージの使用を強くお勧めします。



注記

パフォーマンスに関するヒントの詳細は、『[Red Hat Enterprise Linux 仮想化のチューニングと最適化ガイド](#)』を参照してください。

2.2.1. セキュリティ導入計画

仮想化技術を導入する際には、ホスト物理マシンとそのオペレーティングシステムが攻撃されないことを確認する必要があります。この場合、ホスト物理マシンとは、システム、デバイス、メモリー、ネットワークのほかにすべてのゲスト仮想マシンを管理する Red Hat Enterprise Linux システムのことです。ホスト物理マシンが保護されていないと、システム内のすべてのゲスト仮想マシンが脆弱になります。仮想化を使用してシステムのセキュリティを強化する方法はいくつかあります。担当者または担当者の企業は導入計画を作成する必要があります。この導入計画には、以下が含まれている必要があります。

- 動作仕様
- ご使用のゲスト仮想マシンに必要なサービスの指定
- ホスト物理サーバーとこれらのサービスに必要なサポートの指定

以下は、導入計画の作成時に考慮すべきセキュリティ問題です。

- ホスト物理マシン上では必要となるサービスのみを実行する。ホスト物理マシン上で実行されているプロセスやサービスが少ないほど、セキュリティのレベルとパフォーマンスが高くなります。
- ハイパーバイザー上で SELinux を有効にする。SELinux と仮想化の使い方については、「[SELinux と強制アクセス制御 \(MAC\)](#)」を参照してください。
- ファイアウォールを使用してホスト物理マシンへのトラフィックを制限する。攻撃からホスト物理マシンを保護するデフォルト拒否ルールでファイアウォールをセットアップできます。また、ネットワークを介するサービスを制限することも重要です。
- 標準ユーザーのホストのオペレーティングシステムに対するアクセスを許可しない。ホストのオペレーティングシステムの特権アカウントが設定されている場合、特権のないアカウントにアクセスを許可すると、セキュリティが危険にさらされる可能性があります。

2.2.2. クライアントアクセス制御

libvirt のクライアントアクセス制御フレームワークでは、システム管理者は複数のクライアントユーザー、管理オブジェクト、および API 操作に対する細かい権限のルールをセットアップすることができます。これにより、クライアントの接続を最小限の特権セットに制限することができます。

デフォルト設定では、**libvirtd** デーモンには 3 つのレベルのアクセス制御があります。最初は、すべての接続が非認証状態で行われます。この状態では、認証を完了するために必要な API 操作のみが許可されます。認証が成功した後は、クライアント接続に使用されたソケットによって、接続はすべての API 呼び出しへの完全な無制限アクセスを持つか、または「読み取り専用」操作に制限されます。アクセス制御フレームワークでは、管理者が、認証された接続に細かいアクセス権ルールを定義することができます。libvirt のすべての API 呼び出しには、使用されるオブジェクトに対して検証される一連の権限があります。さらに、特定のフラグが API 呼び出しに設定されているかについて権限のチェックが行われます。API 呼び出しに渡されるオブジェクトのチェックのほかにも、一部のメソッドは結果をフィルタリングします。

2.2.2.1. アクセス制御ドライバー

アクセス制御フレームワークは、今後追加される任意のアクセス制御技術との統合を可能にするためにプラグ可能なシステムとして設計されています。デフォルトでは、ドライバーは使用されません。そのため、アクセス制御のチェックは全く行われません。libvirt は **polkit** を実際のアクセス制御ドライバーとして使用するためのサポートを提供します。**polkit** アクセス制御ドライバーの使用方法については、[設定ドキュメント](#) を参照してください。

アクセス制御ドライバーは、**access_drivers** パラメーターを使用して **libvirtd.conf** 設定ファイルで設定されます。このパラメーターは、さまざまなアクセス制御ドライバー名を受け入れます。複数のアクセス制御ドライバーが必要とされる場合は、すべてのアクセス制御ドライバーのアクセスが付与されるようにそれらが処理される必要があります。「**polkit**」をドライバーとして有効にするには、以下のコマンドを実行します。

```
# augtool -s set '/files/etc/libvirt/libvirtd.conf/access_drivers[1]'
polkit
```

ドライバーをデフォルト (アクセス制御なし) に戻すには、以下のコマンドを入力します。

```
# augtool -s rm /files/etc/libvirt/libvirtd.conf/access_drivers
```

libvirtd.conf に変更を加えると、**libvirtd** デーモンの再起動が必要になることに注意してください。

2.2.2.2. オブジェクトおよびアクセス権

libvirt は、アクセス制御を、その API の主なオブジェクトタイプすべてに適用します。それぞれのオブジェクトタイプには、それぞれ一連の権限が定義されます。特定の API 呼び出しについてチェックされる権限を判別するには、該当 API の API 参照マニュアル文書を参照してください。オブジェクトと権限の詳細の一覧は、libvirt.org を参照してください。

2.2.2.3. ブロックデバイスをゲストに追加する際のセキュリティ上の懸念事項

- ホスト物理マシンでは、ファイルシステムを特定するために、**fstab** ファイルや、**initrd** ファイルまたはカーネルコマンドラインなどでファイルシステムのラベルを使用しないようにしてください。ゲスト仮想マシンがパーティションや LVM ボリューム全体への書き込みアクセスを持つ場合、ファイルシステムのラベルを使用するとセキュリティ上のリスクが発生します。ゲスト仮想マシンが、ホスト物理マシンに属するファイルシステムのラベルを独自のブロックデバイスストレージに書き込む可能性があるためです。これにより、ホスト物理マシンの再起動時に、ホスト物理マシンがこのゲスト仮想マシンのディスクをシステムディスクとして誤って使用してしまう可能性があり、ホスト物理マシンシステムが危険にさらされる可能性があります。

fstab ファイル、**initrd** ファイルまたはカーネルコマンドラインなどで使用する場合、デバイスの識別にはその **UUID** を使用した方がよいでしょう。それでも、特定のファイルシステムでは **UUID** の使用が完全に安全であるとは言えませんが、**UUID** を使用した場合には同様のセキュリティ侵害の可能性は確実に低くなります。

- ゲスト仮想マシンはディスク全域、またはブロックデバイス全域 (例: **/dev/sdb**) に書き込みアクセスを持つべきではありません。ブロックデバイス全域にアクセスを持つゲスト仮想マシンはボリュームラベルを修正できる場合があります、これがホスト物理マシンシステムの攻撃に使用される可能性があります。パーティション (例: **/dev/sdb1**) または LVM ボリュームを使用して、この問題を回避してください。LVM 管理および設定例については、「[CLI コマンドでの LVM 管理](#)」または「[LVM 設定の例](#)」を参照してください。

/dev/sdb1 または **/dev/sdb** などの raw ディスクなどのパーティションへの raw アクセスを使用する場合、**global_filter** 設定を使用して、安全なディスクのみをスキャンできるよう LVM を設定する必要があります。**global_filter** コマンドを使用した LVM 設定スクリプトの例については、「[サンプル lvm.conf ファイル](#)」を参照してください。

2.3. RED HAT ENTERPRISE LINUX のホストセキュリティ推奨プラクティス

ホストのセキュリティは、セキュアな仮想化インフラストラクチャーの極めて重要な要素であるため、以下の推奨プラクティスは Red Hat Enterprise Linux ホストシステムのセキュリティ保護の開始点として役立ちます。

- ゲストシステムの使用と管理のサポートに必要なサービスのみを実行します。ファイルサービスや印刷サービスなどのサービスを追加で提供する必要がある場合には、それらのサービスを Red Hat Enterprise Linux ゲストで実行することを検討した方がよいでしょう。

- システムへの直接のアクセスはシステムの管理を行う必要がある人に制限してください。共有の **root** アクセスを無効にして、代わりに **sudo** などのツールを使用して、管理ロールに基づいて管理者に特権的アクセスを付与することを検討してください。
- SELinux がご使用のインストールに応じて適切に設定され、**enforcing** モードで稼働していることを確認します。これは、適正なプラクティスである上、**sVirt** によって提供される高度な仮想化セキュリティー機能は SELinux に依存しています。SELinux と sVirt に関する詳しい情報は「[4章 sVirt](#)」を参照してください。
- ホストシステムで監査が有効化され、**libvirt** が監査レコードを生成するように設定されていることを確認します。監査が有効化されると、**libvirt** はゲストの設定変更および起動/停止イベントの監査レコードを生成します。これは、ゲストの状態をトラッキングするのに役立ちます。また、**libvirt** の監査イベントは、標準の監査ログ検査ツール以外に、専用の **auvirt** ツールでも確認することができます。
- システムのリモート管理はすべてセキュアなネットワークチャネル上のみで実行されるようにしてください。SSH のようなツールや、TLS または SSL などのネットワークプロトコルは認証とデータ暗号化の両方を提供し、承認済みの管理者のみがシステムをリモートで管理できるようにするのに役立ちます。
- ご使用のインストールに応じてファイアウォールが適切に設定されており、ブート時にアクティブ化されることを確認します。システムの使用および管理に必要なネットワークポートのみを許可する必要があります。
- ディスク全体またはブロックデバイス (例: **/dev/sdb**) への直接のアクセスをゲストに許可するのは控えて、代わりにゲストストレージにはパーティション (例: **/dev/sdb1**) や LVM ボリュームを使用します。
- スタッフが仮想化環境における十分なトレーニングを受けており、知識が十分にあることを確認してください。



警告

SR-IOV が仮想マシンで利用不可能な場合に USB デバイス、物理ファンクションまたは物理デバイスをアタッチすると、デバイスへのアクセスが提供され、これでファームウェアを上書きすることができます。これは、攻撃者が悪意のあるコードによってデバイスのファームウェアを上書きし、仮想マシン間でデバイスを移動する際やホストの起動時に問題を生じさせるという潜在的なセキュリティー上の問題を引き起こします。可能な場合は、SR-IOV 仮想ファンクションデバイス割り当ての使用が推奨されます。



注記

本ガイドは、ほとんどの仮想化環境でみられるセキュリティー関連の課題、脆弱性、解決策と推奨される対処方法について説明することを目的としています。ただし、Red Hat Enterprise Linux システムのセキュリティーを保護する際に従うべき推奨プラクティスが数多くあり、これらはスタンドアロン、仮想化ホスト、ゲストインスタンスを問わず適用されます。これらの推奨プラクティスにはシステムの更新、パスワードのセキュリティー、暗号化、ファイアウォールの設定などが含まれます。この情報については、『[Red Hat Enterprise Linux セキュリティーガイド](#)』で詳しく説明しています。

2.3.1. パブリッククラウドオペレーター向けの特殊な考慮事項

パブリッククラウドサービスオペレーターは、従来の仮想化ユーザーのリスクを超える数多くのセキュリティリスクにさらされます。悪意のあるゲストの脅威や、仮想化インフラストラクチャー全体にわたる顧客データの機密性および整合性に対する要件により、ホスト/ゲスト間ならびにゲスト間における仮想ゲストの分離は極めて重要となります。

パブリッククラウドオペレーターは上記の **Red Hat Enterprise Linux** 仮想化推奨プラクティスに加えて、以下の点も考慮する必要があります。

- ゲストからハードウェアへの直接のアクセスを無効にしてください。PCI、USB、FireWire、Thunderbolt、eSATA などのデバイスパススルーメカニズムは、管理を難しくする上、多くの場合は基礎となるハードウェアに依存してゲスト間の分離を強制します。
- クラウドオペレーターのプライベート管理ネットワークを顧客のゲストネットワークと分離して、顧客ネットワークを相互に分離することにより、以下が可能になります。
 - ゲストがネットワーク経由でホストシステムにアクセスできないようにする。
 - 顧客がクラウドプロバイダーの内部ネットワーク経由で別の顧客のゲストシステムに直接アクセスできないようにする。

第3章 ゲストのセキュリティ

3.1. ゲストのセキュリティが重要である理由

ホストシステムのセキュリティは、そのホスト上で実行されているゲストを確実にセキュリティ保護するために極めて重要となりますが、ホストのセキュリティによって個別のゲストマシンの適切なセキュリティ保護の必要性がなくなる訳ではありません。システムを仮想化ゲストとして実行する場合、従来の非仮想化システムに関連するセキュリティ上のリスクはすべて依然として存在します。ゲストシステムのセキュリティが侵害されると、ビジネスデータや顧客の機密情報など、ゲストシステムにアクセス可能なリソースはいずれも攻撃を受けやすくなる可能性があります。

3.2. ゲストセキュリティの推奨プラクティス

『Red Hat Enterprise Linux セキュリティガイド』に記載の Red Hat Enterprise Linux システムのセキュリティ保護に関する推奨プラクティスはすべて、従来の非仮想化システムと仮想化ゲストとしてインストールされたシステムの両方に適用されますが、仮想化環境内でゲストを実行する場合に極めて重要となるセキュリティ関連のプラクティスがいくつかあります。

- ゲストの管理はすべてリモートで実行される可能性が高いため、システムの管理は必ずセキュリティ保護されたネットワークチャネルで行うようにしてください。SSH などのツールや TLS または SSL などのネットワークプロトコルは、認証とデータの暗号化の両方を提供し、承認された管理者のみがシステムをリモートで管理できるようにします。
- 一部の仮想化テクノロジーでは、特殊なゲストエージェントまたはドライバーを使用して仮想化固有の機能を有効にします。このようなエージェントやアプリケーションは、SELinux のような Red Hat Enterprise Linux の標準のセキュリティ機能を使用して確実に保護してください。
- 仮想化環境では、ゲストシステムの保護境界線の外部から機密データがアクセスされるリスクがより高くなります。保管されている機密データは **dm-crypt** や **GnuPG** などの暗号化ツールを使用して保護してください。ただし、暗号化キーの機密性の確保には特に注意が必要です。



注記

Kernel Same-page Merging (KSM) のようなページ重複排除技術を使用すると、サイドチャネルが導入され、ゲストの情報漏洩に使用される可能性があります。これが懸念される場合には、ゲストごとまたは全体で KSM を無効にすることができます。KSM についての詳細情報は、『[Red Hat Enterprise Linux 7 仮想化のチューニングと最適化ガイド](#)』を参照してください。

第4章 SVIRT

4.1. 概要

KVM 下の仮想マシンは Linux プロセスとして実装されているため、KVM は標準の Linux セキュリティモデルを活用して分離とリソースの制御を行います。Linux カーネルには、米国国家安全保障局によって開発されたプロジェクトである SELinux (Security-Enhanced Linux) が搭載されており、柔軟性の高いカスタマイズ可能なセキュリティポリシーを通して、強制アクセス制御 (MAC)、マルチレベルセキュリティ (MLS)、およびマルチカテゴリーセキュリティ (MCS) を追加します。SELinux は、Linux カーネル上で実行されるプロセス (仮想マシンプロセスを含む) を対象とした、リソースの厳重な分離および隔離を行います。sVirt プロジェクトは SELinux を基盤として、仮想マシンの分離と制御された共有をさらに促進します。たとえば、粒度の細かいパーミッションを適用して仮想マシンをグループ化し、リソースを共有することができます。

セキュリティの観点からすると、ハイパーバイザーは攻撃者の格好的対象です。これは、ハイパーバイザーがセキュリティ侵害を受けると、そのホストシステム上で実行されている全仮想マシンのセキュリティも被害を受けることになる可能性があるためです。仮想化テクノロジーに SELinux を組み込むと、ホストシステムや他の仮想マシンへのアクセスを試みる悪意のある仮想マシンに対するハイパーバイザーのセキュリティを強化するのに役立ちます。

以下の図は、ゲストを分離することによって、セキュリティ侵害されたハイパーバイザー (またはゲスト) がさらなる攻撃を加えたり、別のインスタンスにまで被害を拡大したりする能力を抑える仕組みを示しています。

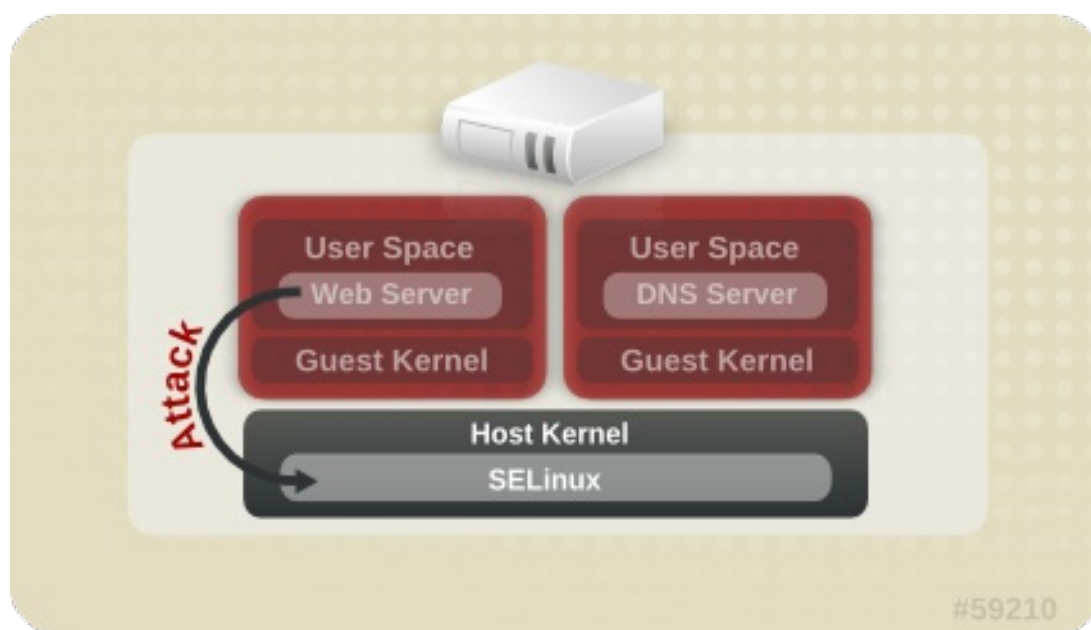


図4.1 SELinux によって分離される攻撃パス



注記

SELinux の詳細については、『[Red Hat Enterprise Linux SELinux ユーザーおよび管理者のガイド](#)』を参照してください。

4.2. SELINUX と強制アクセス制御 (MAC)

Security-Enhanced Linux (SELinux) は、Linux カーネルにおける MAC の実装です。標準の任意アクセス制御 (DAC) がチェックされたあとに、許可された操作をチェックします。SELinux は、実行中のプロセスとそれらの動作 (例: ファイルシステムオブジェクトへのアクセスを試みるなど) に対して、ユー

ザーがカスタマイズ可能なセキュリティーポリシーを適用することができます。Red Hat Enterprise Linux では SELinux がデフォルトで有効化されており、アプリケーションやシステムサービス (例: ハイパーバイザー) の脆弱性の悪用によって生じる可能性のある潜在的被害の範囲を制限します。

sVirt は、仮想化管理用の抽象化レイヤーである libvirt と一体化して、仮想マシン用の MAC フレームワークを提供します。このアーキテクチャーは、libvirt によってサポートされている全仮想化プラットフォームと、sVirt によりサポートされている全 MAC 実装が相互運用可能となります。

4.3. SVIRT の設定

SELinux ブール値は、オン/オフ切り替えが可能な変数で、機能やその他の特殊条件を迅速に有効化/無効化することができます。ブール値は、一時的な変更の場合は `setsebool boolean_name {on|off}`、再起動時に変更を永続化する場合は `setsebool -P boolean_name {on|off}` のいずれかを実行することによって切り替えることができます。

以下の表は、libvirt で始動された場合に KVM に影響する SELinux ブール値を示しています。これらのブール値 (オンまたはオフ) の現在の状態は、コマンド `getsebool -a|grep virt` を実行することにより確認できます。

表4.1 KVM SELinux のブール値

SELinux のブール値	説明
staff_use_svirt	staff ユーザーによる sVirt ドメイン作成、およびそのドメインへの移行を有効にします。
unprivuser_use_svirt	非特権ユーザーによる sVirt ドメイン作成、およびそのドメインへの移行を有効にします。
virt_sandbox_use_audit	サンドボックスコンテナによる監査メッセージの送信を有効にします。
virt_sandbox_use_netlink	サンドボックスコンテナによるネットリンクシステム呼び出しの使用を有効にします。
virt_sandbox_use_sys_admin	サンドボックスコンテナによる <code>sys_admin</code> システム呼び出し (<code>mount</code> 等) の使用を有効にします。
virt_transition_userdomain	ユーザードメインとしての仮想プロセスの実行を有効にします。
virt_use_comm	virt によるシリアルおよびパラレル通信ポートの使用を有効にします。
virt_use_execmem	制限された仮想ゲストによる実行可能メモリおよび実行可能スタックの使用を有効にします。
virt_use_fusefs	virt による FUSE マウントされたファイルの読み取りを有効にします。

SELinux のブール値	説明
virt_use_nfs	virt による NFS マウントされたファイルの管理を有効にします。
virt_use_rawip	virt による rawip ソケットとの通信を有効にします。
virt_use_samba	virt による CIFS マウントされたファイルの管理を有効にします。
virt_use_sanlock	制限された仮想ゲストによる sanlock との通信を有効にします。
virt_use_usb	virt による USB デバイスの使用を有効にします。
virt_use_xserver	仮想マシンによる X Window System との通信を有効にします。



注記

SELinux ブール値の詳細については、『[Red Hat Enterprise Linux SELinux ユーザーおよび管理者のガイド](#)』を参照してください。

4.4. SVIRT のラベル

SELinux の保護下にある他のサービスと同様に、sVirt はプロセススペースのメカニズム、ラベル、制限を使用してセキュリティを強化し、ゲストインスタンスを制御します。ラベルは、現在実行中の仮想マシンに基づいて、システム上のリソースに自動的に適用されます (動的) が、管理者が手動で指定して (静的)、特別な要件がある場合でも対応することが可能です。

4.4.1. sVirt ラベルのタイプ

以下の表には、仮想マシンのプロセス、イメージファイル、共有コンテンツなどのリソースに割り当てることができる、異なる sVirt ラベルについての説明をまとめています。

表4.2 sVirt ラベル

タイプ	SELinux コンテキスト	説明/効果
仮想マシンプロセス	system_u:system_r:svirt_t:MCS1	MCS1は無作為に選択されたフィールドです。現在は、約 500,000 のラベルがサポートされています。
仮想マシンのイメージ	system_u:object_r:svirt_image_t:MCS1	これらのイメージファイルやデバイスの読み取り/書き込みができるのは、同じ MCS1 フィールドが付いた svirt_t プロセスのみです。

タイプ	SELinux コンテキスト	説明/効果
仮想マシンの共有読み取り/書き込みコンテンツ	<code>system_u:object_r:svirt_image_t:s0</code>	<code>svirt_t</code> プロセスはすべて、 <code>svirt_image_t:s0</code> のファイルおよびデバイスに書き込むことができます。
仮想マシンの共有読み取り専用コンテンツ	<code>system_u:object_r:svirt_content_t:s0</code>	<code>svirt_t</code> プロセスはすべて、このラベルがついたファイル/デバイスを読み取ることができます。
仮想マシンのイメージ	<code>system_u:object_r:virt_content_t:s0</code>	イメージが存在する場合に使用されるシステムのデフォルトラベルです。 <code>svirt_t</code> 仮想プロセスは、このラベルの付いたファイル/デバイスを読み取ることはできません。

4.4.2. 動的設定

動的ラベル設定は、`sVirt` を `SELinux` と併用する場合のデフォルトのラベルオプションです。以下の例は、動的ラベリングを示しています。

```
# ps -eZ | grep qemu-kvm

system_u:system_r:svirt_t:s0:c87,c520 27950 ? 00:00:17 qemu-kvm
```

この例では、`qemu-kvm` プロセスに `system_u:system_r:svirt_t:s0` のベースラベルが付いています。`libvirt` システムは、このプロセス用に一意の `MCS` ラベル `c87,c520` を生成しています。ベースラベルと `MCS` ラベルを組み合わせることにより、そのプロセス用の完全なセキュリティーラベルが形成されます。同様に、`libvirt` は同じ `MCS` ラベルとベースラベルを使用してイメージラベルを形成します。このイメージラベルは次に、ディスクイメージやディスクデバイス、`PCI` デバイス、`USB` デバイス、`kernel/initrd` ファイルなど、仮想マシンがアクセスする必要のある全ホストファイルに自動的に適用されます。各プロセスは、異なるラベルを使用して、他の仮想マシンから分離されます。

以下の例は、`/var/lib/libvirt/images` 内のゲストディスクイメージに適用された、仮想マシンの一意のセキュリティーラベル(この場合は、対応する `MCS` ラベルが `c87,c520`)を示しています。

```
# ls -lZ /var/lib/libvirt/images/*

system_u:object_r:svirt_image_t:s0:c87,c520 image1
```

以下の例は、ゲストの `XML` 設定内の動的ラベルを示しています。

```
<seclabel type='dynamic' model='selinux' relabel='yes'>
  <label>system_u:system_r:svirt_t:s0:c87,c520</label>
  <imagelabel>system_u:object_r:svirt_image_t:s0:c87,c520</imagelabel>
</seclabel>
```

4.4.3. ベースラベルを使用した動的設定

デフォルトのダイナミックモードのベースセキュリティーラベルを上書きするには、以下の例に示したように、XML ゲスト設定内の **<baselabel>** オプションを手動で設定することができます。

```
<seclabel type='dynamic' model='selinux' relabel='yes'>
  <baselabel>system_u:system_r:svirt_custom_t:s0</baselabel>
  <label>system_u:system_r:svirt_custom_t:s0:c87,c520</label>
  <imagelabel>system_u:object_r:svirt_image_t:s0:c87,c520</imagelabel>
</seclabel>
```

4.4.4. 動的リソースラベルを使用した静的設定

一部のアプリケーションは、セキュリティーラベルの生成を完全に制御する必要がありますが、リソースのラベル付けは依然として **libvirt** が行う必要があります。以下のゲスト XML 設定は、動的リソースラベルを使用した静的設定の例を示しています。

```
<seclabel type='static' model='selinux' relabel='yes'>
  <label>system_u:system_r:svirt_custom_t:s0:c87,c520</label>
</seclabel>
```

4.4.5. リソースラベルを使用しない静的設定

MLS (マルチレベルセキュリティー) または厳重に管理された環境で主に使用される、リソース再ラベルを使用しない静的設定が可能です。静的ラベルにより管理者は、仮想マシン用に **MCS/MLS** フィールドなどの特定のラベルを選択することができます。静的なラベルが付いた仮想マシンを実行する管理者は、イメージファイルに正しいラベルを設定する責任を担います。仮想マシンは常にそのラベルで起動し、**sVirt** システムは静的なラベルの付いた仮想マシンのコンテンツは決して変更しません。以下のゲスト XML 設定は、このシナリオの例を示しています。

```
<seclabel type='static' model='selinux' relabel='no'>
  <label>system_u:system_r:svirt_custom_t:s0:c87,c520</label>
</seclabel>
```

第5章 仮想化環境におけるネットワークセキュリティ

5.1. ネットワークセキュリティの概要

大半の状況では、ネットワークはシステム、アプリケーション、管理インターフェースへの唯一のアクセス方法です。ネットワークは、仮想化システムおよびそれらのシステムでホストされているアプリケーションの可用性の管理において極めて重要な役割を果たすので、仮想化システムとデータをやり取りするネットワークチャネルをセキュアな状態に確保することは非常に重要です。

ネットワークのセキュリティ保護により、管理者は機密データのアクセスを制御して、情報の漏えいや改ざんから保護することができます。

5.2. ネットワークセキュリティ推奨プラクティス

ネットワークセキュリティはセキュアな仮想化インフラストラクチャーの重要な要素です。ネットワークのセキュリティ保護については、以下の推奨プラクティスを参照してください。

- システムのリモート管理はすべてセキュアなネットワークチャネル上のみで実行されるようにしてください。SSHのようなツールや、TLS または SSL などのネットワークプロトコルは認証とデータ暗号化の両方を提供し、システムへのセキュアなアクセスとその制御を行います。
- ゲストアプリケーションによる機密データの転送はセキュアなネットワークチャネルで行われるようにします。TLS や SSL などのプロトコルが利用できない場合には、IPsec などを使用することを検討してください。
- ファイアウォールを設定して、ブート時にアクティブ化されるようにします。システムの使用と管理に必要なネットワークポートのみを許可してください。ファイアウォールルールは、定期的にテストと見直しを行ってください。

5.2.1. SPICE への接続のセキュリティ保護

SPICE リモートデスクトッププロトコルは SSL/TLS をサポートしています。これは、SPICE のすべての通信チャネル (main、display、inputs、cursor、playback、record) で有効化する必要があります。

5.2.2. ストレージへの接続のセキュリティ保護

仮想化システムのネットワークストレージへの接続は、さまざまな方法で行うことができます。各アプローチにはセキュリティ上のさまざまな利点と懸念点がありますが、セキュリティ上の同一の原則がそれぞれに適用されます。使用前にはリモートのストアプールを認証し、転送中のデータの機密性と整合性を保護します。

データは保管時にもセキュアな状態を維持する必要があります。Red Hat では、データを保管する前に暗号化またはデジタル署名すること、もしくはこの両方を推奨しています。



注記

ネットワークストレージの詳細については、『[Red Hat Enterprise Linux 仮想化の導入および管理ガイド](#)』の「ストレージプール」の章を参照してください。

付録A 追加情報

A.1. SELINUX および SVIRT

SELinux および sVirt に関する詳細情報:

- SELinux のメイン Web サイト: <https://www.nsa.gov/what-we-do/research/selinux/documentation/assets/files/presentations/2004-ottawa-linux-symposium-bof-presentation.pdf>
- SELinux のドキュメント: <https://www.nsa.gov/what-we-do/research/selinux/documentation/index.shtml>
- sVirt のメイン Web サイト: <http://selinuxproject.org/page/SVirt>
- Dan Walsh 氏のブログ: <http://danwalsh.livejournal.com/>
- 非公式の SELinux FAQ: <http://www.crypt.gen.nz/selinux/faq.html>

A.2. 仮想化セキュリティー

仮想化セキュリティーに関する追加情報

- NIST (National Institute of Standards and Technology) 完全仮想化セキュリティーガイドライン: <http://www.nist.gov/itl/csd/virtual-020111.cfm>

付録B 改訂履歴

改訂 1.0-18.2 翻訳ファイルを XML ソースバージョン 1.0-18 と同期	Tue Feb 27 2018	Terry Chuang
改訂 1.0-18.1 翻訳ファイルを XML ソースバージョン 1.0-18 と同期	Sun Sep 24 2017	Terry Chuang
改訂 1.0-18 7.4 GA 公開用バージョン	Thu Jul 27 2017	Jiri Herrmann
改訂 1.0-15 7.3 GA 公開用バージョン	Mon Oct 17 2016	Jiri Herrmann
改訂 1.0-9 改訂履歴の整理	Thu Oct 08 2015	Jiri Herrmann
改訂 1.0-8 7.1 GA リリース向けのバージョン	Wed Feb 18 2015	Scott Radvan