

セキュリティ構成ガイド

2018 年 5 月 3 日

vRealize Automation 7.4



vmware®

最新の技術ドキュメントは VMware の Web サイト (<https://docs.vmware.com/jp/>) にあります
このドキュメントに関するご意見および感想がある場合は、docfeedback@vmware.com までお送りください。

VMware, Inc.
3401 Hillview Ave.
Palo Alto, CA 94304
www.vmware.com

ヴィエムウェア株式会社
105-0013 東京都港区浜松町 1-30-5
浜松町スクエア 13F
www.vmware.com/jp

Copyright © 2015–2018 VMware, Inc. 無断転載を禁ず。 [著作権および商標情報](#)。

目次

- 1 セキュアな構成 5
- 2 vRealize Automation のセキュアなベースラインの概要 6
- 3 インストール メディアの整合性の確認 7
- 4 VMware システム ソフトウェア インフラストラクチャのセキュリティ強化 8
 - VMware vSphere 環境のセキュリティ強化 8
 - Infrastructure as a Service (IaaS) ホストのセキュリティ強化 8
 - Microsoft SQL Server のセキュリティ強化 9
 - Microsoft .NET のセキュリティ強化 9
 - Microsoft Internet Information Services (IIS) のセキュリティ強化 9
- 5 インストールされたソフトウェアの確認 11
- 6 VMware セキュリティ アドバイザリおよびパッチ 12
- 7 セキュアな構成 13
 - vRealize Automation アプライアンスのセキュリティ強化 13
 - root パスワードの変更 13
 - root パスワードのハッシュと複雑性の確認 14
 - root パスワード履歴の確認 14
 - パスワード有効期限の管理 15
 - セキュア シェルと管理者アカウントの管理 16
 - 仮想アプライアンス管理インターフェイスのユーザーの変更 20
 - ブートローダー認証の設定 21
 - NTP の設定 21
 - 転送中の vRealize Automation アプライアンス データの TLS の構成 22
 - 保存データのセキュリティの確認 30
 - vRealize Automation アプリケーション リソースの構成 31
 - コンソール プロキシ構成のカスタマイズ 34
 - サーバ応答ヘッダーの構成 36
 - vRealize Automation アプライアンス セッション タイムアウトの設定 37
 - 必須でないソフトウェアの管理 38
 - Infrastructure as a Service (IaaS) コンポーネントのセキュリティ保護 42
 - Windows Time サービスを無効にする 42
 - 転送中の Infrastructure as a Service (IaaS) データの TLS の構成 42
 - TLS 暗号スイートの構成 44

ホスト サーバのセキュリティの確認	44
アプリケーション リソースの保護	45
Infrastructure as a Service (IaaS) ホスト マシンのセキュリティ保護	46

8 ホストのネットワーク セキュリティの構成 47

VMware アプライアンスのネットワーク設定の構成	47
ネットワーク インターフェイスのユーザーによるコントロールの制限	47
TCP バックログのキュー サイズの設定	48
ブロードキャスト アドレスへの ICMPv4 エコーの拒否	48
IPv4 プロキシ ARP を無効にする	48
IPv4 ICMP リダイレクト メッセージの拒否	49
IPv6 ICMP リダイレクト メッセージの拒否	50
IPv4 Martian パケットのログ	50
IPv4 リバース パス フィルタリングの使用	51
IPv4 転送の拒否	52
IPv6 転送の拒否	53
IPv4 TCP Syncookie の使用	53
IPv6 ルーター通知の拒否	54
IPv6 ルーター要請の拒否	55
ルーター要請の IPv6 ルーター プリファレンスの拒否	55
IPv6 ルーター プリフィックスの拒否	56
IPv6 ルーター通知のホップ制限設定の拒否	57
IPv6 ルーター通知 Autoconf 設定の拒否	57
IPv6 近隣要請の拒否	58
IPv6 最大アドレス数の制限	59
Infrastructure as a Service (IaaS) ホストのネットワーク設定の構成	60
ポートおよびプロトコルの構成	60
必須のユーザー ポート	60
管理者に必要なポート	60

9 監査とログ 63

セキュアな構成

セキュアな構成では、ユーザーによる vRealize Automation 環境のセキュアな構成、評価、および最適化に役立つ情報を提供します。

ここでは、一般的な vRealize Automation 環境におけるセキュアなデプロイの検証および構成を記述し、ユーザーが情報に基づいて、適切なセキュリティ設定を選択できるよう解説し、その手順について説明します。

対象者

この情報は、vRealize Automation システム管理者と、システム セキュリティのメンテナンスと構成を担当する他のユーザーを対象としています。

VMware の技術ドキュメントの用語集

VMware の技術ドキュメントには、新しい用語などをまとめた用語集があります。当社の技術ドキュメントで 사용되는用語の定義については、<http://www.vmware.com/support/pubs> をご覧ください。

vRealize Automation のセキュアな ベースラインの概要

2

VMware は、vRealize Automation システムのセキュアなベースラインを検証および構成できる包括的な推奨事項を提供します。

VMware が指定した適切なツールおよび手順を使用して、vRealize Automation システムを検証し、セキュリティ強化されたセキュアなベースライン構成を維持します。一部の vRealize Automation コンポーネントのセキュリティはすべて強化された状態または一部強化された状態でインストールされますが、VMware のセキュリティ推奨事項、自社のセキュリティ ポリシー、および既知の脅威を考慮して、各コンポーネントの構成を確認および検証する必要があります。

vRealize Automation のセキュリティ状態

vRealize Automation のセキュリティ状態は、システムとネットワークの構成、組織のセキュリティ ポリシー、およびセキュリティのベスト プラクティスに基づいた包括的にセキュアな環境を想定しています。

vRealize Automation システムのセキュリティ強化を検証および構成する場合、VMware のセキュリティ強化に関する推奨事項に記載されている次の領域について考慮します。

- 安全な展開
- セキュアな構成
- ネットワーク セキュリティ

使用するシステムがセキュリティ強化されていることを確認するには、VMware の推奨事項と各地域のセキュリティ ポリシーがそれぞれの概念的な領域に関連していることについて考慮します。

システム コンポーネント

vRealize Automation システムのセキュリティ強化と安全な構成を考慮する場合、すべてのコンポーネントと、システムの機能をサポートするその仕組みを理解します。

安全なシステムを計画および実装する場合、次のコンポーネントを考慮します。

- vRealize Automation アプライアンス
- IaaS コンポーネント

vRealize Automation とコンポーネントの動作を理解するには、VMware vRealize Automation ドキュメント センターで『基盤と概念』を参照してください。vRealize Automation の標準展開とアーキテクチャの詳細については、リファレンス アーキテクチャを参照してください。

インストールメディアの整合性の確認

ユーザーは、VMware 製品をインストールする前にインストールメディアの整合性を常に確認する必要があります。

ダウンロードしたファイルの整合性と信頼性を確認するために、ISO、オフラインバンドル、またはパッチをダウンロードしたら、必ず SHA1 ハッシュを確認します。VMware から入手した物理メディアのセキュリティシールが破損している場合は、そのソフトウェアを VMware に返却して交換してください。

メディアをダウンロードしたら、MD5/SHA1 サムの値を使用して、ダウンロードの整合性を確認します。MD5/SHA1 のハッシュ出力と VMware Web サイトに表示されている値を比較します。SHA1 または MD5 ハッシュと一致する必要があります。

インストールメディアの整合性確認の詳細については、<http://kb.vmware.com/kb/1537> を参照してください。

VMware システム ソフトウェア インフラストラクチャのセキュリティ強化

セキュリティ強化プロセスの一環として、VMware システムをサポートしているデプロイ済みソフトウェア インフラストラクチャを評価し、VMware セキュリティ強化ガイドラインを満たしていることを確認します。

VMware システムのセキュリティを強化する前に、サポートするソフトウェア インフラストラクチャ内のセキュリティ上の欠陥を確認して対応し、完全にセキュリティ強化されたセキュアな環境を作成します。考慮すべきソフトウェア インフラストラクチャの要素には、オペレーティングシステム コンポーネント、サポートするソフトウェア、およびデータベース ソフトウェアが含まれます。これらの要素や、その他のコンポーネントにおけるセキュリティ上の懸念には、メーカーの推奨およびその他の関連するセキュリティ プロトコルに従って対応します。

この章では次のトピックについて説明します。

- VMware vSphere[®] 環境のセキュリティ強化
- Infrastructure as a Service (IaaS) ホストのセキュリティ強化
- Microsoft SQL Server のセキュリティ強化
- Microsoft .NET のセキュリティ強化
- Microsoft Internet Information Services (IIS) のセキュリティ強化

VMware vSphere[®] 環境のセキュリティ強化

VMware vSphere[®] 環境を評価し、適切なレベルの vSphere セキュリティ強化ガイダンスが守られ、保守されていることを確認します。

セキュリティ強化に関するガイダンスの詳細については、

<http://www.vmware.com/security/hardening-guides.html>（英語）を参照してください。

包括的にセキュリティ強化された環境の一部として、VMware vSphere[®] インフラストラクチャが VMware によって定められたセキュリティ ガイドラインを満たしている必要があります。

Infrastructure as a Service (IaaS) ホストのセキュリティ強化

IaaS Microsoft Windows ホスト マシンが VMware ガイドラインに従ってセキュリティ強化されていることを確認します。

該当する Microsoft Windows セキュリティ強化およびセキュリティのベスト プラクティスのガイドラインで推奨事項を確認し、Windows Server ホストが確実にかつ適切にセキュリティ強化されるようにします。セキュリティ強化の推奨事項に従わない場合、Windows リリース上の安全でないコンポーネントから既知のセキュリティ脆弱性が狙われる可能性があります。

お使いのバージョンがサポートされていることを確認するには、[vRealize Automation のサポート マトリックス](#)（英語）を参照してください。

Microsoft 製品のセキュリティ強化プラクティスに関する適切なガイダンスについては、Microsoft のベンダーにお問い合わせください。

Microsoft SQL Server のセキュリティ強化

Microsoft SQL Server データベースが Microsoft および VMware によって定められたセキュリティ ガイドラインを満たしていることを確認します。

該当する Microsoft SQL Server セキュリティ強化およびセキュリティのベスト プラクティスのガイドラインで推奨事項を確認します。Microsoft SQL Server のインストールされたバージョンに関するすべての Microsoft セキュリティ通知を確認します。セキュリティ強化の推奨事項に従わない場合、Microsoft SQL Server バージョン上の安全でないコンポーネントから既知のセキュリティ脆弱性が狙われる可能性があります。

お使いのバージョンの Microsoft SQL Server がサポートされていることを確認するには、[vRealize Automation のサポート マトリックス](#)（英語）を参照してください。

Microsoft 製品のセキュリティ強化プラクティスに関するガイダンスについては、Microsoft のベンダーにお問い合わせください。

Microsoft .NET のセキュリティ強化

包括的にセキュリティ強化された環境の一部として、Microsoft .NET が Microsoft および VMware によって定められたセキュリティ ガイドラインを満たしている必要があります。

該当する Microsoft .NET セキュリティ強化およびセキュリティのベスト プラクティスのガイドラインで規定された推奨事項を確認します。また、使用している Microsoft SQL Server のバージョンに関するすべての Microsoft セキュリティ通知を確認します。セキュリティ強化の推奨事項に従わない場合、安全でない Microsoft.NET コンポーネントから既知のセキュリティの脆弱性が狙われる可能性があります。

お使いのバージョンの Microsoft.NET がサポートされていることを確認するには、[vRealize Automation のサポート マトリックス](#)（英語）を参照してください。

Microsoft 製品のセキュリティ強化プラクティスに関するガイダンスについては、Microsoft のベンダーにお問い合わせください。

Microsoft Internet Information Services (IIS) のセキュリティ強化

Microsoft Internet Information Services (IIS) が Microsoft および VMware のセキュリティ ガイドラインをすべて満たしていることを確認します。

該当する Microsoft IIS セキュリティ強化およびセキュリティのベスト プラクティスのガイドラインで規定された推奨事項を確認します。また、使用している IIS のバージョンに関するすべての Microsoft セキュリティ通知を確認します。セキュリティ強化の推奨事項に従わない場合、既知のセキュリティの脆弱性が狙われる可能性があります。

お使いのバージョンがサポートされていることを確認するには、[vRealize Automation のサポート マトリックス](#)（英語）を参照してください。

Microsoft 製品のセキュリティ強化プラクティスに関するガイダンスについては、Microsoft のベンダーにお問い合わせください。

インストールされたソフトウェアの 確認

5

サードパーティ製や使用されていないソフトウェアの脆弱性は未認証のシステム アクセスや可用性の中断のリスクを高めるため、VMware ホスト マシンにインストールされているすべてのソフトウェアを確認し、その使用状況を評価することが重要です。

VMware ホスト マシンには、システムの安全な運用のために必要なソフトウェア以外をインストールしないでください。使用されていない、または関係ないソフトウェアはアンインストールします。

サポート対象外のソフトウェアがインストールされているインベントリ

インストール済み製品の VMware デプロイとインベントリを評価し、関係のないサポート対象外のソフトウェアがインストールされていないことを確認します。

サードパーティ製品に対するサポート ポリシーの詳細については、VMware サポート記事 (<https://www.vmware.com/support/policies/thirdparty.html>) (英語) を参照してください。

サードパーティ製ソフトウェアの確認

VMware は、テストおよび検証が行われていないサードパーティ製ソフトウェアのインストールをサポートせず、推奨しません。VMware ホスト マシンに、セキュアでない、パッチが適用されていない、または認証されていないサードパーティ製ソフトウェアがインストールされている場合、システムが不正アクセスや可用性の中断のリスクにさらされる可能性があります。サポートされていないサードパーティ製ソフトウェアを使用する必要がある場合は、セキュアな構成およびパッチ適用の要件についてサードパーティ ベンダーにお問い合わせください。

VMware セキュリティ アドバイザリおよびパッチ

6

システムで最高レベルのセキュリティを維持するために、VMware セキュリティ アドバイザリに従って、関連するすべてのパッチを適用します。

VMware は製品のセキュリティ アドバイザリを公開しています。製品を既知の脅威から保護するには、このアドバイザリを注視します。

vRealize Automation のインストール、パッチ、アップグレード履歴を評価し、公開された VMware セキュリティ アドバイザリが順守および適用されていることを確認します。

現在の VMware セキュリティ アドバイザリの詳細については、
<http://www.vmware.com/jp/security/advisories.html> を参照してください。

セキュアな構成

システム構成に応じて、vRealize Automation 仮想アプライアンスと Infrastructure as a Service (IaaS) コンポーネントのセキュリティ設定を確認および更新します。また、他のコンポーネントとアプリケーションの構成を確認および更新します。

vRealize Automation インストールをセキュアに構成するには、コンポーネントが連携するように、各コンポーネントの構成に個別に対応します。十分に安全なベースラインを実現できるようにすべてのシステム コンポーネントの構成を検討します。

この章では次のトピックについて説明します。

- [vRealize Automation アプライアンスのセキュリティ強化](#)
- [Infrastructure as a Service \(IaaS\) コンポーネントのセキュリティ保護](#)

vRealize Automation アプライアンスのセキュリティ強化

システム構成の必要に応じて、vRealize Automation アプライアンスのセキュリティ設定を確認して更新します。

仮想アプライアンスと、そのホスト オペレーティング システムのセキュリティ設定を構成します。また、その他の関連コンポーネントとアプリケーションの構成も設定または確認します。適切な構成を実現するために、既存の設定の確認が必要な場合や、設定の変更や追加が必要な場合があります。

root パスワードの変更

該当するセキュリティ要件を満たすために、vRealize Automation アプライアンス の root パスワードを変更することができます。

仮想アプライアンス管理インターフェイスを使用して、vRealize Automation アプライアンスで root パスワードを変更します。root パスワードの複雑さが、組織のパスワード要件を満たしていることを確認します。

手順

- 1 vRealize Automation アプライアンス の仮想アプライアンス管理インターフェイスを開きます。
`https://<vRealizeAppliance-url>:5480`
- 2 仮想アプライアンス管理インターフェイスの [管理] タブを選択します。
- 3 [管理] サブメニューを選択します。
- 4 [現在の管理者パスワード] テキスト ボックスに、現在のパスワードを入力します。

- 5 [新しい管理者パスワード] テキスト ボックスに新しいパスワードを入力します。
- 6 [新しい管理者パスワードを再入力] テキスト ボックスに新しいパスワードを入力します。
- 7 [設定の保存] をクリックして、変更を保存します。

root パスワードのハッシュと複雑性の確認

root パスワードの複雑さが、組織のパスワード要件を満たしていることを確認します。

root ユーザーがユーザー アカウントに適用される pam_cracklib モジュール パスワードの複雑性チェックをバイパスするため、root パスワードの複雑性を検証する必要があります。

アカウント パスワードの先頭には **\$6\$** を指定する必要があります。これは sha512 ハッシュを表します。これは、セキュリティ強化されたすべてのアプライアンスの標準ハッシュです。

手順

- 1 root パスワードのハッシュを確認するには、root としてログインして **# more /etc/shadow** コマンドを実行します。

ハッシュ情報が表示されます。

図 7-1. パスワードのハッシュ結果

```
vcac148-084-111:~ # more /etc/shadow
bin:*:16332:0:60:7:::
daemon:*:16332:0:60:7:::
haldaemon:*:16332:0:60:7:::
mail:*:15870::60:::
man:*:16332:0:60:7:::
messagebus:*:16332:0:60:7:::
nobody:*:15870::60:::
ntp:*:16332:0:60:7:::
polkituser:*:16332:0:60:7:::
postfix:*:16332:0:60:7:::
root:$6$PHxGPY5A$ba8KzK4SS44UEHPfAtgs
P/:16346:0:365:7:::
```

- 2 root パスワードに sha512 ハッシュが含まれていない場合は、**passwd** コマンドを実行して変更します。

セキュリティ強化されたすべてのアプライアンスでは、**/etc/pam.d/common-password** ファイル内の **pw_history** モジュールの **enforce_for_root** が有効になります。システムは、デフォルトで前回のパスワードを 5 つまで記憶します。古いパスワードは、**/etc/securetty/passwd** ファイルでユーザーごとに保存されます。

root パスワード履歴の確認

root アカウントにパスワード履歴が適用されていることを確認します。

セキュリティ強化されたすべてのアプライアンスでは、**/etc/pam.d/common-password** ファイル内の **pw_history** モジュールの **enforce_for_root** が有効になります。システムは、デフォルトで前回のパスワードを 5 つまで記憶します。古いパスワードは、**/etc/securetty/passwd** ファイルでユーザーごとに保存されます。

手順

- 1 次のコマンドを実行します。

```
cat /etc/pam.d/common-password-vmware.local | grep pam_pwhistory.so
```

- 2 返される結果に **enforce_for_root** が表示されていることを確認します。

```
password required pam_pwhistory.so enforce_for_root remember=5 retry=3
```

パスワード有効期限の管理

すべてのアカウントのパスワード有効期限を、組織のセキュリティ ポリシーと一致するように構成します。

デフォルトでは、セキュリティ強化されたすべての VMware 仮想アプライアンス アカウントで 60 日のパスワード有効期限を使用します。大半のセキュリティ強化アプライアンスでは、root アカウントは 365 日のパスワード有効期限に設定されています。ベスト プラクティスとして、すべてのアカウントの有効期限がセキュリティと運用の要件基準を満たしていることを確認します。

root パスワードの有効期限が切れている場合、回復することはできません。管理および root のパスワードの有効期限が切れないようにサイト固有のポリシーを実装する必要があります。

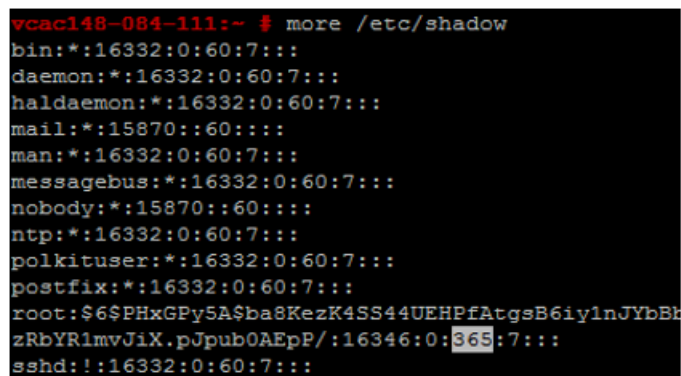
手順

- 1 仮想アプライアンス マシンに root としてログインし、次のコマンドを実行してすべてのアカウントのパスワード有効期限を確認します。

```
# cat /etc/shadow
```

パスワード有効期限は、シャドウ ファイルの 5 番目のフィールドです（フィールドは、コロンで区切られています）。root の有効期限は日単位で設定されます。

図 7-2. パスワード有効期限フィールド



```
vcac148-084-111:~ # more /etc/shadow
bin:!:16332:0:60:7:::
daemon:!:16332:0:60:7:::
haldaemon:!:16332:0:60:7:::
mail:!:15870::60:::
man:!:16332:0:60:7:::
messagebus:!:16332:0:60:7:::
nobody:!:15870::60:::
ntp:!:16332:0:60:7:::
polkituser:!:16332:0:60:7:::
postfix:!:16332:0:60:7:::
root:$6$PHxGPy5A$ba8KezK4SS44UEHPfAtgsB6iy1nJYbBkzRbYRlmvJiX.pJpub0AEpP/:16346:0:365:7:::
sshd:!:16332:0:60:7:::
```

- 2 root アカウントの有効期限を変更するには、次の形式のコマンドを実行します。

```
# passwd -x 365 root
```

このコマンドで、365 はパスワード有効期限までの日数を指定します。組織の有効期限基準を満たすために任意のユーザーを変更するには、このコマンドの「root」を対象のアカウントに置き換え、日数を置き換えて実行します。

セキュア シェルと管理者アカウントの管理

リモート接続のため、セキュリティ強化されたすべてのアプライアンスにセキュア シェル (SSH) プロトコルが含まれます。システムのセキュリティを維持するためには、必要な場合にのみ SSH を使用して、適切に管理します。

SSH は、VMware 仮想アプライアンスに対するリモート接続をサポートするインタラクティブなコマンドライン環境です。デフォルトで、SSH アクセスには高い権限を持つユーザー アカウントの認証情報が必要です。root ユーザーの SSH アクティビティは一般にロールベースのアクセス制御 (RBAC) と仮想アプライアンスの監査制御をバイパスします。

ベスト プラクティスとして、本番環境で SSH を無効にし、その他の方法で解決できない問題をトラブルシューティングする場合にのみ有効にします。特定の目的で、組織のセキュリティ ポリシーに従って必要な間だけ有効にします。vRealize Automation アプライアンスではデフォルトで SSH が無効です。vSphere の構成によっては、Open Virtualization Format (OVF) テンプレートを展開するときに SSH を有効または無効にすることがあります。

マシンで SSH が有効になっているかどうかを決定する単純なテストとしては、SSH を使用して接続を開くことを試行します。接続が開き、認証情報が要求される場合、SSH は有効で接続に使用できます。

セキュア シェル root ユーザー アカウント

VMware アプライアンスでは事前構成済みのユーザー アカウントがないため、root アカウントはデフォルトで SSH を使用して直接ログインできます。できるだけ早く root として SSH を無効にします。

非否認に関するコンプライアンス標準を満たすため、すべてのセキュリティ強化アプライアンスで SSH サーバは、セカンダリ グループ wheel への SSH アクセスを制限するように AllowGroups wheel エントリが事前設定されています。役目を分離するため、sshd などの別のグループを使用するように `/etc/ssh/sshd_config` ファイルの AllowGroups wheel エントリを変更することができます。

wheel グループは `pam_wheel` モジュールによってスーパーユーザーのアクセスが有効になっており、wheel グループのメンバーは `su root` を実行できます (root パスワードが必要です)。グループの分離を使用すると、ユーザーはアプライアンスに対して SSH を実行できますが、`su` を実行して root 権限に切り替えることはできません。AllowGroups フィールドのその他のエントリはアプライアンスの適切な機能に確保しているため、削除したり変更したりしないでください。変更後は、コマンド `# service sshd restart` を実行して SSH デーモンを再起動する必要があります。

vRealize Automation アプライアンス上のセキュア シェルを有効または無効にする

セキュア シェル (SSH) は、トラブルシューティング目的のときにのみ、vRealize Automation アプライアンスで有効にします。通常の本番運用中、次のコンポーネントでは SSH を無効にします。

仮想アプライアンス管理コンソールを使用して、vRealize Automation アプライアンスの SSH を有効または無効にできます。

手順

- 1 vRealize Automation アプライアンスの仮想アプライアンス管理コンソール (VAMI) に移動します。
`https://<vRealizeAppliance url>: 5480`
- 2 [管理者] タブをクリックします。

- 3 [管理者] サブメニューをクリックします。
- 4 [SSH サービスの有効化] チェック ボックスを選択して SSH を有効にするか、または選択解除して無効にします。
- 5 [設定の保存] をクリックして、変更を保存します。

セキュア シェルのローカル管理者アカウントの作成

セキュリティのベスト プラクティスとして、仮想アプライアンス ホスト マシンで セキュア シェル (SSH) のローカル 管理アカウントを作成して構成します。また、適切なアカウントを作成した後は、root による SSH アクセスを削除 します。

SSH のローカル管理アカウント、またはセカンダリ wheel グループのメンバー、またはその両方を作成します。直 接 root アクセスを無効にする前に、許可された管理者が AllowGroups を使用して SSH にアクセスできることと、 wheel グループを使用して su を実行して root 権限に切り替えられることをテストします。

手順

- 1 root として仮想アプライアンスにログインし、適切なユーザー名で次のコマンドを実行します。

```
# useradd -g users <username> -G wheel -m -d /home/<username>
# passwd username
```

wheel は、AllowGroups に ssh アクセスのために指定されたグループです。複数のセカンダリ グループを追加 するには、**-G wheel,sshd** を使用します。

- 2 このユーザーに切り替えて新しいパスワードを指定し、パスワードの複雑性の確認を実施します。

```
# su <-username>
# <username@hostname>:~>passwd
```

パスワードの複雑性が要件を満たしている場合は、パスワードが更新されます。パスワードの複雑性が要件を満 たしていない場合、パスワードは元のパスワードに戻され、パスワード コマンドを再実行する必要があります。

- 3 SSH への直接ログインを削除するには、**/etc/ssh/sshd_config** ファイルを編集して、
(#)PermitRootLogin yes を PermitRootLogin no に置き換えます。

または、仮想アプライアンス管理インターフェイス (VAMI) の [管理] タブで [管理者の SSH ログインを有効化] チェック ボックスの選択を切り替えることで、SSH を有効または無効にできます。

次に進む前に

root としての直接ログインを無効にします。デフォルトで、強化されたアプライアンスは、コンソールを通じた root への直接ログインを許可します。否認防止のための管理アカウントを作成し、それらのアカウントで su root による wheel アクセスをテストしたら、root として **/etc/security** ファイルを編集して、**tty1** エントリを **console** に置き換えて、直接 root ログインを無効にします。

- 1 テキスト エディタで **/etc/securetty** ファイルを開きます。
- 2 **tty1** を見つけて **console** で置き換えます。

- 3 ファイルを保存し、終了します。

セキュア シェル アクセスの制限

システム セキュリティ強化プロセスの一環として、すべての VMware 仮想アプライアンス ホスト マシンで `tcp_wrappers` パッケージを適切に構成することにより、セキュア シェル (SSH) アクセスを制限します。また、これらのアプライアンスに必要な SSH キー ファイルの権限を管理します。

すべての VMware 仮想アプライアンスには、TCP サポート デーモンが `libwrap` されたデーモンにアクセスできるネットワーク サブネットを制御することを許可する `tcp_wrappers` パッケージが含まれます。デフォルトでは、`/etc/hosts.allow` ファイルにセキュア シェルへのすべてのアクセスを許可する汎用エントリ **Sshd:** **ALL : ALLOW** が含まれています。組織の必要に応じてこのアクセスを制限します。

手順

- 1 仮想アプライアンス ホスト マシン上の `/etc/hosts.allow` ファイルをテキスト エディタで開きます。
- 2 セキュアな運用のため、ローカル ホスト エントリと管理ネットワーク サブネットのみを含めるように本番環境の汎用エントリを変更します。

```
sshd:127.0.0.1 : ALLOW
sshd: [::1] : ALLOW
sshd: 10.0.0.0 :ALLOW
```

この例では、すべてのローカル ホストの接続と、クライアントが 10.0.0.0 サブネットで行う接続が許可されます。

- 3 すべての適切なマシン ID (ホスト名、IP アドレス、完全修飾ドメイン名 (FQDN)、ループバックなど) を追加します。
- 4 ファイルを保存し、終了します。

セキュア シェル サーバ構成のセキュリティ強化

すべての VMware アプライアンスは、可能な場合、デフォルトのセキュリティ強化された構成を備えています。ユーザーは、構成ファイルのグローバル オプション セクションのサーバおよびクライアント サービスの設定を調べることによって、構成が適切にセキュリティ強化されていることを確認できます。

可能な場合、`/etc/hosts.allow` ファイルで、SSH サーバの使用を管理サブネットに制限します。

手順

- 1 VMware アプライアンスで `/etc/ssh/sshd_config` サーバ構成ファイルを開き、設定が正しいことを確認します。

設定	ステータス
サーバデーモン プロトコル	Protocol 2
CBC 暗号	aes256-ctr および aes128-ctr
TCP 転送	AllowTCPForwarding no
サーバゲートウェイ ポート	Gateway Ports no
X11 転送	X11Forwarding no

設定	ステータス
SSH サービス	AllowGroups フィールドを使用してアクセスを許可されたグループを指定します。このグループに適切なメンバーを追加します。
GSSAPI 認証	GSSAPIAuthentication no (未使用の場合)
Keberos 認証	KeberosAuthentication no (未使用の場合)
ローカル変数 (AcceptEnv グローバル オプション)	コメント アウトにより、無効に設定、または LC_* または LANG 変数に対して有効に設定
トンネルの構成	PermitTunnel no
ネットワーク セッション	MaxSessions 1
ユーザーの同時接続	root およびその他のユーザーに対して 1 に設定します。 <code>/etc/security/limits.conf</code> ファイルも、同じ設定にする必要があります。
Strict モードの確認	Strict Modes yes
権限分離	UsePrivilegeSeparation yes
rhosts RSA 認証	RhostsESAAuthentication no
圧縮	Compression delayed または Compression no
メッセージ認証コード	MACs hmac-sha1
ユーザー アクセス制限	PermitUserEnvironment no

2 変更内容を保存し、ファイルを閉じます。

セキュア シェル クライアント構成のセキュリティ強化

システム セキュリティ強化プロセスの一環として、仮想アプライアンス ホスト マシンの SSH クライアント構成ファイルを調べて、VMware ガイドラインに従って構成されていることを確認することによって、SSH クライアントのセキュリティ強化を検証します。

手順

- 1 SSH クライアント構成ファイル `/etc/ssh/ssh_config` を開き、グローバル オプション セクションの設定が正しいことを確認します。

設定	ステータス
クライアント プロトコル	Protocol 2
クライアント ゲートウェイ ポート	Gateway Ports no
GSSAPI 認証	GSSAPIAuthentication no
ローカル変数 (SendEnv グローバル オプション)	LC_* または LANG 変数のみ指定
CBC 暗号	aes256-ctr および aes128-ctr のみ
メッセージ認証コード	MACs hmac-sha1 エントリでのみ使用

2 変更内容を保存し、ファイルを閉じます。

セキュア シェル キー ファイルの権限の確認

悪意のある攻撃の可能性を最小限に抑えるためには、仮想アプライアンス ホスト マシンで重要な SSH キー ファイルの権限を維持します。

SSH 構成を設定または更新したら、次の SSH キー ファイルの権限に変更がないことを常に確認します。

- `/etc/ssh/*key.pub` 内のパブリック ホスト キーは root ユーザーが所有し、権限が 0644 (-rw-r--r--) に設定されています。
- `/etc/ssh/*key` 内のプライベート ホスト キーは root ユーザーが所有し、権限が 0600 (-rw-----) に設定されています。

SSH キー ファイルの権限を確認する

パブリック キー ファイルとプライベート キー ファイルの両方に SSH 権限が適用されていることを確認します。

手順

- 1 次のコマンドを実行して SSH パブリック キー ファイルを確認します。 `ls -l /etc/ssh/*key.pub`。
- 2 所有者が root であり、グループの所有者が root であり、ファイル権限が 0644 (-rw-r--r--) に設定されていることを確認します。
- 3 次のコマンドを実行して、問題を修正します。


```
chown root /etc/ssh/*key.pub
chgrp root /etc/ssh/*key.pub
chmod 644 /etc/ssh/*key.pub
```
- 4 次のコマンドを実行して SSH プライベート キー ファイルを確認します。 `ls -l /etc/ssh/*key`。
- 5 次のコマンドを実行して、問題を修正します。


```
chown root /etc/ssh/*key
chgrp root /etc/ssh/*key
chmod 644 /etc/ssh/*key
```

仮想アプライアンス管理インターフェイスのユーザーの変更

仮想アプライアンス管理インターフェイスでユーザーを追加および削除して、適切なレベルのセキュリティを作成することができます。

仮想アプライアンス管理インターフェイスの root ユーザー アカウントでは、認証に PAM を使用するため、PAM で設定されたクリップ レベルも適用されます。仮想アプライアンス管理インターフェイスを適切に分離していないと、攻撃者が総当たり攻撃を試みたときにシステムの root アカウントでロックアウトが発生することがあります。さらに、root アカウントが組織内の複数のユーザーによる否認防止を付与するのに不十分とみなされた場合は、管理インターフェイスの管理者ユーザーを変更することになる可能性もあります。

開始する前に

手順

- 1 新しいユーザーを作成し、仮想アプライアンス管理インターフェイス グループに追加するには、次のコマンドを実行します。

```
useradd -G vami,root <user>
```

- 2 ユーザーのパスワードを作成します。

```
passwd <user>
```

- 3 (オプション) 仮想アプライアンス管理インターフェイスの root アクセス権を無効にするには、次のコマンドを実行します。

```
usermod -R vami root
```

注意 仮想アプライアンス管理インターフェイスへの root アクセス権を無効にすると、[管理] タブの管理者、または root、パスワードをアップデートする機能も無効になります。

ブートローダー認証の設定

適切なレベルのセキュリティを提供するには、VMware 仮想アプライアンスでブートローダー認証を構成します。

システムのブートローダーに認証が必要ない場合、システム コンソールのアクセス権を持つユーザーが、システム ブート構成を変更したり、シングル ユーザー モードまたはメンテナンス モードでシステムをブートしたりできます。この結果、サービス拒否または未認証のシステム アクセスが可能になります。ブートローダー認証は VMware 仮想アプライアンスでデフォルトで設定されていないため、その構成には GRUB パスワードを作成する必要があります。

手順

- 1 仮想アプライアンスの `/boot/grub/menu.lst` ファイル内で `password --md5 <password-hash>` 行を特定して、ブート パスワードが保存されているかどうかを確認します。

- 2 パスワードが保存されていない場合は、仮想アプライアンスで `# /usr/sbin/grub-md5-crypt` コマンドを実行します。

MD5 パスワードが生成され、コマンドによって md5 ハッシュ出力が指定されます。

- 3 `# password --md5 <hash from grub-md5-crypt>` コマンドを実行して、`menu.lst` ファイルにパスワードを追加します。

NTP の設定

クリティカルなタイムソースでは、ホストの時刻同期を無効にして、vRealize Automation アプライアンスで Network Time Protocol (NTP) を使用します。

vRealize Automation アプライアンスの NTP デーモンは、同期された時間サービスを提供します。NTP は、デフォルトでは無効になっているため、手動で設定する必要があります。可能な場合は、本番環境で NTP を使用して、正確な監査とログ保存を通じてユーザー アクションを追跡し、潜在的な悪意のある攻撃と侵入を検出します。NTP のセキュリティ上の注意事項については、NTP の Web サイトを参照してください。

NTP の構成ファイルは、各アプライアンス上の **/etc/** フォルダに含まれています。仮想アプライアンス管理インターフェイスの [管理] タブで、vRealize Automation アプライアンスの NTP サービスを有効にし、タイム サーバを追加できます。

手順

- 1 仮想アプライアンス ホスト マシン上の **/etc/ntp.conf** 構成ファイルをテキスト エディタを使用して開きます。
- 2 ファイルの所有権を **root:root** に設定します。
- 3 権限を **0640** に設定します。
- 4 NTP サービスに対する DNS amp（サービス拒否増幅）攻撃のリスクを低減するには、**/etc/ntp.conf** ファイルを開き、ファイルに restrict 行が表示されることを確認します。

```
restrict default kod nomodify notrap nopeer noquery
restrict -6 default kod nomodify notrap nopeer noquery
restrict 127.0.0.1
restrict -6 ::1
```

- 5 変更内容を保存し、ファイルを閉じます。

転送中の vRealize Automation アプライアンス データの TLS の構成

vRealize Automation アプライアンス コンポーネントの転送チャネルをセキュリティ保護するために、vRealize Automation デプロイで強力な TLS プロトコルを使用していることを確認します。

パフォーマンスに関する考慮事項のために、一部のアプリケーション サービス間のローカルホスト接続に対して TLS は有効になっていません。多層防御が必要な場合は、すべてのローカルホスト通信で TLS を有効にします。

重要 ロード バランサの TLS を終了する場合は、すべてのロード バランサで SSLv2、SSLv3、および TLS 1.0 などのセキュアでないプロトコルを無効にします。

localhost 構成で TLS を有効にする

デフォルトでは、一部の localhost 通信は TLS を使用しません。セキュリティの強化を提供するため、すべての localhost 接続で TLS を有効にできます。

手順

- 1 SSH を使用して vRealize Automation アプライアンスに接続します。
- 2 次のコマンドを実行して、VCAC キーストアの権限を設定します。

```
usermod -A vco,coredump,pivotal vco
chown vcac.pivotal /etc/vcac/vcac.keystore
chmod 640 /etc/vcac/vcac.keystore
```

3 HAProxy の構成を更新します。

- a `/etc/haproxy/conf.d` にある HAProxy 構成ファイルを開き、`20-vcac.cfg` サービスを選択します。
- b 次の文字列を含む行を見つけます。

server local 127.0.0.1...。その後、この行の末尾に次の文字列を追加します：**ssl verify none**

このセクションには、次のような行が含まれています。

backend-horizon	backend-vro
backend-vra	backend-artifactory
backend-vra-health	

- c backend-horizon のポートを 8080 から 8443 に変更します。

4 keystorePass のパスワードを取得します。

- a `/etc/vcac/security.properties` ファイル内の **certificate.store.password** プロパティを見つけます。

例：**certificate.store.password=s2enc~iom0GXATG+RB8ff7Wdm4Bg==**

- b 次のコマンドを使用して、値を復号化します。

vcac-config prop-util -d --p VALUE

例：**vcac-config prop-util -d --p s2enc~iom0GXATG+RB8ff7Wdm4Bg==**

5 vRealize Automation サービスを構成します。

- a `/etc/vcac/server.xml` ファイルを開きます。
- b 次の属性を Connector タグに追加します。certificate.store.password は、`/etc/vcac/security.properties` にある証明書ストア パスワード値に置き換えます。

```
scheme="https" secure="true" SSLEnabled="true" sslProtocol="TLS"
keystoreFile="/etc/vcac/vcac.keystore" keyAlias="apache" keystorePass="
certificate.store.password"
```

6 vRealize Orchestrator サービスを構成します。

- a `/etc/vco/app-server.xml` ファイルを開きます。
- b 次の属性を Connector タグに追加します。certificate.store.password は、`/etc/vcac/security.properties` にある証明書ストア パスワード値に置き換えます。

```
scheme="https" secure="true" SSLEnabled="true" sslProtocol="TLS"
keystoreFile="/etc/vcac/vcac.keystore" keyAlias="apache" keystorePass="
certificate.store.password"
```

- 7 vRealize Orchestrator、vRealize Automation、および haproxy サービスを再起動します。

```
service vcac-server restart
service vco-server restart
service haproxy restart
```

注意 vco-server が再起動しない場合は、ホスト コンピュータを再起動します。

- 8 仮想アプライアンス管理インターフェイスを構成します。

- a `/opt/vmware/share/htdocs/service/café-services/services.py` ファイルを開きます。
- b `conn = httplib.HTTP()` の行を `conn = httplib.HTTPS()` に変更してセキュリティを強化します。

連邦情報処理標準 (FIPS) 140-2 準拠の処理を有効にする

vRealize Automation アプライアンスは、すべての受信および送信ネットワーク トラフィックについて、TLS 経由の転送中データに対して OpenSSL の連邦情報処理標準 (FIPS) 140-2 認定済みバージョンを使用するようになりました。

vRealize Automation アプライアンス管理インターフェイスで FIPS モードを有効または無効にできます。root としてログインしているときに、コマンドラインから次のコマンドを使用して FIPS を構成することもできます。

```
vcac-vami fips enable
vcac-vami fips disable
vcac-vami fips status
```

FIPS を有効にすると、ポート 443 の受信および送信 vRealize Automation アプライアンス ネットワーク トラフィックには FIPS 140-2 準拠の暗号化が使用されます。FIPS の設定に関係なく、vRealize Automation では vRealize Automation アプライアンスに保存されたデータには AES-256 が使用され、セキュアに保護されます。

注意 一部の内部コンポーネントでは認定済みの暗号化モジュールがまだ使用されていないため、現在は vRealize Automation の一部のみで FIPS 準拠が有効になっています。認定済みのモジュールがまだ実装されていない場合、すべての暗号化アルゴリズムで AES-256 ベースの暗号化が使用されます。

注意 次の手順では、構成を変更するときに、物理マシンが再起動されます。

手順

- 1 vRealize Automation アプライアンスの管理インターフェイスに root としてログインします。
https:// vrealize-automation-appliance-FQDN:5480
- 2 [vRA 設定] - [ホストの設定] の順に選択します。
- 3 右上にある [アクション] という見出しの下ボタンをクリックして FIPS を有効または無効にします。
- 4 [はい] をクリックして vRealize Automation アプライアンスを再起動します。

SSLv3、TLS 1.0、および TLS 1.1 が無効になっていることを確認する

セキュリティ強化プロセスの一環として、デプロイした vRealize Automation アプライアンスがセキュアな通信チャネルを使用していることを確認します。

開始する前に

[\[localhost 構成で TLS を有効にする\]](#) を実行します。

手順

- 1 vRealize Automation アプライアンスの HAProxy https ハンドラで SSLv3、TLS 1.0、および TLS 1.1 が無効になっていることを確認します。

確認するファイル	次の記載があることを確認	確認する行
/etc/haproxy/conf.d/20-vcac.cfg	no-ssl3 no-tls10 no-tls11 force-tls12	bind 0.0.0.0:443 ssl crt /etc/apache2/server.pem ciphers !aNULL:!eNULL:kECDH+AESGCM:ECDH+AESGCM:RSA+AESGCM:kECDH+AES:ECDH+AES:RSA+AES:@STRENGTH no-ssl3 no-tls10 no-tls11
/etc/haproxy/conf.d/30-vro-config.cfg	no-ssl3 no-tls10 no-tls11 force-tls12	bind :::8283 v4v6 ssl crt /opt/vmware/etc/lighttpd/server.pem ciphers !aNULL:!eNULL:kECDH+AESGCM:ECDH+AESGCM:RSA+AESGCM:kECDH+AES:ECDH+AES:RSA+AES:@STRENGTH no-ssl3 no-tls10 no-tls11

- 2 サービスを再起動します。

```
service haproxy restart
```

- 3 /opt/vmware/etc/lighttpd/lighttpd.conf ファイルを開いて、無効にするための正しいエントリがあることを確認します。

注意 Lighttpd には、TLS 1.0 または TLS 1.1 を無効にするディレクティブがありません。TLS 1.0 と TLS 1.1 の使用に対する制限は、OpenSSL が TLS 1.0 と TLS 1.1 の暗号スイートを使用しないように適用することで、部分的に緩和できます。

```
ssl.use-ssl2 = "disable"
ssl.use-ssl3 = "disable"
```

- 4 vRealize Automation アプライアンスのコンソール プロキシで SSLv3、TLS 1.0、および TLS 1.1 が無効になっていることを確認します。

- a `/etc/vcac/security.properties` ファイルを編集して、次の行を追加または変更します。

```
consoleproxy.ssl.server.protocols = TLSv1.2
```

- b 次のコマンドを実行して、サーバを再起動します。

```
service vcac-server restart
```

- 5 vCO サービスで SSLv3、TLS 1.0、および TLS 1.1 が無効になっていることを確認します。

- a `/etc/vco/app-server/server.xml` ファイル内の `<Connector>` タグを見つけて、次の属性を追加します。

```
sslEnabledProtocols = "TLSv1.2"
```

- b 次のコマンドを実行して vCO サービスを再起動します。

```
service vco-server restart
```

- 6 vRealize Automation サービスで SSLv3、TLS 1.0、および TLS 1.1 が無効になっていることを確認します。

- a `/etc/vcac/server.xml` ファイル内の `<Connector>` タグに次の属性を追加します。

```
sslEnabledProtocols = "TLSv1.2"
```

- b 次のコマンドを実行して vRealize Automation サービスを再起動します。

```
service vcac-server restart
```

- 7 RabbitMQ で SSLv3、TLS 1.0、および TLS 1.1 が無効になっていることを確認します。

`/etc/rabbitmq/rabbitmq.config` ファイルを開いて、`{versions, ['tlsv1.2', 'tlsv1.1']}` が `ssl` および `ssl_options` セクションに表示されていることを確認します。

```
[
  {ssl, [
    {versions, ['tlsv1.2', 'tlsv1.1']},
    {ciphers, ["AES256-SHA", "AES128-SHA"]}
  ]},
  {rabbit, [
    {tcp_listeners, [{"127.0.0.1", 5672}]},
    {frame_max, 262144},
    {ssl_listeners, [5671]},
    {ssl_options, [
      {cacertfile, "/etc/rabbitmq/certs/ca/cacert.pem"},
      {certfile, "/etc/rabbitmq/certs/server/cert.pem"},
      {keyfile, "/etc/rabbitmq/certs/server/key.pem"},
      {versions, ['tlsv1.2', 'tlsv1.1']},
      {ciphers, ["AES256-SHA", "AES128-SHA"]},
      {verify, verify_peer},
      {fail_if_no_peer_cert, false}
    ]},
    {mnesia_table_loading_timeout, 600000},
```

```
{cluster_partition_handling, autoheal},
{heartbeat, 600}
]],
{kernel, [{net_ticktime, 120}]}
].
```

- 8 RabbitMQ サーバを再起動します。

```
# service rabbitmq-server restart
```

- 9 vIDM サービスで SSLv3、TLS 1.0、および TLS 1.1 が無効になっていることを確認します。

SSLEnabled="true" を含むコネクタの各インスタンスで

opt/vmware/horizon/workspace/conf/server.xml ファイルを開いて、次の行が表示されていることを確認します。

```
sslEnabledProtocols="TLSv1.2"
```

vRealize Automation コンポーネント用の TLS 暗号スイートの構成

セキュリティを最大限高めるには、強力な暗号を使用するように vRealize Automation コンポーネントを設定する必要があります。

サーバとブラウザの間でネゴシエートされる暗号化により、TLS セッションで使用される暗号化の強度が決まります。

強力な暗号のみが確実に選択されるようにするため、vRealize Automation コンポーネントで強度の弱い暗号を無効にします。強力な暗号のみをサポートし、十分に大きいキー サイズを使用するように、サーバを設定します。また、適切な順序ですべての暗号を構成します。

NULL 暗号スイート、aNULL、または eNULL などの認証を提供しない暗号スイートを無効にします。この他に、匿名の Diffie-Hellman キー交換 (ADH)、エクスポート レベルの暗号 (EXP、DES を含んでいる暗号)、ペイロードトラフィックの暗号化に対する 128 ビットより小さいサイズのキー、ペイロードトラフィックのハッシュメカニズムとしての MD5 の使用、IDEA 暗号スイート、および RC4 暗号スイートも無効にします。また、Diffie-Hellman (DHE) キー交換を使用する暗号スイートが無効になっていることを確認します。

HA プロキシの強度の弱い暗号を無効にする

使用可能な暗号のリストに照らし合わせて、vRealize Automation アプライアンス HA プロキシ サービスの暗号を確認し、強度が弱いとみなされるものすべてを無効にします。

NULL 暗号スイート、aNULL、または eNULL などの認証を提供しない暗号スイートを無効にします。この他に、匿名の Diffie-Hellman キー交換 (ADH)、エクスポート レベルの暗号 (EXP、DES を含んでいる暗号)、ペイロードトラフィックの暗号化に対する 128 ビットより小さいサイズのキー、ペイロードトラフィックのハッシュメカニズムとしての MD5 の使用、IDEA 暗号スイート、および RC4 暗号スイートも無効にします。

手順

- 1 `/etc/haproxy/conf.d/20-vcac.cfg` ファイルのバインド ディレクティブの暗号エントリを確認し、強度が弱いとみなされるすべてのものを無効にします。

```
bind 0.0.0.0:443 ssl crt /etc/apache2/server.pem ciphers !aNULL:!  
eNULL:kECDH+AESGCM:ECDH+AESGCM:RSA+AESGCM:kECDH+AES:ECDH+AES:RSA  
+AES:@STRENGTH no-sslv3 no-tls1 no-tls11
```

- 2 `/etc/haproxy/conf.d/30-vro-config.cfg` ファイルのバインド ディレクティブの暗号エントリを確認し、強度が弱いとみなされるすべてのものを無効にします。

```
bind :::8283 v4v6 ssl crt /opt/vmware/etc/lighttpd/server.pem ciphers !  
aNULL:!eNULL:kECDH+AESGCM:ECDH+AESGCM:RSA+AESGCM:kECDH+AES:ECDH+AES:RSA  
+AES:@STRENGTH no-sslv3 no-tls1 no-tls11
```

vRealize Automation アプライアンス vRealize Automation アプライアンス コンソール プロキシ サービスの強度の弱い暗号を無効にする

使用可能な暗号のリストに照らし合わせて、vRealize Automation アプライアンス コンソール プロキシ サービスの暗号を確認し、強度が弱いとみなされるものすべてを無効にします。

NULL 暗号スイート、aNULL、または eNULL などの認証を提供しない暗号スイートを無効にします。この他に、匿名の Diffie-Hellman キー交換 (ADH)、エクスポート レベルの暗号 (EXP、DES を含んでいる暗号)、ペイロードトラフィックの暗号化に対する 128 ビットより小さいサイズのキー、ペイロードトラフィックのハッシュ メカニズムとしての MD5 の使用、IDEA 暗号スイート、および RC4 暗号スイートも無効にします。

手順

- 1 `/etc/vcac/security.properties` ファイルをテキスト エディタで開きます。
- 2 不要な暗号スイートを無効にするには、ファイルに行を追加します。

次の行に必要な変更を加えて使用します。

```
consoleproxy.ssl.ciphers.disabled=<cipher_suite_1, cipher_suite_2,etc>
```

たとえば、AES 128 および AES 256 の暗号スイートを無効にするには、次の行を追加します。

```
consoleproxy.ssl.ciphers.disabled=TLS_DH_DSS_WITH_AES_128_CBC_SHA,  
TLS_DH_DSS_WITH_AES_256_CBC_SHA, TLS_DH_RSA_WITH_AES_256_CBC_SHA,  
TLS_DHE_DSS_WITH_AES_128_CBC_SHA, TLS_DHE_DSS_WITH_AES_256_CBC_SHA,  
TLS_DHE_RSA_WITH_AES_128_CBC_SHA, TLS_DHE_RSA_WITH_AES_256_CBC_SHA
```

- 3 次のコマンドを使用してサーバを再起動します。

```
service vcac-server restart
```

vRealize Automation アプライアンス vCO サービスの強度の弱い暗号を無効にする

使用可能な暗号のリストに照らし合わせて、vRealize Automation アプライアンス vCO サービスの暗号を確認し、強度が弱いとみなされるものすべてを無効にします。

NULL 暗号スイート、aNULL、または eNULL などの認証を提供しない暗号スイートを無効にします。この他に、匿名の Diffie-Hellman キー交換 (ADH)、エクスポート レベルの暗号 (EXP、DES を含んでいる暗号)、ペイロードトラフィックの暗号化に対する 128 ビットより小さいサイズのキー、ペイロードトラフィックのハッシュメカニズムとしての MD5 の使用、IDEA 暗号スイート、および RC4 暗号スイートも無効にします。

手順

- 1 `/etc/vco/app/server/server.xml` ファイル内で `<Connector>` タグを見つけます。
- 2 目的の暗号スイートを使用するように、暗号の属性を編集または追加します。

次の例を参照してください。

```
ciphers="TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256,TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256,TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256,TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256,TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384,TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384,TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384,TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384"
```

vRealize Automation アプライアンス RabbitMQ サービスの強度の弱い暗号を無効にする

使用可能な暗号のリストに照らし合わせて、vRealize Automation アプライアンス RabbitMQ サービスの暗号を確認し、強度が弱いとみなされるものすべてを無効にします。

NULL 暗号スイート、aNULL、または eNULL などの認証を提供しない暗号スイートを無効にします。この他に、匿名の Diffie-Hellman キー交換 (ADH)、エクスポート レベルの暗号 (EXP、DES を含んでいる暗号)、ペイロードトラフィックの暗号化に対する 128 ビットより小さいサイズのキー、ペイロードトラフィックのハッシュメカニズムとしての MD5 の使用、IDEA 暗号スイート、および RC4 暗号スイートも無効にします。

手順

- 1 サポートされる暗号スイートを評価します。`# /usr/sbin/rabbitmqctl eval 'ssl:cipher_suites()'` コマンドを実行します。

次の例で返される暗号は、サポートされる暗号のみを表しています。RabbitMQ サーバは、`rabbitmq.config` ファイルでこれらの暗号を使用またはアドバタイズするように構成されていない限り、使用またはアドバタイズを行いません。

```
["ECDHE-ECDSA-AES256-GCM-SHA384","ECDHE-RSA-AES256-GCM-SHA384",
"ECDHE-ECDSA-AES256-SHA384","ECDHE-RSA-AES256-SHA384",
"ECDH-ECDSA-AES256-GCM-SHA384","ECDH-RSA-AES256-GCM-SHA384",
"ECDH-ECDSA-AES256-SHA384","ECDH-RSA-AES256-SHA384",
"DHE-RSA-AES256-GCM-SHA384","DHE-DSS-AES256-GCM-SHA384",
"DHE-RSA-AES256-SHA256","DHE-DSS-AES256-SHA256","AES256-GCM-SHA384",
"AES256-SHA256","ECDHE-ECDSA-AES128-GCM-SHA256",
"ECDHE-RSA-AES128-GCM-SHA256","ECDHE-ECDSA-AES128-SHA256",
"ECDHE-RSA-AES128-SHA256","ECDH-ECDSA-AES128-GCM-SHA256",
"ECDH-RSA-AES128-GCM-SHA256","ECDH-ECDSA-AES128-SHA256",
"ECDH-RSA-AES128-SHA256","DHE-RSA-AES128-GCM-SHA256",
"DHE-DSS-AES128-GCM-SHA256","DHE-RSA-AES128-SHA256","DHE-DSS-AES128-SHA256",
"AES128-GCM-SHA256","AES128-SHA256","ECDHE-ECDSA-AES256-SHA",
"ECDHE-RSA-AES256-SHA","DHE-RSA-AES256-SHA","DHE-DSS-AES256-SHA",
"ECDH-ECDSA-AES256-SHA","ECDH-RSA-AES256-SHA","AES256-SHA",
"ECDHE-ECDSA-DES-CBC3-SHA","ECDHE-RSA-DES-CBC3-SHA","EDH-RSA-DES-CBC3-SHA",
```

```
"EDH-DSS-DES-CBC3-SHA", "ECDH-ECDSA-DES-CBC3-SHA", "ECDH-RSA-DES-CBC3-SHA",
"DES-CBC3-SHA", "ECDHE-ECDSA-AES128-SHA", "ECDHE-RSA-AES128-SHA",
"DHE-RSA-AES128-SHA", "DHE-DSS-AES128-SHA", "ECDH-ECDSA-AES128-SHA",
"ECDH-RSA-AES128-SHA", "AES128-SHA"]
```

- 2 組織のセキュリティ要件を満たす、サポートされる暗号を選択します。

たとえば、**ECDHE-ECDSA-AES128-GCM-SHA256 & ECDHE-ECDSA-AES256-GCM-SHA384** だけを許可するには、**/etc/rabbitmq/rabbitmq.config** ファイルを確認し、**ssl** および **ssl_options** に次の行を追加します。

```
{ciphers, [{"ECDHE-ECDSA-AES128-GCM-SHA256"}, {"ECDHE-ECDSA-AES256-GCM-SHA384"}]}
```

- 3 次のコマンドを使用して、RabbitMQ サーバを再起動します。

```
service rabbitmq-server restart
```

保存データのセキュリティの確認

vRealize Automation で使用されるデータベースのユーザーおよびアカウントのセキュリティを確認します。

Postgres ユーザー

Postgres の Linux ユーザー アカウントは Postgres データベースのスーパーユーザー アカウント ロールに関連付けられています。デフォルトでは、アカウントはロックされています。**root** ユーザー アカウントからしかアクセスできないため、このユーザーには最もセキュアな構成です。このユーザー アカウントのロックを解除しないでください。

データベースのユーザー アカウント ロール

デフォルトの Postgres ユーザー アカウント ロールは、アプリケーション機能以外では使用しないでください。デフォルト以外のデータベースのレビューまたはレポート処理をサポートするには、追加のアカウントを作成して、パスワードを適切に保護する必要があります。

コマンド ラインで次のスクリプトを実行します。

```
vcac-vami add-db-user newUsername newPassword
```

これで、ユーザーによって指定された新しいユーザーとパスワードが追加されます。

注意 マスター スレーブ HA Postgres 設定が構成されている場合は、マスター Postgres データベースに対してこのスクリプトを実行する必要があります。

PostgreSQL クライアント認証の構成

ローカル trust 認証が vRealize Automation アプライアンス PostgreSQL データベースで構成されていないことを確認します。この構成では、データベーススーパー ユーザーを含むすべてのローカルユーザーが、任意の PostgreSQL ユーザーとしてパスワードなしで接続することが可能となります。

注意 Postgre のスーパー ユーザー アカウントは、ローカル trust のままにする必要があります。

暗号化されたパスワードが送信されるため、md5 認証方法をお勧めします。

クライアント認証構成設定は、`/storage/db/pgdata/pg_hba.conf` ファイル内に存在します。

```
# TYPE  DATABASE      USER          ADDRESS          METHOD

# "local" is for Unix domain socket connections only
local    all             postgres               trust
# IPv4 local connections:
#host    all             all                127.0.0.1/32      md5
hostssl  all             all                127.0.0.1/32      md5
# IPv6 local connections:
#host    all             all                ::1/128           md5
hostssl  all             all                ::1/128           md5

# Allow remote connections for VCAC user.
#host    vcac             vcac              0.0.0.0/0         md5
hostssl  vcac             vcac              0.0.0.0/0         md5
hostssl  vcac             vcac              ::0/0             md5
# Allow remote connections for VCAC replication user.
#host    vcac             vcac_replication  0.0.0.0/0         md5
hostssl  vcac             vcac_replication  0.0.0.0/0         md5
hostssl  vcac             vcac_replication  ::0/0             md5
# Allow replication connections by a user with the replication privilege.
#host    replication      vcac_replication  0.0.0.0/0         md5
hostssl  replication      vcac_replication  0.0.0.0/0         md5
hostssl  replication      vcac_replication  ::0/0             md5
```

`pg_hba.conf` ファイルを編集する場合、変更を有効にするには、次のコマンドを実行して Postgre サーバを再起動する必要があります。

```
# cd /opt/vmware/vpostgres/9.2/bin
# su postgres
# ./pg_ctl restart -D /storage/db/pgdata/ -m fast
```

vRealize Automation アプリケーション リソースの構成

vRealize Automation アプリケーション リソースを確認し、ファイルの権限を制限します。

手順

- 1 SUID および GUID ビットが設定されたファイルが正しく定義されていることを確認するには、次のコマンドを実行します。

```
find / -path /proc -prune -o -type f -perm +6000 -ls
```

次のリストが表示されます。

```
2197357  24 -rwsr-xr-x  1 polkituser root      23176 Mar 31
2015 /usr/lib/PolicyKit/polkit-set-default-helper
2197354  16 -rwxr-sr-x  1 root      polkituser 14856 Mar 31
2015 /usr/lib/PolicyKit/polkit-read-auth-helper
2197353  12 -rwsr-x---  1 root      polkituser 10744 Mar 31
```

```

2015 /usr/lib/PolicyKit/polkit-grant-helper-pam
2197352 20 -rwxr-sr-x 1 root polkituser 19208 Mar 31
2015 /usr/lib/PolicyKit/polkit-grant-helper
2197351 20 -rwxr-sr-x 1 root polkituser 19008 Mar 31
2015 /usr/lib/PolicyKit/polkit-explicit-grant-helper
2197356 24 -rwxr-sr-x 1 root polkituser 23160 Mar 31
2015 /usr/lib/PolicyKit/polkit-revoke-helper
2188203 460 -rws--x--x 1 root root 465364 Apr 21 22:38 /usr/lib64/ssh/ssh-
keysign
2138858 12 -rwxr-sr-x 1 root tty 10680 May 10 2010 /usr/sbin/utempter
2142482 144 -rwsr-xr-x 1 root root 142890 Sep 15 2015 /usr/bin/passwd
2142477 164 -rwsr-xr-x 1 root shadow 161782 Sep 15 2015 /usr/bin/chage
2142467 156 -rwsr-xr-x 1 root shadow 152850 Sep 15 2015 /usr/bin/chfn
1458298 364 -rwsr-xr-x 1 root root 365787 Jul 22 2015 /usr/bin/sudo
2142481 64 -rwsr-xr-x 1 root root 57776 Sep 15 2015 /usr/bin/newgrp
1458249 40 -rwsr-x--- 1 root trusted 40432 Mar 18 2015 /usr/bin/crontab
2142478 148 -rwsr-xr-x 1 root shadow 146459 Sep 15 2015 /usr/bin/chsh
2142480 156 -rwsr-xr-x 1 root shadow 152387 Sep 15 2015 /usr/bin/gpasswd
2142479 48 -rwsr-xr-x 1 root shadow 46967 Sep 15 2015 /usr/bin/expiry
311484 48 -rwsr-x--- 1 root messagebus 47912 Sep 16 2014 /lib64/dbus-1/dbus-
daemon-launch-helper
876574 36 -rwsr-xr-x 1 root shadow 35688 Apr 10 2014 /sbin/unix_chkpwd
876648 12 -rwsr-xr-x 1 root shadow 10736 Dec 16 2011 /sbin/unix2_chkpwd
49308 68 -rwsr-xr-x 1 root root 63376 May 27 2015 /opt/likewise/bin/ksu
1130552 40 -rwsr-xr-x 1 root root 40016 Apr 16 2015 /bin/su
1130511 40 -rwsr-xr-x 1 root root 40048 Apr 15 2011 /bin/ping
1130600 100 -rwsr-xr-x 1 root root 94808 Mar 11 2015 /bin/mount
1130601 72 -rwsr-xr-x 1 root root 69240 Mar 11 2015 /bin/umount
1130512 36 -rwsr-xr-x 1 root root 35792 Apr 15 2011 /bin/ping6
2012 /lib64/dbus-1/dbus-daemon-launch-helper

```

- 2 仮想アプライアンスのすべてのファイルに所有者があることを確認するには、次のコマンドを実行します。

```
find / -path /proc -prune -o -nouser -o -nogroup
```

- 3 すべてのファイルの仮想アプライアンスに対する権限を確認し、あらゆるユーザーが書き込み可能なファイルがないことを確認するには、次のコマンドを実行します。

```
find / -name ".*" -type f -perm -a+w | xargs ls -ldb
```

- 4 vcac ユーザーのみが適切なファイルを所有していることを確認するには、次のコマンドを実行します。

```
find / -name "proc" -prune -o -user vcac -print | egrep -v -e "**/vcac/*" |
egrep -v -e "**/vmware-vcac/*"
```

結果が表示されない場合、すべての正しいファイルが vcac ユーザーによってのみ所有されています。

- 5 次のファイルが vcac ユーザーによってのみ書き込み可能であることを確認します。

```

/etc/vcac/vcac/security.properties
/etc/vcac/vcac/solution-users.properties
/etc/vcac/vcac/sso-admin.properties
/etc/vcac/vcac/vcac.keystore

```



```
/etc/vcac/vcac/vcac.properties
```

また、次のファイルおよびそのサブディレクトリも確認します。

```
/var/log/vcac/*
```

```
/var/lib/vcac/*
```

```
/var/cache/vcac/*
```

- 6 vcac または root ユーザーのみが次のディレクトリとそのサブディレクトリにある正しいファイルを読み取ることができていることを確認します。

```
/etc/vcac/*
```

```
/var/log/vcac/*
```

```
/var/lib/vcac/*
```

```
/var/cache/vcac/*
```

- 7 次のディレクトリとそのサブディレクトリ内に示されるように、正しいファイルが vco または root ユーザーのみによって所有されていることを確認します。

```
/etc/vco/*
```

```
/var/log/vco/*
```

```
/var/lib/vco/*
```

```
/var/cache/vco/*
```

- 8 次のディレクトリとそのサブディレクトリ内に示されるように、正しいファイルが vco または root ユーザーのみによって書き込み可能であることを確認します。

```
/etc/vco/*
```

```
/var/log/vco/*
```

```
/var/lib/vco/*
```

```
/var/cache/vco/*
```

- 9 次のディレクトリとそのサブディレクトリ内に示されるように、正しいファイルが vco または root ユーザーのみによって読み取り可能であることを確認します。

```
/etc/vco/*
```

```
/var/log/vco/*
```

```
/var/lib/vco/*
```

```
/var/cache/vco/*
```

コンソール プロキシ構成のカスタマイズ

vRealize Automation のリモート コンソールの構成をカスタマイズして、トラブルシューティングや組織の活動を促進できます。

vRealize Automation をインストール、構成、または保守する際に、いくつかの設定を変更してトラブルシューティングやデバッグを行えます。加えた変更を記録して監査し、該当するコンポーネントが用途に従って確実にセキュリティで保護されるようにします。設定の変更が正しくセキュリティで保護されているかどうか分からない場合は、本番環境には適用しないでください。

VMware Remote Console チケットの有効期限のカスタマイズ

VMware Remote Console 接続の確立に使用されるリモート コンソールのチケットの有効期限をカスタマイズすることができます。

ユーザーが VMware Remote Console に接続すると、システムは、仮想マシンへの特定の接続を確立するワンタイム認証情報を作成して返します。チケットの有効期限を分単位の時間枠で設定できます。

手順

- 1 `/etc/vcac/security.properties` ファイルをテキスト エディタで開きます。
- 2 `consoleproxy.ticket.validitySec=30` の形式でファイルに行を追加します。
この行の数値は、チケットの有効期限が切れるまでの分数を指定します。
- 3 ファイルを保存し、終了します。
- 4 `/etc/init.d/vcac-server restart` コマンドを使用して、vcac サーバを再起動します。

チケットの有効期限の値が、指定した分単位の時間枠にリセットされます。

コンソール プロキシ サーバのポートのカスタマイズ

VMware Remote Console のコンソール プロキシがメッセージをリッスンするポートをカスタマイズすることができます。

手順

- 1 `/etc/vcac/security.properties` ファイルをテキスト エディタで開きます。
- 2 `consoleproxy.service.port=8445` の形式でファイルに行を追加します。
数値は、コンソール プロキシ サービスのポート番号を指定します。この場合は 8445 です。
- 3 ファイルを保存し、終了します。
- 4 `/etc/init.d/vcac-server restart` コマンドを使用して、vcac サーバを再起動します。

プロキシ サービス ポートの番号が、指定したポート番号に変更されます。

X-XSS-Protection 応答ヘッダーの構成

Haproxy 構成ファイルに、X-XSS-Protection 応答ヘッダーを追加します。

手順

- 1 編集するために `/etc/haproxy/conf.d/20-vcac.cfg` を開きます。
- 2 フロント エンド セクションで、次の行を追加します。

```
rspdel X-XSS-Protection:\ 1;\ mode=block
rspadd X-XSS-Protection:\ 1;\ mode=block
```

- 3 次のコマンドを使用して、HAProxy 構成を再ロードします。

```
/etc/init.d/haproxy reload
```

HTTP Strict Transport Security 応答ヘッダーの構成

HAProxy 構成に HTTP Strict Transport (HSTS) 応答ヘッダーを追加します。

手順

- 1 編集するために `/etc/haproxy/conf.d/20-vcac.cfg` を開きます。
- 2 フロント エンド セクションで、次の行を追加します。

```
rspdel Strict-Transport-Security:\ max-age=31536000
rspadd Strict-Transport-Security:\ max-age=31536000
```

- 3 次のコマンドを使用して、HAProxy 構成を再ロードします。

```
/etc/init.d/haproxy reload
```

X-Frame-Options 応答ヘッダーの構成

X-Frame-Options 応答ヘッダーは、場合によっては 2 回表示されることがあります。

X-Frame-Options 応答ヘッダーが 2 回表示されることがあるのは、vIDM サービスがこのヘッダーをバックエンドと HAProxy に追加するためです。適切な構成にすることで、2 回表示されることを防止できます。

手順

- 1 編集するために `/etc/haproxy/conf.d/20-vcac.cfg` を開きます。
- 2 フロント エンド セクションで、次の行を見つけます。

```
rspadd X-Frame-Options:\ SAMEORIGIN
```

- 3 上記の手順で見つけた行の前に、次の行を追加します。

```
rspdel X-Frame-Options:\ SAMEORIGIN
```

- 4 次のコマンドを使用して、HAProxy 構成を再ロードします。

```
/etc/init.d/haproxy reload
```

サーバ応答ヘッダーの構成

セキュリティのベストプラクティスとして、潜在的な攻撃者が使用可能な情報を制限するように vRealize Automation システムを構成します。

可能な範囲で、システムで共有する ID やバージョンに関する情報量を最小限にします。ハッカーや悪意のある攻撃者は、この情報を使用して Web サーバまたはバージョンを標的にした攻撃を作成することができます。

Lighttpd サーバ応答ヘッダーの構成

ベストプラクティスとして、vRealize Automation アプライアンス lighttpd サーバ用に空白のサーバヘッダーを作成します。

手順

- 1 `/opt/vmware/etc/lighttpd/lighttpd.conf` ファイルをテキストエディタで開きます。
- 2 `server.tag = " "` をファイルに追加します。
- 3 変更内容を保存し、ファイルを閉じます。
- 4 `# /opt/vmware/etc/init.d/vami-lighttpd restart` コマンドを実行して、lighttpd サーバを再起動します。

TCServer 応答ヘッダーの vRealize Automation アプライアンス向けの構成

ベストプラクティスとして、vRealize Automation アプライアンスと連携して使用する TCServer 応答ヘッダーのカスタムの空白サーバヘッダーを作成して、有益な情報を取得しようとする悪意のある攻撃の可能性を制限します。

手順

- 1 テキストエディタで `/etc/vco/app-server/server.xml` ファイルを開きます。
- 2 それぞれの `<Connector>` 要素に `server=" "` を追加します。
たとえば、`<Connector protocol="HTTP/1.1" server="" />` のように指定します。
- 3 変更内容を保存し、ファイルを閉じます。
- 4 次のコマンドを使用してサーバを再起動します。

```
service vco-server restart
```

Internet Information Services のサーバ応答ヘッダーの構成

ベストプラクティスとして、Identity Appliance と連携して使用する Internet Information Services (IIS) サーバのカスタムの空白サーバヘッダーを作成して、有益な情報を取得しようとする悪意のある攻撃の可能性を制限します。

手順

- 1 `C:\Windows\System32\inetsrv\urlscan\UrlScan.ini` ファイルをテキスト エディタで開きます。
- 2 `RemoveServerHeader=0` を検索し、`RemoveServerHeader=1.` に変更します。
- 3 変更内容を保存し、ファイルを閉じます。
- 4 `iisreset` コマンドを実行して、サーバを再起動します。

次に進む前に

IIS マネージャー コンソールで、リストから HTTP 応答ヘッダーを削除して、IIS X-Powered By ヘッダーを無効にします。

- 1 IIS マネージャー コンソールを開きます。
- 2 HTTP 応答ヘッダーを開き、リストから削除します。
- 3 `iisreset` コマンドを実行して、サーバを再起動します。

vRealize Automation アプライアンス セッション タイムアウトの設定

会社のセキュリティ ポリシーに従って、vRealize Automation アプライアンス でセッションのタイムアウト設定を構成します。

ユーザーによる操作がない場合の vRealize Automation アプライアンス のデフォルト セッション タイムアウトは 30 分です。組織のセキュリティ ポリシーに準拠するようにこのタイムアウト値を調整するには、vRealize Automation アプライアンス ホスト マシンの `web.xml` ファイルを編集します。

手順

- 1 テキスト エディタで `/usr/lib/vcac/server/webapps/vcac/WEB-INF/web.xml` ファイルを開きます。
- 2 `session-config` を特定して、セッション タイムアウト値を設定します。次のコード サンプルを参照してください。

```
<!-- 30 minutes session expiration time -->
<session-config>
  <session-timeout>30</session-timeout>
  <tracking-mode>COOKIE</tracking-mode>
  <cookie-config>
    <path>/</path>
  </cookie-config>
</session-config>
```

- 3 次のコマンドを実行して、サーバを再起動します。

```
service vcac-server restart
```

必須でないソフトウェアの管理

セキュリティ リスクを最小限に抑えるため、必須でないソフトウェアは vRealize Automation ホスト マシンから削除または構成します。

セキュリティ侵害を引き起こす可能性を最小限に抑えるため、メーカーの推奨事項とセキュリティのベスト プラクティスに従って削除しないすべてのソフトウェアを構成します。

USB 大容量ストレージ ハンドラのセキュリティ保護

VMware 仮想アプライアンス ホスト マシンで USB デバイス ハンドラとして使用されないように、USB 大容量ストレージ ハンドラをセキュリティ保護します。潜在的な攻撃者がこのハンドラを悪用してシステムに侵入する可能性があります。

手順

- 1 テキスト エディタで `/etc/modprobe.conf.local` ファイルを開きます。
- 2 このファイルに `install usb-storage /bin/true` 行が表示されていることを確認します。
- 3 ファイルを保存し、終了します。

Bluetooth プロトコル ハンドラのセキュリティ保護

潜在的な攻撃者の悪用を防ぐために、仮想アプライアンス ホスト マシンで Bluetooth プロトコル ハンドラをセキュリティ保護します。

Bluetooth プロトコルをネットワーク スタックにバインドすると、ホストの攻撃対象が増える可能性があります。

手順

- 1 テキスト エディタで `/etc/modprobe.conf.local` ファイルを開きます。
- 2 このファイルに次の行が表示されていることを確認します。
`install bluetooth /bin/true`
- 3 ファイルを保存し、終了します。

Stream Control Transmission Protocol のセキュリティ保護

Stream Control Transmission Protocol (SCTP) がデフォルトでシステムにロードされないようにします。潜在的な攻撃者がこのプロトコルを悪用してシステムに侵入する可能性があります。

本当に必要ない限り、Stream Control Transmission Protocol (SCTP) モジュールがロードされないようにシステムを構成します。SCTP は、未使用の IETF 標準化トランスポート レイヤー プロトコルです。このプロトコルをネットワーク スタックにバインドすると、ホストの攻撃対象が増えます。ローカル プロセスに権限を付与しないと、カーネルがプロトコルを使用してソケットを開くことで、プロトコル ハンドラを動的にロードする可能性があります。

手順

- 1 テキスト エディタで `/etc/modprobe.conf.local` ファイルを開きます。

- 2 このファイルに次の行が表示されていることを確認します。

```
install sctp /bin/true
```

- 3 ファイルを保存し、終了します。

Datagram Congestion Protocol のセキュリティ保護

システムのセキュリティ強化策の一環として、Datagram Congestion Protocol (DCCP) がデフォルトで仮想アプライアンス ホスト マシンにロードされないようにします。潜在的な攻撃者がこのプロトコルを悪用してシステムに侵入する可能性があります。

本当に必要ない限りは、Datagram Congestion Control Protocol (DCCP) モジュールをロードしないでください。DCCP はトランスポート レイヤー プロトコルとして予約されていますが、使用されてはいません。このプロトコルをネットワーク スタックにバインドすると、ホストの攻撃対象が増えます。ローカル プロセスに権限を付与しないと、カーネルがソケットを開くプロトコルを使用してプロトコル ハンドラを動的にロードする可能性があります。

手順

- 1 テキスト エディタで `/etc/modprobe.conf.local` ファイルを開きます。
- 2 このファイルに DCCP 行が表示されていることを確認します。

```
install dccp/bin/true
install dccp_ipv4/bin/true
install dccp_ipv6/bin/true
```

- 3 ファイルを保存し、終了します。

ネットワーク ブリッジのセキュリティ保護

ネットワーク ブリッジ モジュールがデフォルトでシステムにロードされないようにします。潜在的な攻撃者がこのモジュールを悪用してシステムに侵入する可能性があります。

本当に必要ない限り、ネットワーク ブリッジ モジュールがロードされないようにシステムを構成します。潜在的な攻撃者がネットワーク パーティションおよびセキュリティをバイパスすることでこのモジュールを悪用する可能性があります。

手順

- 1 すべての VMware アプライアンス ホスト マシンで次のコマンドを実行します。

```
# rmmod bridge
```

- 2 テキスト エディタで `/etc/modprobe.conf.local` ファイルを開きます。
- 3 このファイルに次の行が表示されていることを確認します。

```
install bridge /bin/false
```

- 4 ファイルを保存し、終了します。

Reliable Datagram Socket プロトコルのセキュリティ保護

システムのセキュリティ強化策の一環として、Reliable Datagram Socket プロトコル (RDS) がデフォルトで仮想アプライアンス ホスト マシンにロードされないようにします。潜在的な攻撃者がこのプロトコルを悪用してシステムに侵入する可能性があります。

Reliable Datagram Socket (RDS) プロトコルをネットワーク スタックにバインドすると、ホストの攻撃対象が増えます。ローカル プロセスに権限を付与しないと、システムがソケットを開くプロトコルを使用してプロトコル ハンドラを動的にロードする可能性があります。

手順

- 1 テキスト エディタで `/etc/modprobe.conf.local` ファイルを開きます。
- 2 このファイルに `install rds /bin/true` 行が表示されていることを確認します。
- 3 ファイルを保存し、終了します。

Transparent Inter-process Communication プロトコルのセキュリティ保護

システムのセキュリティ強化策の一環として、Transparent Inter-Process Communication (TIPC) プロトコルがデフォルトで仮想アプライアンス ホスト マシンにロードされないようにします。潜在的な攻撃者がこのプロトコルを悪用してシステムに侵入する可能性があります。

Transparent Inter-Process Communications (TIPC) プロトコルをネットワーク スタックにバインドすると、ホストの攻撃対象が増えます。ローカル プロセスに権限を付与しないと、カーネルがソケットを開くプロトコルを使用してプロトコル ハンドラを動的にロードする可能性があります。

手順

- 1 テキスト エディタで `/etc/modprobe.conf.local` ファイルを開きます。
- 2 このファイルに `install tipc /bin/true` 行が表示されていることを確認します。
- 3 ファイルを保存し、終了します。

Internetwork Packet Exchange プロトコルのセキュリティ保護

Internetwork Packet Exchange (IPX) プロトコルがデフォルトでシステムにロードされないようにします。潜在的な攻撃者がこのプロトコルを悪用してシステムに侵入する可能性があります。

本当に必要ない限りは、Internetwork Packet Exchange (IPX) プロトコル モジュールをロードしないでください。IPX プロトコルは廃止されたネットワーク レイヤー プロトコルです。このプロトコルをネットワーク スタックにバインドすると、ホストの攻撃対象が増えます。ローカル プロセスに権限を付与しないと、システムがソケットを開くプロトコルを使用してプロトコル ハンドラを動的にロードする可能性があります。

手順

- 1 テキスト エディタで `/etc/modprobe.conf.local` ファイルを開きます。

- 2 このファイルに次の行が表示されていることを確認します。

```
install ipx /bin/true
```

- 3 ファイルを保存し、終了します。

Appletalk プロトコルのセキュリティ保護

Appletalk プロトコルがデフォルトでシステムにロードされないようにします。潜在的な攻撃者がこのプロトコルを悪用してシステムに侵入する可能性があります。

本当に必要な限りは、Appletalk プロトコル モジュールをロードしないでください。このプロトコルをネットワーク スタックにバインドすると、ホストの攻撃対象が増えます。ローカル プロセスに権限を付与しないと、システムがソケットを開くプロトコルを使用してプロトコル ハンドラを動的にロードする可能性があります。

手順

- 1 テキスト エディタで `/etc/modprobe.conf.local` ファイルを開きます。
- 2 このファイルに次の行が表示されていることを確認します。

```
install appletalk /bin/true
```

- 3 ファイルを保存し、終了します。

DECnet プロトコルのセキュリティ保護

DECnet プロトコルがデフォルトでシステムにロードされないようにします。潜在的な攻撃者がこのプロトコルを悪用してシステムに侵入する可能性があります。

本当に必要な限りは、DECnet プロトコル モジュールをロードしないでください。このプロトコルをネットワーク スタックにバインドすると、ホストの攻撃対象が増えます。ローカル プロセスに権限を付与しないと、システムがソケットを開くプロトコルを使用してプロトコル ハンドラを動的にロードする可能性があります。

手順

- 1 テキスト エディタで `/etc/modprobe.conf.local` ファイルを開きます。
- 2 このファイルに次の行が表示されていることを確認します。

```
install decnet /bin/true
```

- 3 ファイルを保存し、終了します。

Firewire モジュールのセキュリティ保護

Firewire モジュールがデフォルトでシステムにロードされないようにします。潜在的な攻撃者がこのプロトコルを悪用してシステムに侵入する可能性があります。

本当に必要な限りは、Firewire モジュールをロードしないでください。

手順

- 1 テキスト エディタで `/etc/modprobe.conf.local` ファイルを開きます。

- 2 このファイルに次の行が表示されていることを確認します。

```
install ieee1394 /bin/true
```

- 3 ファイルを保存し、終了します。

Infrastructure as a Service (IaaS) コンポーネントのセキュリティ保護

システムをセキュリティ強化する場合は、vRealize Automation IaaS コンポーネントとそのホスト マシンをセキュリティ保護し、潜在的な攻撃者が悪用できないようにします。

vRealize Automation IaaS コンポーネントとそのコンポーネントが格納されているホストのセキュリティ設定を構成する必要があります。他の関連コンポーネントとアプリケーションの構成を設定または確認する必要があります。場合によっては、既存の設定を確認できます。他の場合には、適切な構成になるように設定を変更または追加する必要があります。

Windows Time サービスを無効にする

セキュリティのベスト プラクティスとして、vRealize Automation 本番環境では、ホストの時刻同期ではなく、認証されたタイム サーバを使用します。

本番環境では、ホストの時刻同期を無効にし、認証されたタイム サーバを使用して、監査とログを通じたユーザーアクションの正確な追跡および潜在的な悪意のある攻撃と侵入の特定をサポートします。

転送中の Infrastructure as a Service (IaaS) データの TLS の構成

IaaS コンポーネントの転送チャネルをセキュリティ保護するために、vRealize Automation デプロイで強力な TLS プロトコルを使用していることを確認します。

Secure Sockets Layer (SSL) や、より新しく開発された Transport Layer Security (TLS) は、異なるシステム コンポーネント間のネットワーク通信中にシステムのセキュリティを確保するために役立つ暗号プロトコルです。SSL はより古い標準であるため、その実装の多くは、潜在的な攻撃に対して適切なセキュリティを提供できなくなっています。SSLv2 および SSLv3 を含む、以前の SSL プロトコルには、重大な脆弱性が見つかっています。これらのプロトコルは、もう安全であるとは認識されません。

組織のセキュリティ ポリシーによっては、TLS 1.0 も無効にする必要がある場合があります。

注意 ロード バランサで TLS を終了する場合は、SSLv2 や SSLv3 などの強力でないプロトコルに加えて、必要に応じて TLS 1.0 も無効にします。

Internet Information Services で SSLv3 を無効にする

セキュリティのベスト プラクティスとして、Infrastructure as a Service (IaaS) ホスト サーバ マシンの Internet Information Services (IIS) では、SSLv3 を無効にします。

手順

- 1 管理者として Windows レジストリ エディタを実行します。

- 2 レジストリ ウィンドウで
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\SecurityProviders\Schannel\Protocols\ に移動します。
- 3 [Protocols] を右クリックし、[新規] - [キー] の順に選択します。
- 4 **SSL 3.0** と入力します。
- 5 ナビゲーションツリーで、新しく作成した [SSL 3.0] キーを右クリックし、ポップアップメニューで [新規] - [キー] の順に選択して **Client** と入力します。
- 6 ナビゲーションツリーで、新しく作成した [SSL 3.0] キーを右クリックし、ポップアップメニューで [新規] - [キー] の順に選択して **Server** と入力します。
- 7 ナビゲーションツリーで、SSL 3.0 の下の [Client] を右クリックし、[新規] - [DWORD (32 ビット) 値] の順に選択して、**DisabledByDefault** と入力します。
- 8 ナビゲーションツリーの SSL 3.0 の下の [Client] を選択し、右側のペインで [DisabledByDefault] をダブルクリックして **1** と入力します。
- 9 ナビゲーションツリーで、SSL 3.0 の下の [Server] を右クリックし、[新規] - [DWORD (32 ビット) 値] の順に選択して **Enabled** と入力します。
- 10 ナビゲーションツリーで、SSL 3.0 の下の [Server] を選択し、右側のペインで有効になっている [DWORD] をダブルクリックして **0** と入力します。
- 11 Windows Server を再起動します。

laaS の TLS 1.0 を無効にする

最大限のセキュリティを提供するには、プールを使用するように laaS を構成して TLS 1.0 を無効にします。

詳細については、<https://support.microsoft.com/en-us/kb/245030> の Microsoft ナレッジベースの記事を参照してください。

手順

- 1 Web ソケットの代わりにプールを使用するように laaS を構成します。
 - a <appSettings> セクションに次の値を追加することで、Manager Service 構成ファイル (C:\Program Files (x86)\VMware\VCAC\Server\ManagerService.exe.config) を更新します。


```
<add key="Extensibility.Client.RetrievalMethod" value="Polling"/>
<add key="Extensibility.Client.PollingInterval" value="2000"/>
<add key="Extensibility.Client.PollingMaxEvents" value="128"/>
```
 - b Manager Service (VMware vCloud Automation Center Service) を再起動します。

2 IaaS サーバで TLS 1.0 が無効になっていることを確認します。

- a 管理者として、レジストリ エディタを実行します。
- b レジストリ ウィンドウで、
`HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\SecurityProviders\Schannel\Protocols\` に移動します。
- c Protocols を右クリックし、[新規] - [キー] の順に選択して **TLS 1.0** と入力します。
- d ナビゲーション ツリーで、新しく作成した TLS 1.0 キーを右クリックし、ポップアップ メニューで [新規] - [キー] の順に選択して **Client** と入力します。
- e ナビゲーション ツリーで、新しく作成した TLS 1.0 キーを右クリックし、ポップアップ メニューで [新規] - [キー] の順に選択して **Server** と入力します。
- f ナビゲーション ツリーで、TLS 1.0 の下の [Client] を右クリックし、[新規] - [DWORD (32 ビット) 値] の順にクリックして、**DisabledByDefault** と入力します。
- g ナビゲーション ツリーの TLS 1.0 の下の [Client] を選択し、右側のペインで [DisabledByDefault] の DWORD をダブルクリックして **1** と入力します。
- h ナビゲーション ツリーで、TLS 1.0 の下の [Server] を右クリックし、[新規] - [DWORD (32 ビット) 値] の順に選択して **Enabled** と入力します。
- i ナビゲーション ツリーの TLS 1.0 の下の [Server] を選択し、右側のペインで [Enabled] の DWORD をダブルクリックして **0** と入力します。
- j Windows Server を再起動します。

TLS 暗号スイートの構成

セキュリティを最大限高めるには、強力な暗号を使用するように vRealize Automation コンポーネントを設定する必要があります。サーバとブラウザの間でネゴシエートされる暗号化により、TLS セッションで使用される暗号化の強度が決まります。強力な暗号のみが確実に選択されるようにするため、vRealize Automation コンポーネントで強度の弱い暗号を無効にします。強力な暗号のみをサポートし、十分に大きいキー サイズを使用するように、サーバを設定します。また、適切な順序ですべての暗号を構成します。

使用できない暗号スイート

NULL 暗号スイート、aNULL、または eNULL などの認証を提供しない暗号スイートを無効にします。この他に、匿名の Diffie-Hellman キー交換 (ADH)、エクスポート レベルの暗号 (EXP、DES を含んでいる暗号)、ペイロードトラフィックの暗号化に対する 128 ビットより小さいサイズのキー、ペイロードトラフィックのハッシュ メカニズムとしての MD5 の使用、IDEA 暗号スイート、および RC4 暗号スイートも無効にします。また、Diffie-Hellman (DHE) キー交換を使用する暗号スイートが無効になっていることを確認します。

ホスト サーバのセキュリティの確認

セキュリティのベスト プラクティスとして、Infrastructure as a Service (IaaS) ホスト サーバ マシンのセキュリティ構成を確認します。

Microsoft は、ホスト サーバ マシンのセキュリティの確認に役立ついくつかのツールを提供しています。これらのツールの適切な使用方法については、Microsoft の担当者にお問い合わせください。

ホスト サーバのセキュアなベースラインの確認

Microsoft Baseline Security Analyzer (MBSA) を実行すると、サーバに最新の更新またはホット フィックスが適用されていることを簡単に確認できます。MBSA を使用すると、未適用の Microsoft のセキュリティ パッチをインストールし、Microsoft のセキュリティ推奨事項に関してサーバを最新の状態に維持することができます。

Microsoft の Web サイトから最新版の MBSA ツールをダウンロードします。

ホスト サーバのセキュリティ構成の確認

Windows セキュリティの構成ウィザード (SCW) および Microsoft Security Compliance Manager (SCM) ツールキットを使用して、ホスト サーバが安全に構成されていることを確認します。

Windows サーバの管理ツールから SCW を実行します。このツールにより、サーバのロールと、ネットワーク、Windows ファイアウォール、レジストリ設定などのインストールされている機能を特定できます。Windows サーバに関連する SCM の最新のセキュリティ強化ガイダンスとこのレポートを比較します。結果に基づいて、ネットワーク サービス、アカウント設定、および Windows ファイアウォールなど、各機能のセキュリティ設定を調整して、この設定をサーバに適用します。

SCW ツールの詳細については、Microsoft Technet の Web サイトを参照してください。

アプリケーション リソースの保護

セキュリティのベスト プラクティスとして、関連するすべての Infrastructure as a Service (IaaS) ファイルに適切な権限があることを確認します。

IaaS ファイルを IaaS インストールと突き合わせて確認します。ほとんどの場合、すべてのフォルダのサブフォルダとファイルはフォルダと同じ設定にする必要があります。

ディレクトリまたはファイル	グループまたはユーザー	読み取りおよび実行				
		完全制御	変更	読み取り	書き込み	
VMware\VCAC\Agents\<agent_name>\logs	システム	X	X	X	X	X
	管理者	X	X	X	X	X
	管理者	X	X	X	X	X
VMware\VCAC\Agents\<agent_name>\temp	システム	X	X	X	X	X
	管理者	X	X	X	X	X
	管理者	X	X	X	X	X
VMware\VCAC\Agents\	システム	X	X	X	X	X
	管理者	X	X	X	X	X
	ユーザー			X	X	
VMware\VCAC\Distributed Execution Manager\	システム	X	X	X	X	X
	管理者	X	X	X	X	X
	ユーザー			X	X	

ディレクトリまたはファイル	グループまたはユーザー	読み取りおよび実行				
		完全制御	変更	読み取り	書き込み	
VMware\vmcac\Distributed Execution Manager\DEM\Logs	システム	X	X	X	X	X
	管理者	X	X	X	X	X
	管理者	X	X	X	X	X
VMware\vmcac\Distributed Execution Manager\DEO\Logs	システム	X	X	X	X	X
	管理者	X	X	X	X	X
	管理者	X	X	X	X	X
VMware\vmcac\Management Agent\	システム	X	X	X	X	X
	管理者	X	X	X	X	X
	ユーザー			X	X	
VMware\vmcac\Server\	システム	X	X	X	X	X
	管理者	X	X	X	X	X
	ユーザー			X	X	
VMware\vmcac\Web API	システム	X	X	X	X	X
	管理者	X	X	X	X	X
	ユーザー			X	X	

Infrastructure as a Service (IaaS) ホスト マシンのセキュリティ保護

セキュリティのベスト プラクティスとして、IaaS ホスト マシンの基本設定を見直して、セキュリティ ガイドラインに準拠していることを確認します。

IaaS ホスト マシンのアカウント、アプリケーション、ポート、およびサービスなどを保護します。

サーバユーザー アカウントの設定の確認

不要なローカル ユーザーおよびドメイン ユーザーのアカウントや設定がないことを確認します。アプリケーションの機能に関連していないユーザー アカウントを制限し、管理、メンテナンス、およびトラブルシューティングのために必要なものだけにします。ドメイン ユーザー アカウントからのリモート アクセスを、サーバの保守に必要な最低限のものに制限します。これらのアカウントを厳密に制御して監査します。

不要なアプリケーションの削除

ホスト サーバからすべての不要なアプリケーションを削除します。不要なアプリケーションがあることで、十分な対策が取られていない未知の脆弱性を原因とする漏洩のリスクが増大します。

不要なポートおよびサービスを無効にする

開いているポートのリストについて、ホスト サーバのファイアウォールを確認します。IaaS コンポーネントまたは重大なシステム操作に必要なではないすべてのポートをブロックします。[「ポートおよびプロトコルの構成」](#) を参照してください。ホスト サーバに対して実行中のサービスを監査し、不要なものを無効にします。

ホストのネットワーク セキュリティの構成

8

既知のセキュリティ脅威に対する最大限の保護を提供するために、すべての VMware ホスト マシンでネットワーク インターフェイスと通信設定を構成します。

包括的なセキュリティ プランの一環として、確立されているセキュリティ ガイドラインに従って、VMware 仮想アプライアンスと Infrastructure as a Service (IaaS) コンポーネントのネットワーク インターフェイスのセキュリティ設定を構成します。

この章では次のトピックについて説明します。

- [VMware アプライアンスのネットワーク設定の構成](#)
- [Infrastructure as a Service \(IaaS\) ホストのネットワーク設定の構成](#)
- [ポートおよびプロトコルの構成](#)

VMware アプライアンスのネットワーク設定の構成

VMware 仮想アプライアンス ホスト マシンで、安全で不可欠な通信のみが確実にサポートされるようにするために、それらのマシンのネットワーク通信設定を確認して編集します。

VMware ホスト マシンのネットワークの IP プロトコル構成を確認し、セキュリティのガイドラインに従ってネットワークの設定を構成します。すべての必須でない通信プロトコルを無効にします。

ネットワーク インターフェイスのユーザーによるコントロールの制限

セキュリティのベスト プラクティスとして、ユーザーには VMware アプライアンス ホスト マシンでジョブを実行するために必要なシステム権限のみを許可します。

ネットワーク インターフェイスを操作する権限をユーザー アカウントに許可すると、ネットワーク セキュリティ メカニズムをバイパスしたり、サービス拒否が発生したりすることがあります。許可されたユーザーだけがネットワーク インターフェイス設定を変更できるようにします。

手順

- 1 各 VMware アプライアンス ホスト マシンで、次のコマンドを実行します。

```
# grep -i '^USERCONTROL=' /etc/sysconfig/network/ifcfg*
```

- 2 各インターフェイスが **NO** に設定されていることを確認します。

TCP バックログのキュー サイズの設定

悪意のある攻撃に対して一定レベルの防御を実現するには、VMware アプライアンス ホスト マシンでデフォルトの TCP バックログのキュー サイズを構成します。

TCP のサービス拒否攻撃を軽減するために、TCP バックログのキュー サイズを適切なデフォルト サイズに設定します。推奨デフォルト設定は 1280 です。

手順

- 1 各 VMware アプライアンス ホスト マシンで次のコマンドを実行します。

```
# cat /proc/sys/net/ipv4/tcp_max_syn_backlog
```

- 2 テキスト エディタで **/etc/sysctl.conf** ファイルを開きます。

- 3 次のエントリをファイルに追加することで、デフォルトの TCP バックログのキュー サイズを設定します。

```
net.ipv4.tcp_max_syn_backlog=1280
```

- 4 変更内容を保存し、ファイルを閉じます。

ブロードキャスト アドレスへの ICMPv4 エコーの拒否

セキュリティのベスト プラクティスとして、VMware アプライアンス ホスト マシンが ICMP ブロードキャスト アドレスのエコー要求を無視することを確認します。

ブロードキャスト Internet Control Message Protocol (ICMP) エコーへの応答は、増幅攻撃に対する攻撃ベクトルを提供し、悪意のあるエージェントによるネットワーク マッピングの利用を容易にする恐れがあります。ICMPv4 エコーを無視するようにアプライアンス ホスト マシンを構成することで、このような攻撃に対する保護を提供します。

手順

- 1 IPv4 ブロードキャスト アドレスのエコー要求を拒否することを確認するために、VMware 仮想アプライアンス ホスト マシンで **# cat /proc/sys/net/ipv4/icmp_echo_ignore_broadcasts** コマンドを実行します。

IPv4 リダイレクトを拒否するようにホスト マシンが構成されている場合、このコマンドは **/proc/sys/net/ipv4/icmp_echo_ignore_broadcasts** に対して 0 の値を返します。

- 2 ICMPv4 ブロードキャスト アドレスのエコー要求を拒否するように仮想アプライアンス ホスト マシンを構成するには、Windows ホスト マシンで、**/etc/sysctl.conf** ファイルをテキスト エディタで開きます。

- 3 **net.ipv4.icmp_echo_ignore_broadcasts=0** というエントリを探します。このエントリの値がゼロに設定されていない場合、またはエントリがない場合は、追加するか、既存のエントリを更新します。

- 4 変更内容を保存し、ファイルを閉じます。

IPv4 プロキシ ARP を無効にする

VMware アプライアンス ホスト マシンで別途必要のない限り、IPv4 プロキシ ARP が無効になっていることを確認して不正な情報共有を防ぎます。

IPv4 プロキシ ARP を使用すると、別のインターフェイスに接続されているホストの代わりに、特定のインターフェイスの ARP 要求にシステムから応答を送信できます。接続されているネットワーク セグメント間でアドレス情報の漏えいを防ぐため、必要でない場合は無効にします。

手順

- 1 IPv4 プロキシ ARP が無効になっていることを確認するために、VMware 仮想アプライアンス ホスト マシンで **# grep [01] /proc/sys/net/ipv4/conf/*/proxy_arp|egrep "default|all"** コマンドを実行します。

ホスト マシンで IPv6 プロキシ ARP が無効になっている場合、このコマンドは 0 の値を返します。

```
/proc/sys/net/ipv4/conf/all/proxy_arp:0
/proc/sys/net/ipv4/conf/default/proxy_arp:0
```

ホスト マシンが正しく構成されている場合は、これ以上の操作は必要ありません。

- 2 ホスト マシンで IPv6 プロキシ ARP を構成する必要がある場合は、**/etc/sysctl.conf** ファイルをテキストエディタで開きます。
- 3 次のエントリを確認します。

```
net.ipv4.conf.default.proxy_arp=0
net.ipv4.conf.all.proxy_arp=0
```

エントリがない場合、またはそれらの値がゼロに設定されていない場合は、エントリを追加するか、既存のエントリを更新します。

- 4 変更内容を保存し、ファイルを閉じます。

IPv4 ICMP リダイレクト メッセージの拒否

セキュリティのベスト プラクティスとして、VMware 仮想アプライアンス ホスト マシンが、IPv4 ICMP リダイレクト メッセージを拒否することを確認します。

ルーターでは、ICMP リダイレクト メッセージを使用して、ターゲットに対するより直接的なルートがあることをホストに通知します。悪意のある ICMP リダイレクト メッセージによって、中間者攻撃が行われる恐れがあります。これらのメッセージは、ホストのルート テーブルを変更することで、認証されない状態になります。システムで必要な場合を除いて、これらのメッセージを無視するようにシステムが構成されていることを確認します。

手順

- 1 IPv4 リダイレクト メッセージを拒否することを確認するために、VMware アプライアンス ホスト マシンで **# grep [01] /proc/sys/net/ipv4/conf/*/accept_redirects|egrep "default|all"** コマンドを実行します。

IPv4 リダイレクトを拒否するようにホスト マシンが構成されている場合、このコマンドは次を返します。

```
/proc/sys/net/ipv4/conf/all/accept_redirects:0
/proc/sys/net/ipv4/conf/default/accept_redirects:0
```

- 2 IPv4 リダイレクト メッセージを拒否するように仮想アプライアンス ホスト マシンを構成する必要がある場合には、テキスト エディタで **/etc/sysctl.conf** ファイルを開きます。
- 3 **net.ipv4.conf** で始まる行の値を確認します。

次のエントリの値がゼロに設定されていない場合、またはこれらのエントリがない場合は、ファイルに追加するか既存のエントリを更新します。

```
net.ipv4.conf.all.accept_redirects=0
net.ipv4.conf.default.accept_redirects=0
```

- 4 変更内容を保存し、ファイルを閉じます。

IPv6 ICMP リダイレクト メッセージの拒否

セキュリティのベスト プラクティスとして、VMware 仮想アプライアンス ホスト マシンが IPv6 ICMP リダイレクト メッセージを拒否することを確認します。

ルーターでは、ICMP リダイレクト メッセージを使用して、ターゲットに対するより直接的なルートがあることをホストに通知します。悪意のある ICMP リダイレクト メッセージによって、中間者攻撃が行われる恐れがあります。これらのメッセージは、ホストのルート テーブルを変更することで、認証されない状態になります。必要な場合を除き、これらを見逃すようにシステムが構成されていることを確認します。

手順

- 1 IPv6 リダイレクト メッセージを拒否することを確認するために、VMware 仮想アプライアンス ホスト マシンで **# grep [01] /proc/sys/net/ipv6/conf/*/accept_redirects | egrep "default|all"** コマンドを実行します。

ホスト マシンが IPv6 リダイレクトを拒否するように構成されている場合、このコマンドは次を返します。

```
/proc/sys/net/ipv6/conf/all/accept_redirects:0
/proc/sys/net/ipv6/conf/default/accept_redirects:0
```

- 2 IPv4 リダイレクト メッセージを拒否するように仮想アプライアンス ホスト マシンを構成するには、テキスト エディタで **/etc/sysctl.conf** ファイルを開きます。
- 3 **net.ipv6.conf** で始まる行の値を確認します。

次のエントリの値がゼロに設定されていない場合、またはこれらのエントリがない場合は、ファイルに追加するか既存のエントリを更新します。

```
net.ipv6.conf.all.accept_redirects=0
net.ipv6.conf.default.accept_redirects=0
```

- 4 変更内容を保存し、ファイルを閉じます。

IPv4 Martian パケットのログ

セキュリティのベスト プラクティスとして、VMware 仮想アプライアンス ホスト マシンが IPv4 Martian パケットをログに記録することを確認します。

Martian パケットには、無効になるとシステムが認識しているアドレスが含まれています。これらのメッセージをログに記録し、不適切な構成や進行中の攻撃を特定できるようにホスト マシンを構成します。

手順

- 1 IPv4 Martian パケットをログ記録していることを確認するために、VMware アプライアンス ホスト マシンで **# grep [01] /proc/sys/net/ipv4/conf/*/log_martians|egrep "default|all"** コマンドを実行します。

Martian パケットをログ記録するように仮想マシンが構成されている場合は、次を返します。

```
/proc/sys/net/ipv4/conf/all/log_martians:1
/proc/sys/net/ipv4/conf/default/log_martians:1
```

ホスト マシンが正しく構成されている場合は、これ以上の操作は必要ありません。

- 2 IPv4 Martian パケットをログ記録するように仮想マシンを構成する必要がある場合は、テキスト エディタで **/etc/sysctl.conf** ファイルを開きます。
- 3 **net.ipv4.conf** で始まる行の値を確認します。

次のエントリの値が 1 に設定されていない場合、またはこれらのエントリがない場合は、ファイルに追加するか既存のエントリを更新します。

```
net.ipv4.conf.all.log_martians=1
net.ipv4.conf.default.log_martians=1
```

- 4 変更内容を保存し、ファイルを閉じます。

IPv4 リバース パス フィルタリングの使用

セキュリティのベスト プラクティスとして、IPv4 リバース パス フィルタリングが VMware 仮想アプライアンス ホスト マシンで使用されていることを確認します。

リバース パス フィルタリングは、ソース アドレスにルートがないパケットや、ソース アドレスのルートが発信元のインターフェイスをポイントしていないパケットがシステムで破棄されるようにすることで、偽装されたソース アドレスに対する保護を行います。可能な限りリバース パス フィルタリングを使用するようにホスト マシンを構成します。システムの役割によっては、リバース パス フィルタリングが、システムで正規のトラフィックが破棄される原因となる場合があります。このような問題が発生する場合、より寛容なモードを使用するか、リバース パス フィルタリングを完全に無効にすることが必要になる可能性があります。

手順

- 1 IPv4 リバース パス フィルタリングを使用していることを確認するために、VMware 仮想アプライアンス ホスト マシンで **# grep [01] /proc/sys/net/ipv4/conf/*/rp_filter|egrep "default|all"** コマンドを実行します。

仮想マシンで IPv4 リバース パス フィルタリングを使用している場合、このコマンドは次を返します。

```
/proc/sys/net/ipv4/conf/all/rp_filter:1
/proc/sys/net/ipv4/conf/default/re_filter:1
```

仮想マシンが正しく構成されている場合は、これ以上の操作は必要ありません。

- 2 IPv4 リバース パス フィルタリングをホスト マシンに構成する必要がある場合は、テキスト エディタで **/etc/sysctl.conf** ファイルを開きます。
- 3 **net.ipv4.conf** で始まる行の値を確認します。

次のエントリの値が 1 に設定されていない場合、またはこれらのエントリがない場合は、ファイルに追加するか既存のエントリを更新します。

```
net.ipv4.conf.all.rp_filter=1
net.ipv4.conf.default.rp_filter=1
```

- 4 変更内容を保存し、ファイルを閉じます。

IPv4 転送の拒否

VMware アプライアンス ホスト マシンが IPv4 転送を拒否することを確認します。

システムが IP 転送向けに構成されていて、指定されたルーターではない場合、攻撃者は、このシステムを利用して、ネットワーク デバイスでフィルタされていない通信のパスを提供することでネットワーク セキュリティをバイパスする可能性があります。このリスクを回避するために、IPv4 転送を拒否するように仮想アプライアンス ホスト マシンを構成します。

手順

- 1 IPv4 転送を拒否することを確認するために、VMware アプライアンス ホスト マシンで **# cat /proc/sys/net/ipv4/ip_forward** コマンドを実行します。

IPv4 転送を拒否するようにホスト マシンが構成されている場合、このコマンドは **/proc/sys/net/ipv4/ip_forward** に対して 0 の値を返します。仮想マシンが正しく構成されている場合は、これ以上の操作は必要ありません。

- 2 IPv4 転送を拒否するように仮想アプライアンス ホスト マシンを構成するには、テキスト エディタで **/etc/sysctl.conf** ファイルを開きます。
- 3 **net.ipv4.ip_forward=0** というエントリを探します。このエントリの値がゼロに設定されていない場合、またはエントリがない場合は、追加するか、既存のエントリを更新します。
- 4 変更内容を保存し、ファイルを閉じます。

IPv6 転送の拒否

セキュリティのベスト プラクティスとして、VMware アプライアンスのホストシステムが IPv6 転送を拒否することを確認します。

システムが IP 転送向けに構成されていて、指定されたルーターではない場合、攻撃者は、このシステムを利用して、ネットワーク デバイスでフィルタされていない通信のパスを提供することでネットワーク セキュリティをバイパスする可能性があります。このリスクを回避するために、IPv6 転送を拒否するように仮想アプライアンス ホスト マシンを構成します。

手順

- 1 IPv6 転送を拒否することを確認するために、VMware アプライアンス ホスト マシンで **# grep**
[01] /proc/sys/net/ipv6/conf/*/forwarding|egrep "default|all" コマンドを実行します。

IPv6 転送を拒否するようにホスト マシンが構成されている場合、このコマンドは次のように応答します。

```
/proc/sys/net/ipv6/conf/all/forwarding:0  
/proc/sys/net/ipv6/conf/default/forwarding:0
```

ホスト マシンが正しく構成されている場合は、これ以上の操作は必要ありません。

- 2 IPv6 転送を拒否するようにホスト マシンを構成する必要がある場合は、**/etc/sysctl.conf** ファイルをテキスト エディタで開きます。
- 3 **net.ipv6.conf** で始まる行の値を確認します。

次のエントリの値がゼロに設定されていない場合、またはこれらのエントリがない場合は、エントリを追加するか既存のエントリを更新します。

```
net.ipv6.conf.all.accept_redirects=0  
net.ipv6.conf.default.accept_redirects=0
```

- 4 変更内容を保存し、ファイルを閉じます。

IPv4 TCP Syncookie の使用

VMware アプライアンス ホスト マシンが IPv4 TCP Syncookie を使用していることを確認します。

TCP SYN フラッド攻撃では、システムの TCP 接続テーブルを SYN_RCVD 状態の接続で満たしてサービス拒否の状態にできる場合があります。Syncookie を使用すると、後続の ACK を受信するまで接続を追跡できないため、インシエータによる接続が有効であり、フラッドソースではないことを確認します。この方法は完全に標準に準拠した方法では動作せず、フラッド状態でのみ作動し、有効な要求を処理しながらシステムを保護できます。

手順

- 1 IPv4 TCP Syncookie を使用していることを確認するために、VMware アプライアンス ホスト マシンで **# cat /proc/sys/net/ipv4/tcp_syncookies** コマンドを実行します。

ホスト マシンが IPv4 転送を拒否するように構成されている場合、**/proc/sys/net/ipv4/tcp_syncookies** コマンドは値 1 を返します。仮想マシンが正しく構成されている場合は、これ以上の操作は必要ありません。

- 2 IPv4 TCP Syncookie を使用するように仮想アプライアンスを構成する必要がある場合は、テキスト エディタで **/etc/sysctl.conf** を開きます。

- 3 **net.ipv4.tcp_syncookies=1** というエントリを探します。

このエントリの値が 1 に現在設定されていない場合、またはこのエントリがない場合は、追加するか、既存のエントリを更新します。

- 4 変更内容を保存し、ファイルを閉じます。

IPv6 ルーター通知の拒否

VMware ホスト マシンが、ルーター通知および ICMP リダイレクトの受け取りを拒否することを確認します（システム操作で必要でない場合）。

IPv6 では、ネットワークからの情報を自動的に使用してシステムがネットワーク デバイスを構成することができます。セキュリティの観点から、重要な構成情報は、認証されていない方法でネットワークから受け入れるよりも、手動で構成することをお勧めします。

手順

- 1 ルーター通知を拒否することを確認するために、VMware アプライアンス ホスト マシンで **# grep [01] /proc/sys/net/ipv6/conf/*/accept_ra | egrep "default|all"** コマンドを実行します。

IPv6 ルーター通知を拒否するようにホスト マシンが構成されている場合、このコマンドは 0 の値を返します。

```
/proc/sys/net/ipv6/conf/all/accept_ra:0
/proc/sys/net/ipv6/conf/default/accept_ra:0
```

ホスト マシンが正しく構成されている場合は、これ以上の操作は必要ありません。

- 2 IPv6 ルーター通知を拒否するようにホスト マシンを構成する必要がある場合は、**/etc/sysctl.conf** ファイルをテキスト エディタで開きます。

- 3 次のエントリを確認します。

```
net.ipv6.conf.all.accept_ra=0
net.ipv6.conf.default.accept_ra=0
```

これらのエントリがない場合、またはそれらの値がゼロに設定されていない場合は、エントリを追加するか、既存のエントリを更新します。

- 4 変更内容を保存し、ファイルを閉じます。

IPv6 ルーター要請の拒否

セキュリティのベスト プラクティスとして、VMware アプライアンス ホスト マシンが、IPv6 ルーター要請を拒否することを確認します（システム操作で必要でない場合）。

ルーター要請設定は、インターフェイスを構築するときに送信するルーター要請の数を決定します。アドレスが静的に割り当てられる場合は、要請を送信する必要はありません。

手順

- 1 IPv6 ルーター要請を拒否することを確認するために、VMware アプライアンス ホスト マシンで **# grep [01] /proc/sys/net/ipv6/conf/*/router_sollicitations | egrep "default|all"** コマンドを実行します。

IPv6 ルーター通知を拒否するようにホスト マシンが構成されている場合、このコマンドは次のように応答します。

```
/proc/sys/net/ipv6/conf/all/router_sollicitations:0
/proc/sys/net/ipv6/conf/default/router_sollicitations:0
```

ホスト マシンが正しく構成されている場合は、これ以上の操作は必要ありません。

- 2 IPv6 ルーター要請を拒否するようにホスト マシンを構成する必要がある場合は、**/etc/sysctl.conf** ファイルをテキスト エディタで開きます。
- 3 次のエントリを確認します。

```
net.ipv6.conf.all.router_sollicitations=0
net.ipv6.conf.default.router_sollicitations=0
```

エントリがない場合、またはそれらの値がゼロに設定されていない場合は、エントリを追加するか、既存のエントリを更新します。

- 4 変更内容を保存し、ファイルを閉じます。

ルーター要請の IPv6 ルーター プリファレンスの拒否

VMware アプライアンス ホスト マシンが、IPv6 ルーター要請を拒否することを確認します（システム操作で必要でない場合）。

要請設定のルーター プリファレンスにより、ルーター プリファレンスが決まります。アドレスが静的に割り当てられる場合は、要請のルーター プリファレンスを受信する必要はありません。

手順

- 1 IPv6 ルーター要請を拒否することを確認するために、VMware アプライアンス ホスト マシンで **# grep [01] /proc/sys/net/ipv6/conf/*/accept_ra_rtr_pref|egrep "default|all"** コマンドを実行します。

IPv6 ルーター通知を拒否するようにホスト マシンが構成されている場合、このコマンドは次のように応答します。

```
/proc/sys/net/ipv6/conf/all/accept_ra_rtr_pref:0
/proc/sys/net/ipv6/conf/default/accept_ra_rtr_pref:0
```

ホスト マシンが正しく構成されている場合は、これ以上の操作は必要ありません。

- 2 IPv6 ルート要請を拒否するようにホスト マシンを構成する必要がある場合は、**/etc/sysctl.conf** ファイルをテキスト エディタで開きます。
- 3 次のエントリを確認します。

```
net.ipv6.conf.all.accept_ra_rtr_pref=0
net.ipv6.conf.default.accept_ra_rtr_pref=0
```

エントリがない場合、またはそれらの値がゼロに設定されていない場合は、エントリを追加するか、既存のエントリを更新します。

- 4 変更内容を保存し、ファイルを閉じます。

IPv6 ルーター プリフィックスの拒否

VMware アプライアンス ホスト マシンが、IPv6 ルーター プリフィックス情報を拒否することを確認します（システム操作で必要でない場合）。

accept_ra_pinfo 設定により、システムがルーターからプリフィックス情報を受け入れるかどうかが決まります。アドレスが静的に割り当てられる場合は、ルーター プリフィックス情報を受信する必要はありません。

手順

- 1 IPv6 ルーター プリフィックス情報を拒否することを確認するために、VMware アプライアンス ホスト マシンで **# grep [01] /proc/sys/net/ipv6/conf/*/accept_ra_pinfo|egrep "default|all"** コマンドを実行します。

IPv6 ルーター通知を拒否するようにホスト マシンが構成されている場合、このコマンドは次のように応答します。

```
/proc/sys/net/ipv6/conf/all/accept_ra_pinfo:0
/proc/sys/net/ipv6/conf/default/accept_ra_pinfo:0
```

ホスト マシンが正しく構成されている場合は、これ以上の操作は必要ありません。

- 2 IPv6 ルーター プリフィックス情報を拒否するようにホスト マシンを構成する必要がある場合は、**/etc/sysctl.conf** ファイルをテキスト エディタで開きます。

- 3 次のエントリを確認します。

```
net.ipv6.conf.all.accept_ra_pinfo=0
net.ipv6.conf.default.accept_ra_pinfo=0
```

エントリがない場合、またはそれらの値がゼロに設定されていない場合は、エントリを追加するか、既存のエントリを更新します。

- 4 変更内容を保存し、ファイルを閉じます。

IPv6 ルーター通知のホップ制限設定の拒否

VMware アプライアンス ホスト マシンが、IPv6 ルーターのホップ制限設定を拒否することを確認します（必要がない場合）。

accept_ra_defrtr 設定では、システムがルーター通知からのホップ制限設定を受け入れるかどうかを制御します。これをゼロに設定すると、ルーターは、送信パケットに対するデフォルトの IPv6 ホップ制限を変更できません。

手順

- 1 IPv6 ルーターのホップ制限設定を拒否することを確認するために、VMware アプライアンス ホスト マシンで **# grep [01] /proc/sys/net/ipv6/conf/*/accept_ra_defrtr | egrep "default|all"** コマンドを実行します。

IPv6 ルーターのホップ制限設定を拒否するようにホスト マシンが構成されている場合、このコマンドは 0 の値を返します。

```
/proc/sys/net/ipv6/conf/all/accept_ra_defrtr:0
/proc/sys/net/ipv6/conf/default/accept_ra_defrtr:0
```

ホスト マシンが正しく構成されている場合は、これ以上の操作は必要ありません。

- 2 IPv6 ルーターのホップ制限設定を拒否するようにホスト マシンを構成する必要がある場合は、**/etc/sysctl.conf** ファイルをテキスト エディタで開きます。
- 3 次のエントリを確認します。

```
net.ipv6.conf.all.autoconf=0
net.ipv6.conf.default.autoconf=0
```

エントリがない場合、またはそれらの値がゼロに設定されていない場合は、エントリを追加するか、既存のエントリを更新します。

- 4 変更内容を保存し、ファイルを閉じます。

IPv6 ルーター通知 Autoconf 設定の拒否

VMware アプライアンス ホスト マシンが、IPv6 ルーターの autoconf 設定を拒否することを確認します（必要がない場合）。

autoconf 設定は、ルーター通知により、システムがインターフェイスにグローバルユニキャスト アドレスを割り当てられるかどうかを制御します。

手順

- 1 IPv6 ルーター autoconf 設定を拒否することを確認するために、VMware アプライアンス ホスト マシンで **# grep [01] /proc/sys/net/ipv6/conf/*/autoconf|egrep "default|all"** コマンドを実行します。

IPv6 ルーター autoconf 設定を拒否するようにホスト マシンが構成されている場合、このコマンドは 0 の値を返します。

```
/proc/sys/net/ipv6/conf/all/autoconf:0
/proc/sys/net/ipv6/conf/default/autoconf:0
```

ホスト マシンが正しく構成されている場合は、これ以上の操作は必要ありません。

- 2 IPv6 ルーター autoconf 設定を拒否するようにホスト マシンを構成する必要がある場合は、**/etc/sysctl.conf** ファイルをテキスト エディタで開きます。
- 3 次のエントリを確認します。

```
net.ipv6.conf.all.autoconf=0
net.ipv6.conf.default.autoconf=0
```

エントリがない場合、またはそれらの値がゼロに設定されていない場合は、エントリを追加するか、既存のエントリを更新します。

- 4 変更内容を保存し、ファイルを閉じます。

IPv6 近隣要請の拒否

VMware アプライアンス ホスト マシンが、IPv6 近隣要請を拒否することを確認します（必要がない場合）。

dad_transmits 設定は、インターフェイスを構築する際に、目的のアドレスがネットワーク上で確実に一意になるようにするために、アドレス（グローバルおよびリンクローカル）ごとに送信する近隣要請の数を決定します。

手順

- 1 IPv6 近隣要請を拒否することを確認するために、VMware アプライアンス ホスト マシンで **# grep [01] /proc/sys/net/ipv6/conf/*/dad_transmits|egrep "default|all"** コマンドを実行します。

IPv6 近隣要請を拒否するようにホスト マシンが構成されている場合、このコマンドは 0 の値を返します。

```
/proc/sys/net/ipv6/conf/all/dad_transmits:0
/proc/sys/net/ipv6/conf/default/dad_transmits:0
```

ホスト マシンが正しく構成されている場合は、これ以上の操作は必要ありません。

- 2 IPv6 近隣要請を拒否するようにホスト マシンを構成する必要がある場合は、`/etc/sysctl.conf` ファイルをテキスト エディタで開きます。
- 3 次のエントリを確認します。

```
net.ipv6.conf.all.dad_transmits=0
net.ipv6.conf.default.dad_transmits=0
```

エントリがない場合、またはそれらの値がゼロに設定されていない場合は、エントリを追加するか、既存のエントリを更新します。

- 4 変更内容を保存し、ファイルを閉じます。

IPv6 最大アドレス数の制限

VMware アプライアンス ホスト マシンで IPv6 の最大アドレス数設定がシステムの運用に最小限必要な数に制限されていることを確認します。

最大アドレス数の設定では、各インターフェイスが使用できるグローバル ユニキャスト IPv6 アドレスの数を決定します。デフォルトでは 16 ですが、ご使用のシステムに必要な、静的に構成されるグローバル アドレスの正確な数に設定する必要があります。

手順

- 1 VMware アプライアンス ホスト マシンで IPv6 最大アドレス数が適切に制限されていることを確認するために、`# grep [1] /proc/sys/net/ipv6/conf/*/max_addresses | egrep "default|all"` コマンドを実行します。

IPv6 最大アドレス数を制限するようにホスト マシンが構成されている場合、このコマンドは 1 の値を返します。

```
/proc/sys/net/ipv6/conf/all/max_addresses:1
/proc/sys/net/ipv6/conf/default/max_addresses:1
```

ホスト マシンが正しく構成されている場合は、これ以上の操作は必要ありません。

- 2 ホスト マシンで IPv6 最大アドレス数を構成する必要がある場合は、テキスト エディタで `/etc/sysctl.conf` ファイルを開きます。
- 3 次のエントリを確認します。

```
net.ipv6.conf.all.max_addresses=1
net.ipv6.conf.default.max_addresses=1
```

エントリがない場合、またはそれらの値が 1 に設定されていない場合は、エントリを追加するか、既存のエントリを更新します。

- 4 変更内容を保存し、ファイルを閉じます。

Infrastructure as a Service (IaaS) ホストのネットワーク設定の構成

セキュリティのベスト プラクティスとして、VMware の要件とガイドラインに従って、VMware IaaS コンポーネントのホスト マシンのネットワーク通信設定を構成します。

適切なセキュリティで vRealize Automation 機能が完全にサポートされるようにするために、IaaS ホスト マシンのネットワーク構成を設定します。

[\[Infrastructure as a Service \(IaaS\) コンポーネントのセキュリティ保護\]](#) を参照してください。

ポートおよびプロトコルの構成

セキュリティのベスト プラクティスとして、すべての vRealize Automation アプライアンスとコンポーネントで VMware ガイドラインに従ってポートおよびプロトコルを構成します。

重要なシステム コンポーネントが本番環境で動作する必要がある場合、vRealize Automation コンポーネントの入力ポートと出力ポートを構成します。不要なポートおよびプロトコルをすべて無効にします。vRealize Automation リファレンス アーキテクチャを参照してください。

必須のユーザー ポート

セキュリティのベスト プラクティスとして、VMware ガイドラインに従って vRealize Automation のユーザー ポートを構成します。

安全なネットワーク上でのみ必要なポートを公開します。

サーバ	ポート
vRealize Automation アプライアンス	443、8443

管理者に必要なポート

セキュリティのベスト プラクティスとして、VMware のガイドラインに従って vRealize Automation の管理者ポートを構成します。

安全なネットワーク上でのみ必要なポートを公開します。

サーバ	ポート
vRealize Application Services サーバ	5480

vRealize Automation アプライアンス ポート

セキュリティのベスト プラクティスとして、VMware の推奨事項に従って vRealize Automation アプライアンスの入出力ポートを構成します。

入力ポート

vRealize Automation アプライアンス に最低限必要な入力ポートを構成します。システム構成に応じてオプションのポートを構成します。

表 8-1. 必要な最小限の入力ポート

ポート	プロトコル	コメント
443	TCP	vRealize Automation コンソールへのアクセスおよび API 呼び出し。
8443	TCP	コンソール プロキシ (VMRC)。
5480	TCP	仮想アプライアンス Web 管理コンソールへのアクセス。
5488、5489	TCP	内部。更新のための vRealize Automation アプライアンスによる使用。
5672	TCP	RabbitMQ メッセージング。 注意 vRealize Automation アプライアンス インスタンスをクラスタ化する場合、状況によっては 4369 と 25672 をオープン ポートに構成する必要があります。
40002	TCP	vIDM サービスに必要。HA 構成に追加したときの他の vRealize Automation アプライアンス ノードからのトラフィックを除く、すべての外部トラフィックに対するファイアウォールです。

必要に応じて、オプションの入力ポートを構成します。

表 8-2. オプションの入力ポート

ポート	プロトコル	コメント
22	TCP	(オプション) SSH。本番環境で、ポート 22 でリッスンする SSH サービスを無効にして、ポート 22 を閉じます。
80	TCP	(オプション) 443 にリダイレクトします。

出力ポート

必要な出力ポートを構成します。

表 8-3. 必要な最小限の出力ポート

ポート	プロトコル	コメント
25,587	TCP、UDP	出力通知メール送信用の SMTP。
53	TCP、UDP	DNS。
67、68、546、547	TCP、UDP	DHCP。
110、995	TCP、UDP	入力通知メール受信用の POP。
143、993	TCP、UDP	入力通知メール受信用の IMAP。
443	TCP	HTTPS 経由の Infrastructure as a Service (IaaS) Manager Service。

必要に応じて、オプションの出力ポートを構成します。

表 8-4. オプションの出力ポート

ポート	プロトコル	コメント
80	TCP	(オプション) ソフトウェア アップデートの取得用。アップデートをダウンロードして個別に適用できます。
123	TCP、UDP	(オプション) ホスト時刻を使用する代わりに直接 NTP に接続。

Infrastructure as a Service (IaaS) のポート

セキュリティのベスト プラクティスとして、VMware ガイドラインに従って IaaS コンポーネントの入力および出力ポートを構成します。

入力ポート

IaaS コンポーネントに必要な最小限の入力ポートを構成します。

表 8-5. 必要な最小限の入力ポート

コンポーネント	ポート	プロトコル	コメント
Manager Service	443	TCP	HTTPS 経由の IaaS コンポーネントおよび vRealize Automation アプライアンスとの通信。プロキシ エージェントが管理する仮想ホストには、入カトラフィック用の TCP ポート 443 も必要です。

出力ポート

IaaS コンポーネントに必要な最小限の出力ポートを構成します。

表 8-6. 必要な最小限の出力ポート

コンポーネント	ポート	プロトコル	コメント
すべて	53	TCP、UDP	DNS。
すべて		TCP、UDP	DHCP。
Manager Service	443	TCP	HTTPS 経由の vRealize Automation アプライアンスとの通信。
Web サイト	443	TCP	HTTPS 経由の Manager Service との通信。
Distributed Execution Manager	443	TCP	HTTPS 経由の Manager Service との通信。
プロキシ エージェント	443	TCP	HTTPS 経由の Manager Service および仮想化ホストとの通信。
ゲスト エージェント	443	TCP	HTTPS 経由の Manager Service との通信。
Manager Service、Web サイト	1433	TCP	MSSQL。

必要な場合は、オプションの出力ポートを構成します。

表 8-7. オプションの出力ポート

コンポーネント	ポート	プロトコル	コメント
すべて	123	TCP、UDP	NTP はオプションです。

監査とログ

セキュリティのベスト プラクティスとして、VMware の推奨事項に従って監査とログを vRealize Automation システムに設定します。

中央のログ ホストにリモートからログを作成することにより、ログ ファイルを安全に保存できます。中央のホストにログ ファイルを収集すると、単一のツールを使用して環境を監視できます。また、インフラストラクチャ内の複数のエンティティに対して、集計分析や、組織的攻撃などの脅威の証拠の検索を実行できます。セキュリティ保護された中央のログ サーバへのログ記録により、ログの改ざんを防ぐことができ、長期間の監査記録も作成できます。

リモート ログ サーバが安全であることの確認

多くの場合、攻撃者は、ホスト マシンのセキュリティを侵害した後に、ログ ファイルを検索して改ざんすることで痕跡を隠し、発見されることなくシステムをコントロールしようと試みます。リモート ログ サーバを適切にセキュリティ保護することは、ログ改ざんの防止に役立ちます。

認証済み NTP サーバの使用

すべてのホスト マシンが、関連するローカル時間オフセットを含めて同じ相対タイム ソースを使用していることと、その相対タイム ソースを、協定世界時 (UTC) などの承認済みの時間標準に関連付けできることを確認します。統制の取れたアプローチでタイム ソースを扱うことで、関連するログ ファイルを確認するときに、侵入者のアクションを迅速に追跡し、関連付けることができます。時間の設定が正しくないと、ログ ファイルの調査や関連付けを行って攻撃を検出することが難しくなり、監査が不正確になる可能性があります。

少なくとも 3 つの外部タイム ソースの NTP サーバを使用するか、信頼できるネットワーク上にいくつかのローカル NTP サーバを構成し、それらが少なくとも 3 つの外部タイム ソースから順番に時刻を取得するようにします。