Red Hat Enterprise Linux 7

仮想化セキュリティーガイド

RHEL の仮想化環境におけるホスト、ゲスト、共有インフラストラクチャーのセキュリティー保護

Last Updated: 2023-03-25

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

RHEL の仮想化環境におけるホスト、ゲスト、共有インフラストラクチャーのセキュリティー保護

Enter your first name here. Enter your surname here.
Enter your organisation's name here. Enter your organisational division here.
Enter your email address here.

法律上の通知

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ーが重要である理由 4

第2章 ホストのセキュリティー 5
2.1. ホスト物理マシンの保護 5
2.2. クライアントアクセス制御 6
2.2.1. アクセス制御ドライバー 6
2.2.2. オブジェクトおよびアクセス権 7
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ー上の懸念事項 7
2.3. 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 7

第3章 ゲストのセキュリティー 9
3.1. ゲストのセキュリティーが重要である理由 9
3.2. ゲストセキュリティーの推奨プラクティス 9
3.3. KERNEL ADRESS SPACE LAYOUT RANDOMIZATION 9
3.4. VIRT-MANAGER を使用した SECUREBOOT RED HAT ENTERPRISE LINUX 7 ゲストの作成 10

第4章 SVIRT 13
4.1. はじめに 13
4.2. SELINUX と強制アクセス制御 (MAC) 13
4.3. SVIRT の設定 14
4.4. SVIRT のラベル付け 15
4.4.1. sVirt ラベルのタイプ 15
4.4.2. 動的設定 16
4.4.3. ベースラベルを使用した動的設定 17
4.4.4. 動的リソースラベルを使用した静的設定 17
4.4.5. リソースラベルを使用しない静的設定 17
4.4.6. sVirt のラベル付けおよび NFS 17

第5章 仮想化環境におけるネットワークセキュリティー 19
5.1. ネットワークセキュリティーの概要 19
5.2. ネットワークセキュリティー推奨プラクティス 19
5.2.1. SPICE への接続のセキュリティー保護 19
5.2.2. ストレージへの接続のセキュリティー保護 19

付録A 追加情報 20
A.1. SELINUX および SVIRT 20
A.2. 仮想化セキュリティー 20

付録B 改訂履歴 21

第1章 はじめに

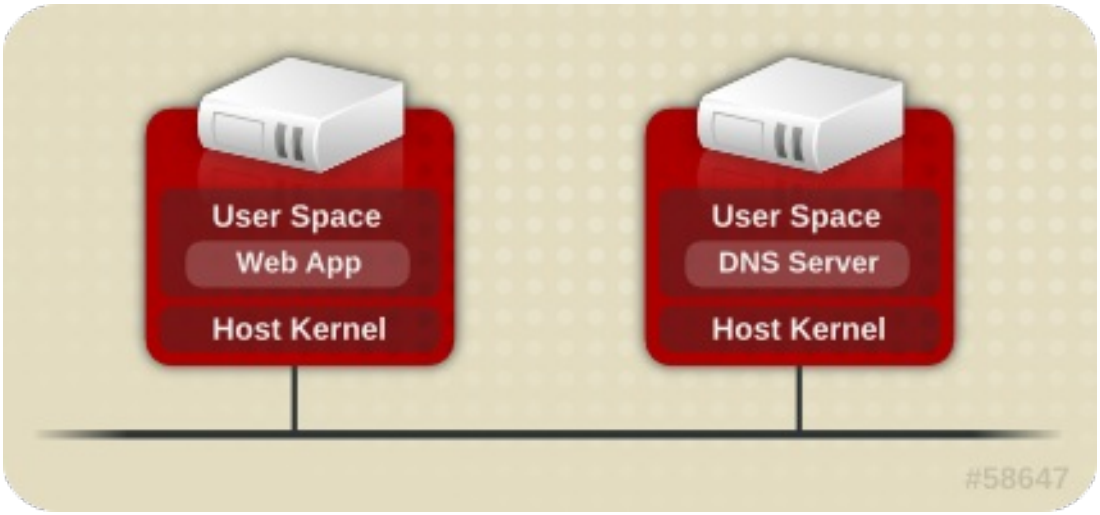
1.1. 仮想化環境と非仮想化環境

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エクス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設定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、ネットワークに直接サービスを提供します。

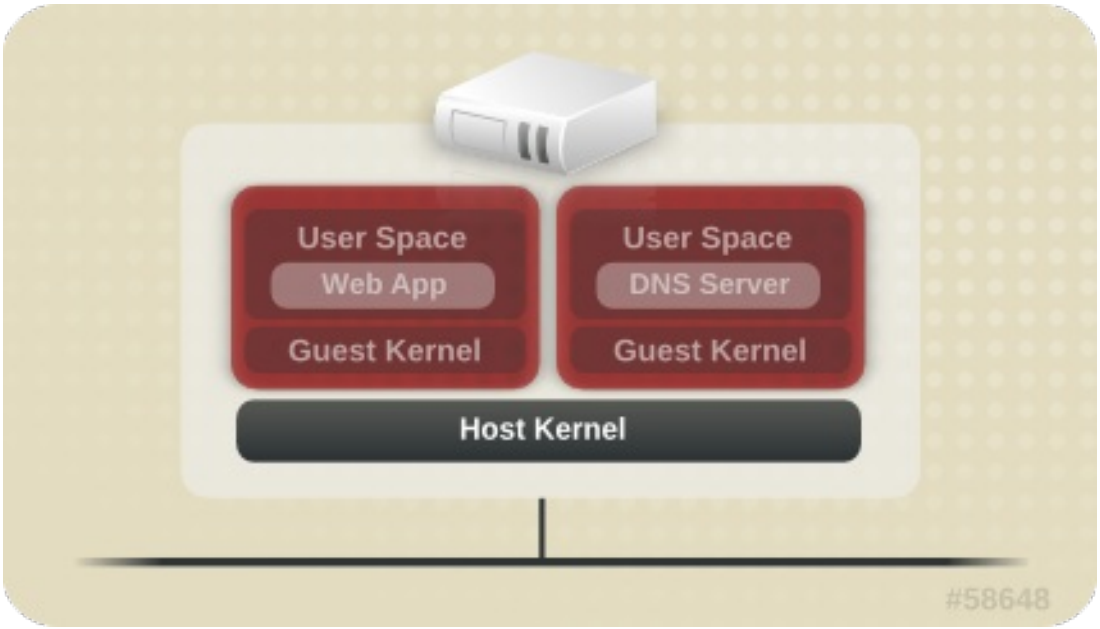
図1.1 非仮想化環境



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 (ゲストの仮想マシンとして) 単一のホストカーネルおよび物理ホストに格納できます。

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、ネットワーク攻撃を除いて、エクスプロイトは通常、影響を受けるマシンに抑えられます。仮想化環境でサービスをグループ化すると、システムに追加の脆弱性が発生します。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のあるセキュリティー上の欠陥がハイパーバイザーに存在する場合、このゲストはホストだけでなく、そのホストで実行されている他のゲストも攻撃できる可能性があります。

1.2. 仮想化セキュリティーが重要である理由

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。仮想化リソースとサービスは、以下のセキュリティーを考慮してデプロイする必要があります。

- ホストおよび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、モニターリング、メンテナー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。
- 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ー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。

第2章 ホストのセキュリティー

Red Hat Enterprise Linux システムは、仮想化テクノロジーをデプロイする際、ホストは、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、全仮想化ゲストを管理および制御します。ホストシステムが危険にさらされると、ゲストとそのデータも脆弱になります。

Red Hat Enterprise Linux ホストシステムのセキュリティー保護は、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。

2.1. ホスト物理マシンの保護

以下のタスクおよびヒントは、Red Hat Enterprise Linux ホストの信頼性を確保し、パフォーマンスを高めるのに役立ちます。

- SELinux がインストール用に適切に設定されており、Enforcing モードで動作していることを確認します。

  ```
  # setenforce 1
  ```

  これは、適正なセキュリティーのプラクティスであることに加えて、sVirt が提供す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 および sVirt の詳細は、*4章sVirt* を参照してください。

- すべての不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。不要なユーザーアカウントは削除してください。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アクセスを付与することを検討してください。

- 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。また、脆弱なアプリケーションは、ホストの攻撃のための進路となります。

- 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/** に格納します。仮想マシンのイメージをこれ以外のディレクトリーに格納する場合は、そのディレクトリーを 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。

- 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てください。

- ホストシステムで 監査が有効になり、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。監査が有効になると、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、ゲストの状態を追跡する上で役に立ちます。さらに、libvirt 監査イベントは、特殊な **auvirt** ユーティリティーを使用して表示することもできます。詳細については、**man auvirt** コマンドを使用してください。

- 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でのみ実行されるようにし

てください。SSH のようなユーティリティーや、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します。

- 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、システムの起動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。

- 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。

- 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、悪意のあるコードを使用して攻撃者が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。

  該当する SR-IOV 仮想機能デバイスの割り当てを使用することが推奨されます。

注記

ホストシステムのセキュリティーのヒントおよび使用方法は、『Red Hat Enterprise Linux セキュリティーガイド』 を参照してください。

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により、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設定できます。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できます。

デフォルトの設定では、**libvirtd** デーモンには、アクセス制御のレベルが 3 つあります。

1. すべての接続は認証されていない状態で開始します。ここで許可される唯一の API 操作は、認証を完了するために必要なものです。

2. 認証が成功すると、接続は、クライアント接続の発信元ソケットに応じて、すべての libvirt API 呼び出しへの完全な無制限のアクセスを持つか、"読み取り専用" 操作のみに制限されます。

3. アクセス制御フレームワークを使用すると、認証された接続に、管理者が定義するきめ細かいアクセス許可ルールを設定できます。

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アクセス権があります。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アクセス権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ーリングします。

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、**none** ドライバーが使用されます。これは、アクセス制御チェックをまったく実行しません。現在、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 アクセスドライバーの使用方法は、設定に関するドキュメント を参照してください。

アクセスドライバーは、***access_drivers*** パラメーターを使用して **/etc/libvirt/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。複数のアクセスドライバーが要求される場合は、アクセスが付与されるためにすべてが正常に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、**augtool** 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 への変更を適用するには、**libvirtd** サービスを再起動します。

```
# systemctl restart libvirtd.service
```

2.2.2. オブジェクトおよびアクセス権

libvirt は、API のすべての主要なオブジェクトタイプにアクセス制御を適用します。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。

2.2.3. 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ー上の懸念事項

- ホストの物理マシンは、ファイルシステムラベルを使用して、**fstab** ファイル、**initrd** ファイル、またはカーネルコマンドラインでファイルシステムを特定しないでください。特定すると、ゲストの仮想マシンにすべてのパーティションまたは LVM ボリュームに対する書き込みアクセスがある場合のセキュリティーリスクが発生します。これは、ゲストの仮想マシンが、ホストの物理マシンに属するファイルシステムのラベルを潜在的に自身のブロックデバイスストレージに書き込むことができるからです。ホストの物理マシンの再起動時に、ホストの物理マシンが、このゲストの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、ホストの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。

  デバイスの UUID を使用して、**/etc/fstab** ファイル、**/dev/initrd** ファイル、またはカーネルコマンドラインでデバイスを識別することをお勧めします。

- ゲスト仮想マシンには、ディスク全域、またはブロックデバイス全域 (例: **/dev/sdb**) への書き込みアクセス権を付与しないでください。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、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 管理および設定の例は 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。

  たとえば、**/dev/sdb1** や **/dev/sdb** などの RAW ディスクのパーティションへの RAW アクセスを使用する場合は、**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に 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は、論理ボリュームマネージャーの管理ガイド を参照してください。

2.3. 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プロバイダーは、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ー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関する要件により、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。

- ゲストからの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から分離し、顧客ネットワークを相互に分離します。これにより、

  - ゲストはネットワーク経由でホストシステムにアクセスできなくなります。

  - ある顧客は、クラウドプロバイダーの内部ネットワークを介して別の顧客のゲストシステムに直接アクセスできなくなります。

第3章 ゲストのセキュリティー

3.1. ゲストのセキュリティーが重要である理由

ホストシステムのセキュリティーは、ホスト上で実行されているゲストのセキュリティーを確保する上で重要ですが、個々のゲストマシンを適切にセキュリティー保護する必要性を排除するものではありません。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ー上のリスクはすべて依然として存在します。重要なビジネスデータや機密性の高い顧客情報など、ゲストシステムにアクセス可能なリソースは、ゲストシステムが危険にさらされる場合、脆弱になる可能性があります。

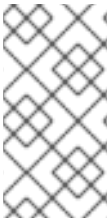
3.2. ゲストセキュリティーの推奨プラクティス

『Red Hat Enterprise Linux セキュリティーガイド』 に記載されている Red Hat Enterprise Linux システムを保護するための推奨プラクティスはすべて、仮想化ゲストとしてインストールされたシステムだけでなく、従来の非仮想化システムにも適用されます。ただし、仮想化環境でゲストを実行する場合には、重要なセキュリティー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ー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。これらのエージェントとアプリケーションが、SELinux などの標準の Red Hat Enterprise Linux セキュリティー機能を使用して保護されていることを確認してください。

- 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。**dm-crypt** や **GnuPG** などの暗号化ツールを使用して保存された機密データを保護します。ただし、暗号化キーの機密性を確保するために特別な注意を払う必要があります。



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 を無効にできます。KSM の詳細は、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。

3.3. KERNEL ADRESS SPACE LAYOUT RANDOMIZATION

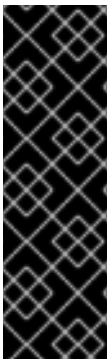
Red Hat Enterprise Linux 7.5 以降では、KVM ゲスト仮想マシンの Kernel Address Space Randomization (KASLR) 機能が含まれています。KASLR は、カーネルイメージを解凍する物理アドレスおよび仮想アドレスをランダム化できるため、カーネルオブジェクトの位置に基づいてゲストのセキュリティー攻撃を防ぎます。

KASLR はデフォルトで有効にされますが、ゲストのカーネルコマンドラインに **nokaslr** 文字列を追加することで、特定のゲストで無効にできます。ゲストの起動オプションを編集するには、次のコマンドを使用します。*guestname* はゲストの名前です。

```
# virt-edit -d guestname /etc/default/grub
```

その後、たとえば、**GRUB_CMDLINE_LINUX** 行を変更します。

```
GRUB_CMDLINE_LINUX="rd.lvm.lv=rhel/root rd.lvm.lv=rhel/swap rhgb quiet nokaslr"
```



重要

KASLR で有効になっているゲストから作成したゲストのダンプファイルは、**crash** ユーティリティーでは読み込みできません。これを修正するには、ゲストの XML 設定ファイルの **<features>** セクションに **<vmcoreinfo/>** 要素を追加します。

宛先のホストが **<vmcoreinfo/>** をサポートしない OS を使用している場合は、**<vmcoreinfo/>** を使用したゲストの 移行 に失敗する点に留意してください。これには、Red Hat Enterprise Linux 7.4 以前のバージョンと、Red Hat Enterprise Linux 6.9 以前のバージョンが該当します。

3.4. VIRT-MANAGER を使用した SECUREBOOT RED HAT ENTERPRISE LINUX 7 ゲストの作成

この手順では、ローカルに保存されているインストール用 DVD や DVD イメージを使って SecureBoot Red Hat Enterprise Linux 7 のゲスト仮想マシンを作成する方法を説明します。Red Hat Enterprise Linux 7 の DVD イメージは、Red Hat カスタマーポータル から入手できます。

SecureBoot 機能により、仮想マシンが暗号化で署名された OS を実行していることを確認できます。マルウェアが仮想マシンのゲスト OS を変更すると、SecureBoot により仮想マシンが起動しなくなり、ホストのマシンにマルウェアが広がらないようになります。

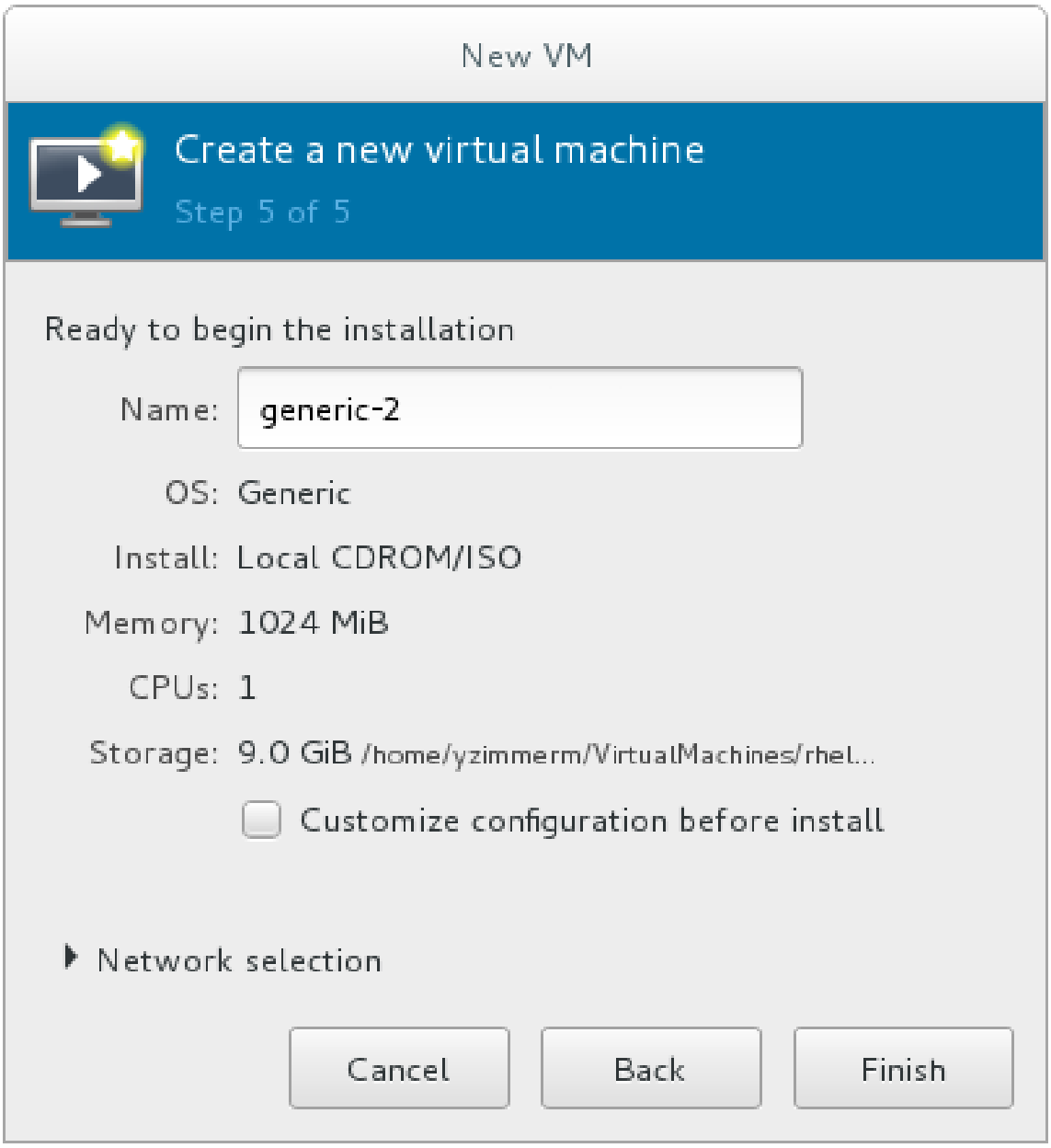
手順3.1 ローカルインストールメディアを使用する virt-manager を使用した SecureBoot Red Hat Enterprise Linux 7 ゲスト仮想マシンの作成

1. virt-manager を使用した Red Hat Enterprise Linux 7 ゲストの作成 の手順 1 から 6 を実行します。

2. **名前を付けて最終設定を行います。**
   仮想マシンに名前を付けます。仮想マシン名には、文字、数字、およびアンダースコア (_)、ピリオド (.)、およびハイフン (-) を含めることができます。仮想マシンを移行するには、仮想マシンの名前は一意でなければならず、数字のみの名前は使用できません。

   デフォルトで、仮想マシンは 'default' というネットワークの Network Address Translation (NAT) を使用して作成されます。ネットワークの選択を変更するには、**Network selection** をクリックしてホストデバイスとソースモードを選択します。

図3.1 設定の確認

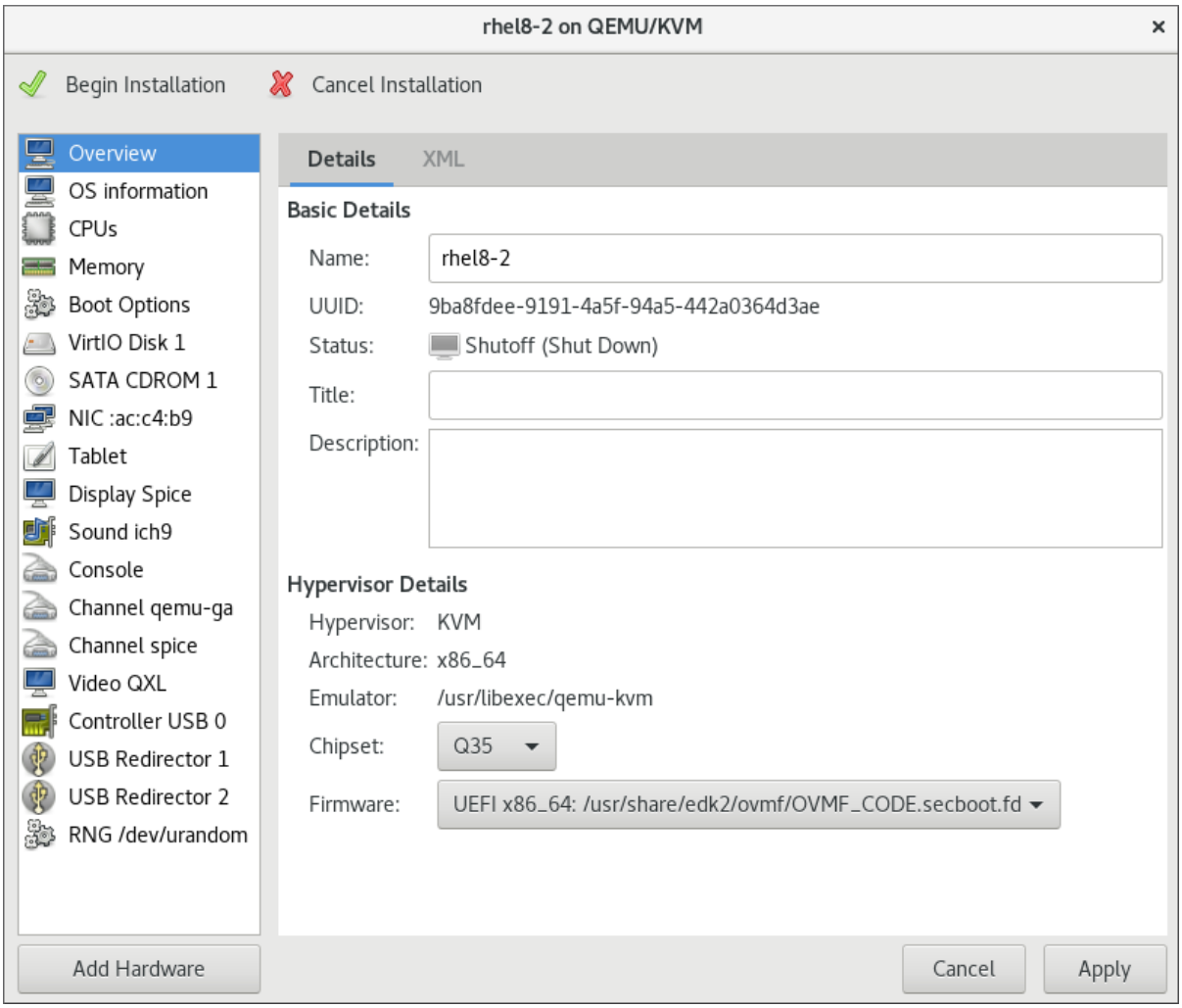


仮想マシンのハードウェアをさらに設定する場合は、**インストールの前に設定をカスタマイズする** のチェックボックスにチェックを入れ、ゲストのストレージまたはネットワークデバイスを変更するか、準仮想化 (virtio) ドライバーを使用するか、デバイスを追加します。仮想マシンの設定を確認し、問題がなければ **Finish** をクリックします。これにより、仮想マシンをさらに設定するための新規ウィザードが開きます。

3. **仮想マシンのハードウェアのカスタマイズ**
ウィザードの概要セクションで、**チップセット** ドロップダウンメニューから Q35 を選択します。

ファームウェア ドロップダウンメニューから UEFI x86_64 を選択します。

図3.2 ハードウェアの設定ウィンドウ



仮想マシンの設定を確認し、問題がなければ **適用** をクリックします。

インストールの開始 をクリックして、指定したネットワーク設定、仮想化タイプ、およびアーキテクチャーで仮想マシンを作成します。

これで、ISO インストールディスクイメージから、SecureBoot Red Hat Enterprise Linux 7 ゲスト仮想マシンが作成されました。

第4章 SVIRT

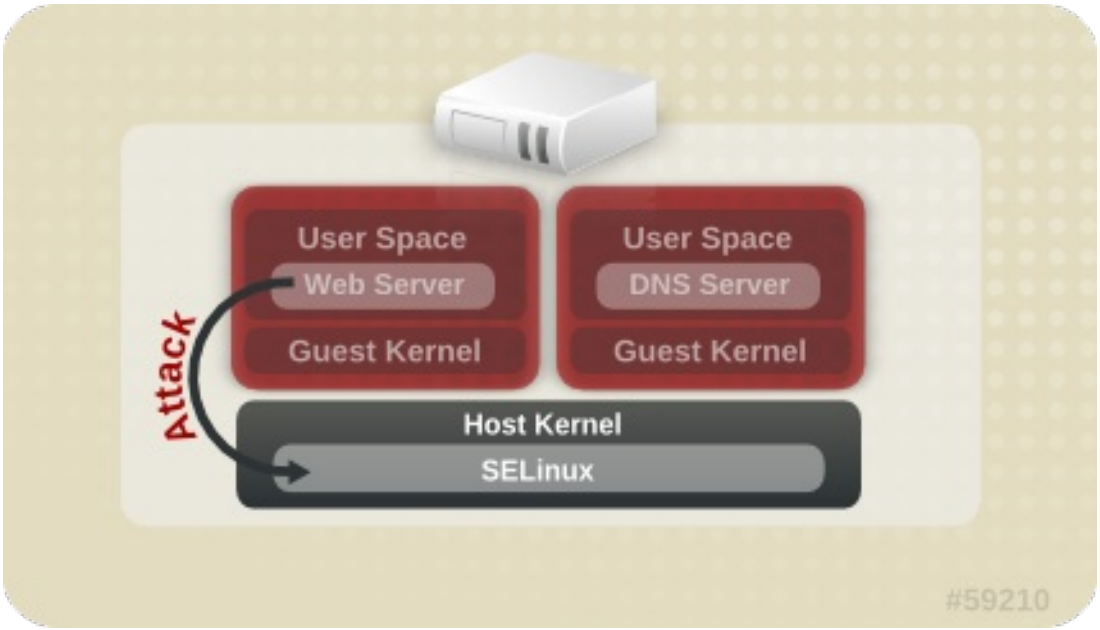
4.1. はじめに

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 セキュリティー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、Security-Enhanced Linux (SELinux) 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ーポリシーを通して、強制アクセス制御 (MAC)、マルチレベルセキュリティー (MLS)、およびマルチカテゴリーセキュリティー (MCS) を追加します。SELinux は、Linux カーネルで実行しているプロセス (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、粒度の細かいアクセス権を適用して仮想マシンをグループ化し、リソースを共有できます。

セキュリティーの観点からすると、ハイパーバイザーは攻撃者の格好の的です。これは、ハイパーバイザーがセキュリティー侵害を受けると、そのホストシステムで実行している仮想マシンのセキュリティーもすべて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ー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で、セキュリティー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げ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では、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 で起動された場合に 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	スタッフユーザーが sVirt ドメインを作成して移行できるようになります。
unprivuser_use_svirt	非特権ユーザーが sVirt ドメインを作成して移行できるようになります。
virt_sandbox_use_audit	サンドボックスコンテナーが監査メッセージを送信できるようになります。
virt_sandbox_use_netlink	サンドボックスコンテナーでネットリンクシステム呼び出しが使用できるようになります。
virt_sandbox_use_sys_admin	サンドボックスコンテナーで sys_admin システム呼び出し (mount 等) が使用できるようになります。
virt_transition_userdomain	仮想プロセスをユーザードメインとして実行できるようになります。
virt_use_comm	virt でシリアルおよびパラレルの通信ポートが使用できるようになります。
virt_use_execmem	制限された仮想ゲストが実行可能メモリーおよび実行可能スタックを使用できるようになります。
virt_use_fusefs	FUSE がマウントしたファイルを virt が読み取りできるようになります。

SELinux のブール値	説明
virt_use_nfs	NFS がマウントしたファイルを virt が管理できるようになります。
virt_use_rawip	virt で rawip ソケットとの通信ができるようになります。
virt_use_samba	CIFS がマウントしたファイルを virt が管理できるようになります。
virt_use_sanlock	制限された仮想ゲストが sanlock と相互作用できるようになります。
virt_use_usb	virt で USB デバイスが使用できるようになります。
virt_use_xserver	仮想マシンが X Window System と対話できるようにします。

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 のラベル付け

SELinux の保護下にある他のサービスと同様に、sVirt はプロセスベースのメカニズム、ラベル、制限を使用してセキュリティーを強化し、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、システムのリソースに自動的に適用 (動的) されますが、管理者が手動で指定 (静的) して、特別な要件がある場合でも対応することができます。

ゲストの sVirt ラベルを編集するには、**virsh edit *guest_name*** コマンドを使用し、以下のセクションで説明されているように **<seclabel>** 要素を追加または編集します。**<seclabel>** は、ゲスト全体の root 要素として使用できます。または、指定のデバイスの特定の sVirt ラベルを選択するために **<source>** 要素のサブ要素として指定することもできます。

<seclabel> 要素の総合的な情報は、libvirt のアップストリームのドキュメント を参照してください。

4.4.1. sVirt ラベルのタイプ

次の表は、仮想マシンプロセス、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さまざまな sVirt ラベルの概要を示しています。

表4.2 sVirt ラベル

タイプ	SELinux コンテキスト	説明/効果

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。
仮想マシンの共有読み取り/書き込みコンテンツ	system_u:object_r:svirt_image_t:s0	**svirt_t** プロセスはすべて、**svirt_image_t:s0** 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	system_u:object_r:svirt_content_t:s0	**svirt_t** 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	system_u:object_r:virt_content_t:s0	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。**svirt_t** 仮想プロセスは、このラベルの付いたファイル/デバイスの読み取りはできません。

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
```

この例の **qemu-kvm** プロセスには、**system_u:system_r:svirt_t:s0** というベースラベルがあります。libvirt システムが、このプロセス用に **c87,c520** という一意の MCS ラベルを生成しました。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、**c87,c520** という対応する MCS ラベル付き) 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

動的モードでデフォルトのベースセキュリティーラベルを上書きするには、以下の例のように、XML ゲスト設定で **<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を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>
```

4.4.6. sVirt のラベル付けおよび NFS

NFSv4.1 または NFSv4.2 ファイルシステムで sVirt ラベリングを使用するには、ゲスト共有用にエクスポートする NFS ディレクトリーの root の SELinux コンテキストを **virt_var_lib_t** へ変更する必要があります。たとえば、**/exports/nfs/** ディレクトリーをエクスポートする場合は、以下のコマンドを実行します。

```
# semanage fcontext -a -t virt_var_lib_t '/exports/nfs/'
# restorecon -Rv /exports/nfs/
```

さらに、**libvirt** が NFS ボリューム上のゲスト仮想マシンの sVirt ラベルを動的に生成する場合、単一の

ホスト内でのラベルの一意性のみを保証します。これは、複数ホスト間で多数のゲストが NFS ボリュームを共有する場合、重複したラベルが発生する可能性があり、それにより潜在的な脆弱性が作成される場合があることを意味します。

この問題を回避するには、次のいずれかを実行します。

- 各仮想化ホストに異なる NFS ボリュームを使用してください。さらに、ゲスト移行 を実行するときは、**--migrate-disks** および **--copy-storage-all** のオプションを使用してゲストストレージをコピーします。

- **virt-install** コマンドでゲストを新規作成した場合は、以下のコマンドでゲストの静的ラベルを設定します。

  - **--security** オプションの使用以下に例を示します。

    ```
    # virt-install --name guest1-rhel7 --memory 2048 --vcpus 2 --disk size=8 --cdrom
    /home/username/Downloads/rhel-workstation-7.4-x86_64-dvd.iso --os-variant rhel7 --
    security model=selinux,label='system_u:object_r:svirt_image_t:s0:c100,c200'
    ```

    これは、ゲスト上のすべてのディスクのセキュリティーラベルを設定します。

  - ***seclabel*** パラメーターを指定して **--disk** オプションを使用します。以下に例を示します。

    ```
    # virt-install --name guest1-rhel7 --memory 2048 --vcpus 2 --disk
    /path/to/disk.img,seclabel.model=selinux,seclabel.label='system_u:object_r:svirt_image_t:s0
    c100,c200' --cdrom /home/username/Downloads/rhel-workstation-7.4-x86_64-dvd.iso --
    os-variant rhel7
    ```

    これは、指定したディスクに対してのみ、セキュリティーラベルを設定します。

第5章 仮想化環境におけるネットワークセキュリティー

5.1. ネットワークセキュリティーの概要

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、管理インターフェイ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ロールを果たすので、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ー保護により、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ー推奨プラクティス

ネットワークセキュリティー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ー保護については、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、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。

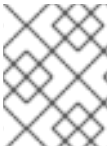
5.2.1. SPICE への接続のセキュリティー保護

SPICE リモートデスクトッププロトコルは、すべての SPICE 通信チャネル (メイン、ディスプレイ、入力、カーソル、再生、レコード) に対して有効にする必要がある SSL/TLS をサポートします。

5.2.2. ストレージへの接続のセキュリティー保護

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ー上のさまざまな利点と懸念点がありますが、同じセキュリティー原則がそれぞれに適用されます。使用前にリモート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使用のセクション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 に関する追加情報:

- 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 ドキュメント: https://www.nsa.gov/what-we-do/research/selinux/documentation/index.shtml.
- sVirt の主要 Web サイト: http://selinuxproject.org/page/SVirt.
- Dan Walsh のブログ: http://danwalsh.livejournal.com/

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) の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 1.0-22 7.7 ベータ版公開用バージョン	Thu May 23 2019	Jiri Herrmann
改訂 1.0-21 7.6 GA リリースのバージョン	Thu Oct 25 2018	Jiri Herrmann
改訂 1.0-21 7.6 ベータ版公開用バージョン	Thu Aug 14 2018	Jiri Herrmann
改訂 1.0-20 7.5 GA 公開用バージョン	Thu Apr 5 2018	Jiri Herrmann
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 リリース向けバージョン	Wed Feb 18 2015	Scott Radvan