安全な構成

2021 年 5 月 19 日
vRealize Operations Manager 8.1

最新の技術ドキュメントは、 VMware の Web サイト （https://docs.vmware.com/jp/）

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

ヴイ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

目次

安全な構成

安全な構成のためのこのドキュメントは、vRealize Operations Manager のデプロイ用の安全なベースラインとして使用することが想定されているものです。システム監視ツールを使用して安全なベースライン構成が継続的に監視および維持されるようにし、予期しない変更がないかどうかを確認するには、このドキュメントを参照してください。

デフォルトでまだ設定されていない強化アクティビティを手動で実行できます。

対象者

この情報は vRealize Operations Manager の管理者を対象としています。

vRealize Operations Manager のセキュリティ状態

1

vRealize Operations Manager 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、およびベスト プラクティスに基づいて、完全に安全な環境であることが想定されます。組織のセキュリティ ポリシーおよびベスト プラクティスに従ってセキュリティ強化アクティビティを実行することが重要です。

このドキュメントは、次のセクションで構成されます。

- 安全なデプロイ
- 安全な構成
- ネットワーク セキュリティ
- 通信

このガイドでは、仮想アプリケーションのインストールについて詳細に説明しています。

システムのセキュリティを適切に強化するため、推奨事項を確認し、組織のセキュリティ ポリシーとリスクへの暴露に関する評価を行います。

vRealize Operations Manager のデプロイのセキュリティ保護

2

ダウンロードされたファイルが真正であることを確認するために、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確認する必要があります。

この章には、次のトピックが含まれています。

インストール メディアの完全性の確認

メディアをダウンロードした後、MD5/SHA1 サム値を使用してダウンロードの完全性を確認します。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後は、ダウンロードしたファイルの完全性と真正さを確実にするために、常に MD5/SHA1 ハッシュを確認してください。VMware から取得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、ソフトウェアを VMware に返品し、交換を依頼してください。

手順

- MD5/SHA1 ハッシュ出力を VMware の Web サイトに掲載されている値と比較します。

  SHA1 ハッシュまたは MD5 ハッシュが一致している必要があります。

  注： vRealize Operations Manager `6.x–x.pak/7.x–x.pak/8.x–x.pak` ファイルは、VMware ソフトウェア発行の証明書によって署名されています。vRealize Operations Manager では、インストールの前に PAK ファイルの署名が検証されます。

デプロイされるソフトウェア インフラストラクチャの強化

強化プロセスの一環として、VMware システムがサポートされるデプロイ済みのソフトウェア インフラストラクチャを強化する必要があります。

VMware システムを強化する前に、サポートするソフトウェア インフラストラクチャにあるセキュリティ欠陥を確認および解決して、完全に強化された安全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要素には、オペレーティング システムのコンポーネント、サポートするソフトウェア、データベース ソフトウェアが含まれます。製造元の推奨事項や他の関連するセキュリティ プロトコルに従って、これらのコンポーネントや他のコンポーネントにあるセキュリティ上の問題を解決します。

VMware vSphere 環境の強化

vRealize Operations Manager のメリットを最大限に活かし、セキュリティ保護されたインフラストラクチャを実現するには、安全な VMware vSphere 環境が必要です。

VMware vSphere 環境を評価し、適切なレベルの vSphere 強化ガイダンスが実施され、維持されていることを確認します。

強化ガイダンスの詳細については、http://www.vmware.com/security/hardening-guides.html を参照してください。

インストールされているサポート対象外のソフトウェアの確認

使用されていないソフトウェアの脆弱性によって、権限のないシステム アクセスや可用性の低下といったリスクが高まる可能性があります。VMware ホスト マシンにインストールされているソフトウェアを確認し、その使用状況を評価します。

システムの安全な運用のために、いずれの vRealize Operations Manager ノード ホストにも、必要でないソフトウェアをインストールしないようにします。使用されていない、または必要でないソフトウェアをアンインストールします。

vRealize Operations Manager などのインフラストラクチャ製品上にサポート対象外のソフトウェア、テストされていないソフトウェア、または未承認のソフトウェアをインストールすることは、インフラストラクチャに対する脅威になります。

インフラストラクチャに対する脅威を最小化するため、VMware 提供のホストでは、VMware によって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インストールしたり使用したりしないでください。

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するために、vRealize Operations Manager のデプロイとインストールされている製品のインベントリを評価します。

サードパーティ製品のサポート ポリシーの詳細については、VMware サポート (http://www.vmware.com/security/hardening-guides.html) 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 で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しないでください。すべてのサードパーティ製ソフトウェアがサードパーティ ベンダーのガイダンスに従って安全に構成され、パッチが適用されていることを確認します。

VMware ホスト マシンにインストールされたサードパーティ製ソフトウェアが真正でない、安全ではない、あるいは脆弱性に対するパッチが適用されていない場合、権限のないアクセスや可用性の低下といったリスクがシステムに生じる可能性があります。VMware 以外によって提供されるすべてのソフトウェアは、適切にセキュリティ保護し、パッチを適用する必要があります。

VMware で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、安全な構成およびパッチ適用の要件について該当のベンダーにお問い合わせください。

VMware によるセキュリティ アドバイザリおよびパッチ

時折、VMware は製品のセキュリティ アドバイザリをリリースします。こうしたアドバイザリを把握することで、安全な基盤製品を確保し、製品が既知の脅威に対して脆弱ではないことを確認できます。

vRealize Operations Manager のインストール、パッチ適用、およびアップグレードの履歴を評価し、リリース済みの VMware セキュリティ アドバイザリに従っていることを確認します。

また、常に最新の vRealize Operations Manager リリースを使用し、最新のセキュリティ修正プログラムを適用することをお勧めします。

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/security/advisories/ を参照してください。

vRealize Operations Manager の安全な構成

3

セキュリティ上のベスト プラクティスとして、vRealize Operations Manager コンソールをセキュリティ保護し、SSH、管理アカウント、およびコンソール アクセスを管理する必要があります。システムが安全な転送チャネルでデプロイされることを確認してください。

また、End Point Operations Management エージェントの実行に関するセキュリティ上のベスト プラクティスに従う必要があります。

この章には、次のトピックが含まれています。

vRealize Operations Manager コンソールのセキュリティ保護

vRealize Operations Manager をインストールした後、初回のログイン時にクラスタ内の各ノードのコンソールをセキュリティ保護する必要があります。

前提条件

vRealize Operations Manager をインストールします。

手順

1. vCenter でノードのコンソールを見つけます。またはノードのコンソールに直接アクセスします。

   vCenter で Alt + F1 キーを押してログイン プロンプトにアクセスします。セキュリティ上の理由から、vRealize Operations Manager のリモート ターミナル セッションは、デフォルトで無効になっています。

2. root としてログインします。

   vRealize Operations Manager では、root パスワードを作成するまで、コマンド プロンプトにアクセスできません。

3. 新しいパスワードのプロンプトで、使用する root パスワードを入力し、今後のために書き留めておきます。
4. root パスワードを再入力します。
5. コンソールからログアウトします。

root パスワードの変更

コンソールを使用して、vRealize Operations Manager のプライマリ ノードまたはデータ ノードの root パスワードをいつでも変更できます。

root ユーザーは、`/etc/pam.d/system-password` にある、`pam_cracklib` モジュールのパスワード複雑性チェックをバイパスします。すべてのセキュリティ強化アプライアンスで、`pw_history` モジュールの `enforce_for_root` を有効にします。これは `/etc/pam.d/system-password` ファイルにあります。デフォルトでは、直前の 5 つのパスワードがシステムに記憶されます。各ユーザーの旧パスワードは、`/etc/security/opasswd` ファイルに格納されます。

前提条件

アプライアンスの root パスワードが、組織で定められているパスワードの複雑性要件を満たしていることを確認します。アカウントのパスワードが `$6$` で始まる場合は、sha512 ハッシュが使用されます。これは、すべてのセキュリティ強化アプライアンスでの標準ハッシュです。

手順

1. アプライアンスの root シェルで `# passwd` コマンドを実行します。
2. 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、`# 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。

3. 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 root パスワードを変更します。

パスワード有効期限の管理

組織のセキュリティ ポリシーに従って、すべてのアカウントのパスワード有効期限を構成します。

すべての VMware セキュリティ強化アプライアンスでは、デフォルトで 60 日のパスワード有効期限が使用されます。大半のセキュリティ強化アプライアンスでは、root アカウントに 365 日のパスワード有効期限が設定されています。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ると、それを復元することはできません。管理パスワードおよび root パスワードの有効期限が切れるのを回避するには、サイト固有のポリシーを導入する必要があります。

手順

1. root として仮想アプライアンス マシンにログインし、`# more /etc/shadow` 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。
2. root アカウントの有効期限を変更するには、`# passwd –x 365 root` コマンドを実行します。

   このコマンドの 365 は、パスワードの有効期限が切れるまでの日数を指定しています。同じコマンドを使用して `root` の代わりに特定のアカウントを使用し、ユーザーを変更します。さらに、組織の有効期限の基準を満たすように日数を置き換えます。

   root パスワードは、デフォルトで 365 日に設定されています。

SSH、管理アカウント、およびコンソール アクセスの管理

リモート接続のために、すべてのセキュリティ強化アプライアンスに Secure Shell (SSH) プロトコルが含まれています。セキュリティ強化アプライアンスでは、デフォルトで SSH が無効化されています。

SSH は、vRealize Operations Manager ノードへの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。SSH には、権限の高いユーザー アカウント認証情報が必要です。SSH アクティビティは、通常、vRealize Operations Manager ノードのロールベース アクセス制御 (RBAC) と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、他の方法で解決できない問題を診断したりトラブルシューティングしたりする場合にのみ、SSH を有効にします。特定の目的のためや、組織のセキュリティ ポリシーに準拠するために必要な場合に限り、SSH を有効のままにします。SSH を有効にする場合は、SSH が確実に攻撃から保護されるようにし、必要な間だけ SSH を有効にするようにしてください。Open Virtualization Format (OVF) テンプレートをデプロイするときは、vSphere の構成に応じて、SSH を有効または無効にできます。

マシン上で SSH が有効になっているかどうかを判断するための簡単なテストとして、SSH を使用して接続を開いてみます。接続が開き、認証情報が要求されたら、SSH は有効になっており、接続に使用できます。

SSH root ユーザー

VMware アプライアンスには事前構成されたデフォルトのユーザー アカウントがないため、デフォルトで、root アカウントで SSH を使用して直接ログインできます。できるだけ早く root の SSH を無効にしてください。

否認防止のコンプライアンス基準を満たすために、すべてのセキュリティ強化アプライアンスの SSH サーバには、2 番目の wheel グループへの SSH アクセスを制限する AllowGroups wheel エントリが事前に構成されています。作業を分担するには、/etc/ssh/sshd_config ファイルで AllowGroups wheel エントリを変更し、sshd などの別のグループを使用します。

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 アクセスが可能になっているため、wheel グループのメンバーは、root パスワードが必要な su-root コマンドを使用できます。グループの分離により、ユーザーはアプライアンスに対して SSH を使用できますが、su コマンドを使用して root としてログインすることはできません。アプライアンスを適切に機能させるために、AllowGroups フィールドの他のエントリの削除や変更は行わないでください。変更後に、# service sshd restart コマンドを実行して SSH デーモンを再起動します。

vRealize Operations Manager ノードで SSH を有効または無効にする

トラブルシューティング用に、vRealize Operations Manager ノードで Secure Shell (SSH) を有効にすることができます。たとえば、サーバをトラブルシューティングするには、SSH を介したサーバへのコンソール アクセスが必要になることがあります。通常の運用には、vRealize Operations Manager ノードで SSH を無効にしてください。

手順

1. vCenter から、vRealize Operations Manager ノードのコンソールにアクセスします。
2. Alt + F1 キーを押してログイン プロンプトにアクセスし、ログインします。
3. #systemctl is-enabled sshd コマンドを実行します。
4. sshd サービスが無効である場合は、#systemctl enable sshd コマンドを実行します。
5. # systemctl start sshd コマンドを実行して sshd サービスを起動します。
6. # systemctl stop sshd コマンドを実行して sshd サービスを停止します。

   vRealize Operations Manager 管理インターフェイスの [SSH ステータス] 列から SSH を有効または無効にすることもできます。

SSH 用のローカル管理アカウントを作成する

root の SSH アクセスを削除する前に、Secure Shell (SSH) を使用でき、セカンダリ wheel グループのメンバーであるローカル管理アカウントを作成する必要があります。

直接 root アクセスを無効にする前に、権限のある管理者が AllowGroups を使用して SSH にアクセスできること、および wheel グループと su コマンドを使用して root としてログインできることをテストしてください。

手順

1. root としてログインし、次のコマンドを実行します。

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

   wheel は、SSH アクセス用に AllowGroups で指定するグループです。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 ユーザーに切り替え、パスワード複雑性チェックを通過できる新しいパスワードを入力します。

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

パスワード複雑性を満たした場合、パスワードが更新されます。パスワード複雑性を満たさなかった場合、パスワードが元のパスワードに戻されるため、パスワード コマンドを再実行する必要があります。

SSH リモート アクセスを許可するためのログイン アカウントを作成し、su コマンドを使用して、wheel アクセスを使用する root としてログインすると、SSH 直接ログインから root アカウントを削除できるようになります。

3 SSH への直接ログインを削除するには、/etc/ssh/sshd_config ファイルを変更します（(#)PermitRootLogin yes を PermitRootLogin no に置き換えます）。

次のステップ

root としての直接ログインを無効にします。強化されたアプライアンスでは、コンソールを通じた root への直接ログインがデフォルトで許可されます。否認防止用の管理アカウントを作成し、wheel アクセス (su–root) 用にテストした後、root として /etc/securetty ファイルを編集し、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。

SSH のアクセスの制限

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 マシン上で SSH パッケージを適切に構成することにより SSH アクセスを制限します。また、こうしたアプライアンスで必要な SSH キー ファイルの権限を維持します。

手順

1 テキスト エディタで仮想アプライアンス ホスト マシン上の /etc/ssh/sshd_config ファイルを開きます。

2 安全な運用のために、本番環境ではローカル ホスト エントリおよび管理ネットワーク サブネットのみを含めるように汎用エントリを変更します。

構成ファイルに次の行を追加します。

```
AllowUsers root@127.0.0.1 root@::1 root@10.0.0.*
```

この例では、すべてのローカル ホスト接続、およびクライアントが 10.0.0.0/24 サブネットから行う接続が許可されます。

3 ファイルを保存して閉じます。

4 systemctl restart sshd を実行して、SSH サービスを再起動します。

SSH キー ファイルの権限の維持

適切なレベルのセキュリティを維持するために、Secure Shell (SSH) キー ファイルの権限を構成します。

手順

1 /etc/ssh/*key.pub にあるパブリック ホスト キー ファイルを確認します。

2 これらのファイルの所有者が root であること、グループの所有者が root であること、およびファイルの権限が 0644 に設定されていることを確認します。

この権限は (-rw-r--r--) です。

3 すべてのファイルを閉じます。

4 /etc/ssh/*key にあるプライベート ホスト キー ファイルを確認します。

5 これらのファイルとグループを root が所有していることと、ファイルの権限が 0600 に設定されていることを確認します。

この権限は (-rw-------) です。

6 すべてのファイルを閉じます。

SSH サーバ構成のセキュリティ保護

可能な場合、Virtual Application のインストール (OVF) には強化されたデフォルト構成が用意されています。ユーザーは、構成ファイルのグローバル オプション セクションにあるサーバおよびクライアント サービスを調べて、構成が適切に強化されていることを確認できます。

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

手順

1 サーバ構成ファイル /etc/ssh/sshd_config を開き、設定が正しいことを確認します。

設定	ステータス
Server Daemon Protocol	Protocol 2
Ciphers	暗号 aes256-ctr,aes128-ctr
TCP Forwarding	AllowTCPForwarding no
Server Gateway Ports	Gateway Ports no
X11 Forwarding	X11Forwarding no
SSH Service	AllowGroups フィールドを使用して、アクセスが許可されるグループを指定し、サービスの使用が許可されるユーザー用のセカンダリ グループにメンバーを追加します。
GSSAPI Authentication	GSSAPIAuthentication no（未使用の場合）
Kerberos Authentication	KerberosAuthentication no（未使用の場合）
Local Variables（AcceptEnv グローバル オプション）	コメントアウトして disabled に設定するか、LC_* または LANG 変数のみに対して enabled に設定
Tunnel Configuration	PermitTunnel no
Network Sessions	MaxSessions 1
Strict Mode Checking	Strict Modes yes
Privilege Separation	UsePrivilegeSeparation yes
rhosts RSA Authentication	RhostsRSAAuthentication no
Compression	Compression delayed または Compression no

設定	ステータス
Message Authentication code	MACs hmac-sha1
User Access Restriction	PermitUserEnvironment no

2 変更内容を保存し、ファイルを閉じます。

SSH クライアント構成を強化する

システム強化監視プロセスの一環として、仮想アプライアンス ホスト マシン上の 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り、SSH クライアントの強化を確認します。

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 セクション内の設定が正しいことを確認します。

設定	ステータス
Client Protocol	Protocol 2
Client Gateway Ports	Gateway Ports no
GSSAPI Authentication	GSSAPIAuthentication no
Local Variables（SendEnv グローバル オプション）	LC_* または LANG 変数のみを指定
CBC Ciphers	暗号 aes256-ctr,aes128-ctr
Message Authentication Codes	MACs `hmac-sha1` エントリのみで使用

2 変更内容を保存し、ファイルを閉じます。

root としての直接ログインを無効にする

強化されたアプライアンスでは、コンソールを使用して root として直接ログインすることがデフォルトで許可されます。セキュリティ上のベスト プラクティスとして、否認防止用の管理アカウントを作成し、`su - root` コマンドを使用して wheel アクセスが可能かどうかをテストした後、直接ログインを無効にすることができます。

前提条件

- SSH 用のローカル管理アカウントを作成するのトピックの手順を実行します。
- root の直接ログインを無効にする前に、管理者としてのシステム アクセスをテストしたことを確認してください。

手順

1 root としてログインし、`/etc/security` ファイルに移動します。

このファイルはコマンド プロンプトからアクセスできます。

2 `tty1` エントリを `console` で置き換えます。

管理者ユーザー アカウントの SSH アクセスを無効にする

セキュリティのベスト プラクティスとして、管理者ユーザー アカウントの SSH アクセスを無効にすることができます。vRealize Operations Manager 管理者アカウントと Linux 管理者アカウントは、同じパスワードを共有しています。管理者ユーザーの SSH アクセスを無効にすることで、すべての SSH ユーザーが最初のログインで vRealize Operations Manager 管理者アカウントと異なるパスワードを使用して低い権限のサービス アカウントにログインした上で、管理者や root などの上位の権限に切り替えるようにして、防御を強化できます。

手順

1 `/etc/ssh/sshd_config` ファイルを編集します。

このファイルはコマンド プロンプトからアクセスできます。

2 ファイル内のいずれかの箇所に `DenyUsers admin` エントリを追加して、ファイルを保存します。

3 sshd サーバを再起動するため、`service sshd restart` コマンドを実行します。

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確保するために、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 ブート ローダーで認証を必要としない場合、システムへのコンソール アクセスを持つユーザーがシステムのブート構成を変更できてしまう可能性があります。また、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、それによってサービス妨害または権限のないシステム アクセスが行われる可能性があります。

デフォルトでは、VMware 仮想アプライアンス上でブート ローダー認証は設定されないので、その構成のために 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/grub.cfg` ファイルにブート パスワードが存在するかどうかを確認します。

2 パスワードが存在しない場合は、仮想アプライアンス上で `/usr/bin/grub2-mkpasswd-pbkdf2` コマンドを実行します。

パスワードが生成され、コマンドによってハッシュ出力が提示されます。

3 `/etc/grub.d/40_custom` の最後に次の行を追加します。

```
set superusers="root"
password_pbkdf2 root <hash of password>
```

4 次のコマンドを使用して、`/boot/grub/grub.cfg` ファイルをバックアップします。

```
cp /boot/grub/grub.cfg /boot/grub/grub.cfg.vropsbackup
```

5 `/usr/sbin/grub2-mkconfig –o /boot/grub/grub.cfg` コマンドを実行して、grub 構成を更新します。

次のステップ

注： 重要：以下で説明するアップグレード手順を実行しないと、アップグレード後に vRealize Operations Manager が開始されません。

パスワードで保護されたブートローダーを使用する場合の vRealize Operations Manager のアップグレード手順は、次のとおりです。

1 次のコマンドを実行して、古い `grub.cfg` をリストアします。

```
cp /boot/grub/grub.cfg.vropsbackup /boot/grub/grub.cfg
```

2 vRealize Operations Manager をアップグレードします。

3 vRealize Operations Manager のアップグレード後に、[ブート ローダー認証の設定] で説明されているすべての手順を実行します。

最低限必要なユーザー アカウントの監視

既存のユーザー アカウントを監視して、不要なユーザー アカウントが削除されるようにする必要があります。

手順

◆ *host*:~ # cat /etc/passwd コマンドを実行して、最低限必要なユーザー アカウントを確認します。

```
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/dev/null:/bin/false
daemon:x:6:6:Daemon User:/dev/null:/bin/false
messagebus:x:18:18:D-Bus Message Daemon User:/var/run/dbus:/bin/false
systemd-bus-proxy:x:72:72:systemd Bus Proxy:/:/bin/false
systemd-journal-gateway:x:73:73:systemd Journal Gateway:/:/bin/false
systemd-journal-remote:x:74:74:systemd Journal Remote:/:/bin/false
systemd-journal-upload:x:75:75:systemd Journal Upload:/:/bin/false
systemd-network:x:76:76:systemd Network Management:/:/bin/false
systemd-resolve:x:77:77:systemd Resolver:/:/bin/false
systemd-timesync:x:78:78:systemd Time Synchronization:/:/bin/false
nobody:x:65534:65533:Unprivileged User:/dev/null:/bin/false
sshd:x:50:50:sshd PrivSep:/var/lib/sshd:/bin/false
apache:x:25:25:Apache Server:/srv/www:/bin/false
ntp:x:87:87:Network Time Protocol:/var/lib/ntp:/bin/false
named:x:999:999::/var/lib/bind:/bin/false
admin:x:1000:1003::/home/admin:/bin/bash
postgres:x:1001:100::/var/vmware/vpostgres/9.6:/bin/bash
```

最低限必要なグループの監視

既存のグループとメンバーを監視して、不要なグループやグループ アクセスが削除されるようにする必要があります。

手順

- <host>:~ # cat /etc/group コマンドを実行して、最低限必要なグループとグループ メンバーシップを確認します。

```
root:x:0:admin
bin:x:1:daemon
sys:x:2:
kmem:x:3:
tape:x:4:
tty:x:5:
daemon:x:6:
floppy:x:7:
disk:x:8:
dialout:x:10:
audio:x:11:
video:x:12:
utmp:x:13:
usb:x:14:
cdrom:x:15:
adm:x:16:
messagebus:x:18:
systemd-journal:x:23:
input:x:24:
mail:x:34:
lock:x:54:
dip:x:30:
systemd-bus-proxy:x:72:
systemd-journal-gateway:x:73:
systemd-journal-remote:x:74:
systemd-journal-upload:x:75:
systemd-network:x:76:
systemd-resolve:x:77:
systemd-timesync:x:78:
nogroup:x:65533:
users:x:100:
sudo:x:27:
wheel:x:28:root,admin
sshd:x:50:
apache:x:25:admin,apache
ntp:x:87:
named:x:999:
vami:x:1000:root
admin:x:1003:
```

vRealize Operations Manager 管理者パスワードのリセット (Linux)

セキュリティ上のベスト プラクティスとして、vApp または Linux インストールによる Linux クラスタ上の vRealize Operations Manager のパスワードをリセットできます。

手順

1 root としてプライマリ ノードのリモート コンソールにログインします。

2 `$VMWARE_PYTHON_BIN $VCOPS_BASE/../vmware-vcopssuite/utilities/sliceConfiguration/bin/vcopsSetAdminPassword.py --reset` コマンドを入力し、プロンプトの指示に従います。

VMware アプライアンスでの NTP の構成

クリティカルな時刻ソーシングを行う場合は、VMware アプライアンスでホスト時刻同期を無効にし、Network Time Protocol (NTP) を使用します。時間同期用の信頼できるリモート NTP サーバを構成する必要があります。NTP サーバは権限のある時間サーバであるか、または少なくとも権限のある時間サーバと同期していることが必要です。

VMware 仮想アプライアンスの NTP デーモンは、同期されたタイム サービスを提供します。NTP はデフォルトで無効になっているため、手動で構成する必要があります。可能な場合は、本番環境で NTP を使用してユーザー アクションを追跡し、正確な監査とログ保存を通じて悪意のある潜在的な攻撃と侵入を検出します。NTP のセキュリティ告知については、NTP の Web サイトを参照してください。

NTP 構成ファイルは、各アプライアンス上の `/etc/ntp.conf` ファイルにあります。

手順

1 仮想アプライアンスのホスト マシン上の `/etc/ntp.conf` 構成ファイルに移動します。

2 ファイル所有権を **`root:root`** に設定します。

3 権限を **`0640`** に設定します。

4 NTP サービスに対するサービス拒否増幅攻撃のリスクを緩和するには、`/etc/ntp.conf` ファイルを開き、そのファイルに restrict 行が存在することを確認します。

```
restrict -4 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。

NTP のセキュリティ告知については、http://support.ntp.org/bin/view/Main/SecurityNotice を参照してください。

Linux 上の TCP タイムスタンプ応答を無効にする

TCP タイムスタンプ応答を使用して、リモート ホストの稼働時間を見積もり、以後の攻撃に利用します。また、一部のオペレーティング システムは、その TCP タイム スタンプの動作に基づいてフィンガープリントされることがあります。

手順

- Linux 上の TCP タイムスタンプ応答を無効にします。
  a `net.ipv4.tcp_timestamps` の値を `0` に設定するには、`sysctl –w net.ipv4.tcp_timestamps=0` コマンドを実行します。
  b デフォルトの `sysctl.conf` ファイルで、`ipv4.tcp_timestamps=0` 値を追加します。

FIPS 140-2 モードを有効にする

vRealize Operations Manager 6.3 以上のリリースで提供される OpenSSL バージョンは、FIPS 140-2 認定です。ただし、FIPS モードはデフォルトでは有効にされていません。

FIPS モードが有効な FIPS 認定の暗号アルゴリズムを使用するセキュリティ コンプライアンス要件がある場合に、FIPS モードを有効にできます。

`Enabling FIPS mode:`

1 SSH またはコンソールを使用して、各クラスタ ノードに `root` としてログインします。

2 テキスト エディタで `/etc/httpd/conf/extra/httpd-ssl.conf` ファイルを開きます。

3 `SSLProxyCipherSuite` の行を見つけて、そのすぐ下に `SSLFIPS on` を追加します。

4 `service httpd restart` コマンドで Apache 構成をリセットします。

`Verifying FIPS mode:`

1 `httpd` サービスの再起動後に、`/var/log/httpd/error.log` ログ ファイルを開きます。

2 `Operating in SSL FIPS mode` というログ イベントを検索します。

`Disabling FIPs mode:`

1 SSH またはコンソールを使用して、各クラスタ ノードに `root` としてログインします。

2 `/etc/httpd/conf/extra/httpd-ssl.conf` ファイルをテキスト エディタで開きます。

3 `SSLFIPS on` を `SSLFIPS off` に置き換えます。

4 `service httpd restart` コマンドで Apache 構成をリセットします。

転送中のデータの TLS

セキュリティ上のベスト プラクティスとして、システムが安全な転送チャネルでデプロイされることを確認してください。

vRealize Operations Manager 用に強力なプロトコルを構成する

SSLv2 や SSLv3 のようなプロトコルは、もはや安全とは見なされていません。また、TLS 1.0 と TLS 1.1 も無効になっていて、TLS 1.2 のみがデフォルトで有効になっています。

注： vRealize Operations Manager 7.5 以降から 8.1 にアップグレードするときに、TLS 設定に対するユーザーの変更は保持されます。vRealize Operations Manager インスタンスを 6.6.1、6.7、7.0 から 8.1 にアップグレードすると、すべての vRealize Operations Manager ノードで TLS 1.0 と TLS 1.1 の両方が無効になります。TLS 1.2 は、デフォルトでサポートされている唯一のプロトコルです。

Apache HTTPD でのプロトコルの正しい使用の確認

vRealize Operations Manager では、SSLv2、SSLv3、TLSv1、TLSv1.1 はデフォルトで無効になっています。システムを本番環境に移行する前に、すべてのロード バランサで脆弱なプロトコルを無効にする必要があります。

手順

1 コマンド プロンプトから `grep SSLProtocol /usr/lib/vmware-vcopssuite/utilities/conf/vcops-apache.conf | grep -v '#'` コマンドを実行し、SSLv2、SSLv3、TLSv1、TLSv1.1 が無効であることを確認します。

プロトコルが無効である場合、このコマンドは出力 `SSLProtocol All -SSLv2 -SSLv3 -TLSv1 -TLSv1.1` を返します。

2 Apache2 サーバを再起動するには、コマンド プロンプトから `systemctl restart httpd` コマンドを実行します。

GemFire TLS ハンドラでのプロトコルの正しい使用の確認

vRealize Operations Manager では、SSLv3、TLS 1.0、TLS 1.1 はデフォルトで無効になっています。システムを本番環境に移行する前に、すべてのロード バランサで脆弱なプロトコルを無効にする必要があります。

手順

1 プロトコルが有効であることを確認します。プロトコルが有効であることを確認するために、各ノードで次のコマンドを実行します。

```
grep cluster-ssl-protocol /usr/lib/vmware-vcops/user/conf/gemfire.properties | grep -v '#'
```

想定される結果は次のとおりです。

```
cluster-ssl-protocols=TLSv1.2
```

```
grep cluster-ssl-protocol /usr/lib/vmware-vcops/user/conf/gemfire.native.properties | grep -v '#'
```

想定される結果は次のとおりです。

```
cluster-ssl-protocols=TLSv1.2
```

```
grep cluster-ssl-protocol /usr/lib/vmware-vcops/user/conf/gemfire.locator.properties | grep -v '#'
```

想定される結果は次のとおりです。

```
cluster-ssl-protocols=TLSv1.2
```

2 TLS 1.0 および TLS 1.1 を再度有効にします。

a 管理者ユーザー インターフェイスに移動し、クラスタをオフラインにします（url/admin）。

b [オンラインにする] をクリックします。

c TLS 1.0 と TLS 1.1 が有効であることを確実にするには、次のコマンドを実行します。

```
sed -i "/^[^#]*cluster-ssl-protocol/ c\cluster-ssl-protocols=TLSv1.2 TLSv1.1
TLSv1" /usr/lib/vmware-vcops/user/conf/gemfire.properties
sed -i "/^[^#]*cluster-ssl-protocol/ c\cluster-ssl-protocols=TLSv1.2 TLSv1.1
TLSv1" /usr/lib/vmware-vcops/user/conf/gemfire.native.properties
sed -i "/^[^#]*cluster-ssl-protocol/ c\cluster-ssl-protocols=TLSv1.2 TLSv1.1
TLSv1" /usr/lib/vmware-vcops/user/conf/gemfire.locator.properties
```

各ノードについてこの手順を繰り返します。

d 管理者ユーザー インターフェイスに移動し、クラスタをオンラインにします。

e [[オンラインにする]] をクリックします。

強力な暗号を使用するように vRealize Operations Manager を構成する

セキュリティを最大限に高めるため、強力な暗号を使用するように vRealize Operations Manager コンポーネントを構成する必要があります。強力な暗号のみが選択されるようにするには、脆弱な暗号の使用を無効にします。強力な暗号をサポートし、十分に大きいキー サイズを使用するようにサーバを構成します。また、暗号は適切な順序で構成します。

vRealize Operations Manager では、DHE キー交換を使用する暗号の使用はデフォルトで無効化されています。本番環境でのシステムの運用を開始する前に、すべてのロード バランサで同じ脆弱な暗号を無効にしてください。

強力な暗号の使用

サーバとブラウザの間でネゴシエートされる暗号化暗号により、TLS セッションで使用されるキー交換方法と暗号化強度が決まります。

Apache HTTPD での暗号化スイートの正しい使用の確認

セキュリティを高めるために、Apache httpd で暗号化スイートが正しく使用されていることを確認します。

手順

1 Apache httpd で暗号化スイートの正しい使用を確認するには、`grep SSLCipherSuite /usr/lib/vmware-vcopssuite/utilities/conf/vcops-apache.conf | grep -v '#'` コマンド プロンプトからコマンドを実行します。

Apache httpd で正しい暗号化スイートが使用されている場合、コマンドは次の出力を返します：
`SSLCipherSuite HIGH:!aNULL!ADH:!EXP:!MD5:! 3DES:!CAMELLIA:!PSK:!SRP:!DH:@STRENGTH`

2 暗号化スイートの正しい使用を構成するには、コマンド プロンプトから sed –i "/^[^#]*SSLCipherSuite/ c\SSLCipherSuite HIGH:\!aNULL\!ADH:\!EXP:\!MD5:\!3DES:\!CAMELLIA:\!PSK:\!SRP:\!DH:@STRENGTH" /usr/lib/vmware–vcopssuite/utilities/conf/vcops–apache.conf コマンドを実行します。

手順 1 で出力が想定どおりではない場合は、このコマンドを実行します。

このコマンドによって、DH/DHE キー交換法を使用するすべての暗号化スイートが無効になります。

3 Apache2 サーバを再起動するために、コマンド プロンプトから /etc/init.d/apache2 restart コマンドを実行します。

4 DH を再有効化するには、コマンド プロンプトから sed –i "/^[^#]*SSLCipherSuite/ c \SSLCipherSuite HIGH:\!aNULL\!ADH:\!EXP:\!MD5:\!3DES:\!CAMELLIA:\!PSK:\!SRP:@STRENGTH" /usr/lib/vmware–vcopssuite/utilities/conf/vcops–apache.conf コマンドを実行することにより、暗号化スイートから !DH を削除します。

5 Apache2 サーバを再起動するために、コマンド プロンプトから systemctl restart httpd コマンドを実行します。

GemFire TLS ハンドラでの暗号化プロトコルの正しい使用の確認

セキュリティを高めるために、GemFire TLS ハンドラで暗号化スイートが正しく使用されていることを確認します。

手順

1 暗号化スイートが有効であることを確認するには、各ノードで次のコマンドを実行してプロトコルが有効であるかどうかを調べます：

```
grep cluster–ssl–ciphers /usr/lib/vmware–vcops/user/conf/gemfire.properties | grep –v '#'
grep cluster–ssl–ciphers /usr/lib/vmware–vcops/user/conf/gemfire.native.properties | grep –v '#'
grep cluster–ssl–ciphers /usr/lib/vmware–vcops/user/conf/gemfire.locator.properties | grep –v '#'
```

2 正しい暗号化スイートを構成します。

a *URL*/admin の管理者ユーザー インターフェイスに移動します。

b クラスタをオフラインにするには、[オフラインにする] をクリックします。

c 正しい暗号化スイートを構成するには、次のコマンドを実行します：

```
sed –i "/^[^#]*cluster–ssl–ciphers/ c\cluster–ssl–
ciphers=TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256
TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384" /usr/lib/vmware–vcops/user/conf/
gemfire.properties
sed –i "/^[^#]*cluster–ssl–ciphers/ c\cluster–ssl–
ciphers=TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256
TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384" /usr/lib/vmware–vcops/user/conf/
gemfire.native.properties
sed –i "/^[^#]*cluster–ssl–ciphers/ c\cluster–ssl–
ciphers=TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256
TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384" /usr/lib/vmware–vcops/user/conf/
gemfire.locator.properties
```

各ノードについてこの手順を繰り返します。

d *URL*/admin の管理者ユーザー インターフェイスに移動します。

e [[オンラインにする]] をクリックします。

localhost 接続での TLS の有効化

デフォルトでは、PostgreSQL データベースへの localhost 接続は TLS を使用しません。TLS を有効にするには、OpenSSL で自己署名の証明書を生成するか、独自の証明書を用意する必要があります。

PostgreSQL への localhost 接続で TLS を有効にするには、次の手順を実行します。

1 OpenSSL で独自の自己署名の証明書を生成するか、独自の自己署名の証明書を用意する

2 PostgreSQL 用証明書のインストール

3 PostgreSQL での TLS の有効化

OpenSSL で独自の自己署名の証明書を生成するか、独自の自己署名の証明書を用意する

PostgreSQL データベースへの localhost 接続は、TLS を使用しません。TLS を有効にするには、OpenSSL で独自の自己署名の証明書を生成するか、独自の証明書を用意します。

- OpenSSL で自己署名の証明書を生成するには、次のコマンドを実行します。

```
openssl req –new –text –out cert.req openssl rsa –in privkey.pem –out cert.pem openssl req –x509 –
in cert.req –text –key cert.pem –out cert.cert
```

- 独自の証明書を用意するには、次の手順を実行します。
  - CAcerts.crt ファイルの所有権を postgres に変更します。
  - postgresql.conf ファイルを編集して、ディレクティブ ssl_ca_file = 'CAcerts.crt を含めます。

CA チェーン付きの証明書を使用している場合は、中間およびルート CA 証明書を含んでいる CAcerts.crt ファイルを同じディレクトリに追加する必要があります。

PostgreSQL 用証明書のインストール

PostgreSQL への localhost 接続で TLS を有効にする場合は、PostgreSQL 用証明書をインストールする必要があります。

手順

1. cert.pem ファイルを /storage/db/vcops/vpostgres/data/server.key にコピーします。
2. cert.cert ファイルを /storage/db/vcops/vpostgres/data/server.crt にコピーします。
3. chmod 600 /storage/db/vcops/vpostgres/data/server.key コマンドを実行します。
4. chmod 600 /storage/db/vcops/vpostgres/data/server.crt コマンドを実行します。
5. chown postgres /storage/db/vcops/vpostgres/data/server.key コマンドと chown postgres /storage/db/vcops/vpostgres/data/server.crt コマンドを実行して、ファイル server.crt と server.key の所有権を root から postgres に変更します。

PostgreSQL での TLS の有効化

PostgreSQL への localhost 接続で TLS を有効にするには、postgresql.conf ファイルを編集する必要があります。

手順

- /storage/db/vcops/vpostgres/data/ にある postgresql.conf ファイルを編集し、次のような変更を加えます。
  a. ssl = on を設定します。
  b. ssl_cert_file = 'server.crt' を設定します。
  c. ssl_key_file = 'server.key' を設定します。

保護する必要のあるアプリケーション リソース

セキュリティ上のベスト プラクティスとして、アプリケーション リソースが保護されるようにします。

次の手順に従って、アプリケーション リソースが保護されるようにします。

手順

1. find / –path /proc –prune –o –type f –perm /6000 –ls コマンドを実行して、正しく定義された SUID および GUID ビット セットがファイルに設定されていることを確認します。

   次のリストが表示されます。

```
584208     44 –rwsr–xr–x   1  root     root        44696 Feb  4  2019 /usr/bin/su
   584210     60 –rwsr–xr–x   1  root     root        54112 Feb  4  2019 /usr/bin/chfn
   584646     56 –rwsr–x–––   1  root     root        51872 Feb  4  2019 /usr/bin/crontab
```

```
   584216     40 -rwsr-xr-x   1  root     root        37128 Feb  4  2019 /usr/bin/newgidmap
   584206     68 -rwsr-xr-x   1  root     root        63736 Feb  4  2019 /usr/bin/passwd
   584211     44 -rwsr-xr-x   1  root     root        44544 Feb  4  2019 /usr/bin/chsh
   584218     40 -rwsr-xr-x   1  root     root        37128 Feb  4  2019 /usr/bin/newuidmap
   587446    144 -rwsr-xr-x   1  root     root       140856 Feb  4  2019 /usr/bin/sudo
   585233     36 -rwsr-xr-x   1  root     root        36144 Feb  4  2019 /usr/bin/umount
   584212     32 -rwsr-xr-x   1  root     root        31048 Feb  4  2019 /usr/bin/expiry
   584209     76 -rwsr-xr-x   1  root     root        71848 Feb  4  2019 /usr/bin/chage
   585231     56 -rwsr-xr-x   1  root     root        52968 Feb  4  2019 /usr/bin/mount
   583901     36 -rwsr-xr-x   1  root     root        34944 Feb  4  2019 /usr/bin/fusermount
   586675     36 -rwsr-xr-x   1  root     root        34952 Feb  4  2019 /usr/bin/fusermount3
   584217     44 -rwsr-xr-x   1  root     root        44472 Feb  4  2019 /usr/bin/newgrp
   584214     80 -rwsr-xr-x   1  root     root        75776 Feb  4  2019 /usr/bin/gpasswd
   582975    428 -rwsr-xr-x   1  root     root       432512 Mar  6  2019 /usr/libexec/ssh-keysign
   587407     80 -rwsr-x---   1  root     root        76224 Feb  4  2019 /usr/libexec/dbus-daemon-
launch-helper
   587109     16 -rwsr-xr-x   1  root     root        14408 Feb  4  2019 /usr/sbin/usernetctl
   587105     16 -rwxr-sr-x   1  root     root        14384 Feb  4  2019 /usr/sbin/netreport
   582750     40 -rwsr-xr-x   1  root     root        38960 Feb  4  2019 /usr/sbin/unix_chkpw
```

2 `find / -path */proc -prune -o -nouser -print -o -nogroup -print` コマンドを実行して、vApp 内のすべてのファイルに所有者が存在することを確認します。

結果が表示されない場合は、すべてのファイルに所有者が存在します。

3 `find / -name "*" -type f -not -path "*/sys*" -not -path "*/proc*" -not -path "*/dev*" -perm -o+w | xargs ls -lb` コマンドを実行して、vApp 上のすべてのファイルの権限を確認し、いずれのファイルも全ユーザーが書き込み可能なファイルではないことを確認します。

Others は書き込み権限を持たないようにしてください。これらのファイルに対する権限は、*##4* または *##5* である必要があります。ここで、*#* は、所有者およびグループに対して指定されるデフォルトの権限セット（*6* または *7* など）と等しいものとします。

4 `find / -path */proc -prune -o ! -user root -o -user admin -print` コマンドを実行して、ファイルが正しいユーザーに所有されていることを確認します。

結果が表示されない場合は、すべてのファイルが `root` または `admin` のどちらかに所有されています。

5 `find /usr/lib/vmware-casa/ -type f -perm -o=w` コマンドを実行して、`/usr/lib/vmware-casa/` ディレクトリ内のファイルが、全ユーザーが書き込み可能なファイルではないことを確認します。

このコマンドで結果が表示されないようにします。

6 `find /usr/lib/vmware-vcops/ -type f -perm -o=w` コマンドを実行して、`/usr/lib/vmware-vcops/` ディレクトリ内のファイルが、全ユーザーが書き込み可能なファイルではないことを確認します。

このコマンドで結果が表示されないようにします。

7 `find /usr/lib/vmware-vcopssuite/ -type f -perm -o=w` コマンドを実行して、`/usr/lib/vmware-vcopssuite/` ディレクトリ内のファイルが、全ユーザーが書き込み可能なファイルではないことを確認します。

このコマンドで結果が表示されないようにします。

Apache の構成

Web ディレクトリの閲覧の無効化

セキュリティ上のベスト プラクティスとして、ユーザーがディレクトリを閲覧できないようにします。ディレクトリを閲覧すると、ディレクトリ トラバーサル攻撃にさらされるリスクが増える可能性があります。

手順

- すべてのディレクトリで、Web ディレクトリの閲覧が無効になっていることを確認します。
  a テキスト エディタで、/etc/apache2/default-server.conf ファイルと /usr/lib/vmware-vcopssuite/utilities/conf/vcops-apache.conf ファイルを開きます。
  b それぞれの <Directory> 項目で、Options 行に該当タグの Indexes オプションが含まれていないことを確認します。

Apache2 サーバのサンプル コードの削除

Apache には、2 つのサンプルの共通ゲートウェイ インターフェイス（CGI）スクリプト（printenv および test-cgi）が含まれます。本番 Web サーバには、運用上必要なコンポーネントのみを含める必要があります。これらのコンポーネントは、攻撃者に対してシステムの重要情報を開示する可能性があります。

セキュリティ上のベスト プラクティスとして、cgi-bin ディレクトリから CGI スクリプトを削除してください。

手順

- test-cgi スクリプトおよび prinenv スクリプトを削除するには、rm /usr/share/doc/packages/apache2/test-cgi コマンドおよび rm /usr/share/doc/packages/apache2/printenv コマンドを実行します。

Apache2 サーバのサーバ トークンの確認

システム セキュリティ強化プロセスの一部として、Apache2 サーバのサーバ トークンを確認します。HTTP 応答の Web サーバ応答ヘッダーには、複数の情報フィールドを含めることができます。情報には、要求された HTML ページ、Web サーバのタイプとバージョン、オペレーティング システムとバージョン、および Web サーバに関連するポートが含まれます。この情報によって、悪意のあるユーザーが広範なツールを使用することなく重要な情報を入手できます。

ディレクティブ ServerTokens は、Prod に設定する必要があります。たとえば、ServerTokens Prod など。このディレクティブは、クライアントに返されるサーバの応答ヘッダー フィールドに、オペレーティング システムの説明およびコンパイルイン モジュールに関する情報を含めるかどうかを制御します。

手順

1. サーバ トークンを確認するには、cat /etc/apache2/sysconfig.d/global.conf | grep ServerTokens コマンドを実行します。
2. ServerTokens OS を ServerTokens Prod に変更するには、sed –i 's/\(ServerTokens\s\+\)OS/\1Prod/g' /etc/apache2/sysconfig.d/global.conf コマンドを実行します。

Apache2 サーバのトレース方法の無効化

標準的な本番処理では、データの侵害に結びつく診断機能を使用することによって未検出な脆弱性を発見できることがあります。データの悪用を避けるために、HTTP Trace メソッドを無効にします。

手順

1. Apache2 サーバの Trace 方法を確認するには、コマンド `grep TraceEnable /usr/lib/vmware-vcopssuite/utilities/conf/vcops-apache.conf` を実行します。
2. Apache2 サーバの Trace 方法を無効にするには、コマンド `sed -i "/^[^#]*TraceEnable/ c \TraceEnable off" /usr/lib/vmware-vcopssuite/utilities/conf/vcops-apache.conf` を実行します。

構成モードを無効にする

ベスト プラクティスとして、vRealize Operations Manager をインストール、構成、または保守するときは、インストールのトラブルシューティングおよびデバッグが有効になるように構成または設定を変更できます。

適切にセキュリティ保護されるように、加える変更をそれぞれカタログ化および監査します。構成変更が正しくセキュリティ保護されているかどうかがよくわからない場合は、本番環境に移行しないでください。

非必須ソフトウェア コンポーネントの管理

セキュリティ上のリスクを最小限にするために、vRealize Operations Manager ホスト マシンの非必須ソフトウェアを削除するか構成します。

セキュリティ侵害を引き起こす可能性を最小限にするため、削除しないすべてのソフトウェアについては、メーカーの推奨事項とセキュリティ上のベスト プラクティスに従って構成します。

USB 大容量ストレージ ハンドラのセキュリティ保護

USB 大容量ストレージ ハンドラをセキュリティ保護し、vRealize アプライアンスでこのハンドラがデフォルトでロードされないようにします。また、vRealize アプライアンスでこのハンドラが USB デバイス ハンドラとして使用されないようにします。潜在的な攻撃者がこのハンドラを悪用して悪意のあるソフトウェアをインストールする可能性があります。

手順

1. テキスト エディタで `/etc/modprobe.d/modprobe.conf` ファイルを開きます。
2. このファイルに `install usb-storage /bin/false` 行が含まれることを確認します。
3. ファイルを保存して閉じます。

Bluetooth プロトコル ハンドラのセキュリティ保護

潜在的な攻撃者による悪用を防止するために、vRealize アプライアンスで Bluetooth プロトコル ハンドラをセキュリティ保護します。

ネットワーク スタックへの Bluetooth プロトコルのバインドは不要であり、バインドするとホストの攻撃対象領域が増加します。vRealize アプライアンスで、Bluetooth プロトコル ハンドラ モジュールがデフォルトでロードされないようにします。

手順

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

2 このファイルに行 `install bluetooth /bin/false` が含まれることを確認します。

3 ファイルを保存して閉じます。

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

vRealize アプライアンスで、Stream Control Transmission Protocol (SCTP) モジュールがデフォルトでロードされないようにします。潜在的な攻撃者がシステムのセキュリティを侵害するためにこのプロトコルを活用する可能性があります。

絶対に必要でなければ、SCTP モジュールをロードしないようにシステムを構成してください。SCTP は、使用されていない 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、ホストの攻撃対象領域が増加します。権限のないローカル プロセスがこのプロトコルを使用してソケットを開くことにより、カーネルでプロトコル ハンドラが動的にロードされる場合があります。

手順

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

2 このファイルに次の行が含まれることを確認します。

```
install sctp /bin/false
```

3 ファイルを保存して閉じます。

Datagram Congestion Control Protocol のセキュリティ保護

システム セキュリティ強化アクティビティの一環として、vRealize アプライアンスで Datagram Congestion Control Protocol (DCCP) モジュールがデフォルトでロードされないようにします。潜在的な攻撃者がシステムのセキュリティを侵害するためにこのプロトコルを活用する可能性があります。

絶対に必要でなければ、DCCP モジュールはロードしないようにしてください。DCCP は提案中のトランスポート レイヤー プロトコルであり、使用されません。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、ホストの攻撃対象領域が増加します。権限のないローカル プロセスがこのプロトコルを使用してソケットを開くことにより、カーネルでプロトコル ハンドラが動的にロードされる場合があります。

手順

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

2 このファイルに DCCP 行が含まれることを確認します。

```
install dccp /bin/false
install dccp_ipv4 /bin/false
install dccp_ipv6 /bin/false
```

3 ファイルを保存して閉じます。

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

システム セキュリティ強化アクティビティの一環として、vRealize アプライアンスで Reliable Datagram Sockets (RDS) プロトコルがデフォルトでロードされないようにします。潜在的な攻撃者がシステムのセキュリティを侵害するためにこのプロトコルを活用する可能性があります。

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、ホストの攻撃対象領域が増加します。権限のないローカル プロセスがこのプロトコルを使用してソケットを開くことにより、カーネルでプロトコル ハンドラが動的にロードされる場合があります。

手順

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

2 このファイルに `install rds /bin/false` 行が含まれることを確認します。

3 ファイルを保存して閉じます。

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

システム セキュリティ強化アクティビティの一環として、仮想アプライアンス ホスト マシンで Transparent Inter-Process Communication (TIPC) プロトコルがデフォルトでロードされないようにします。潜在的な攻撃者がシステムのセキュリティを侵害するためにこのプロトコルを活用する可能性があります。

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、ホストの攻撃対象領域が増加します。権限のないローカル プロセスがこのプロトコルを使用してソケットを開くことにより、カーネルでプロトコル ハンドラが動的にロードされる場合があります。

手順

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

2 このファイルに `install tipc /bin/false` 行が含まれることを確認します。

3 ファイルを保存して閉じます。

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

vRealize アプライアンスで、Internetwork Packet Exchange (IPX) プロトコルがデフォルトでロードされないようにします。潜在的な攻撃者がシステムのセキュリティを侵害するためにこのプロトコルを活用する可能性があります。

絶対に必要でなければ、IPX プロトコル モジュールはロードしないようにしてください。IPX プロトコルは、現在ではほとんど使用されることのないネットワーク 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、ホストの攻撃対象領域が増加します。権限のないローカル プロセスがこのプロトコルを使用してソケットを開くことにより、システムでプロトコル ハンドラが動的にロードされる場合があります。

手順

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

2 このファイルに行 `install ipx /bin/false` が含まれることを確認します。

3 ファイルを保存して閉じます。

AppleTalk プロトコルのセキュリティ保護

vRealize アプライアンスで、AppleTalk プロトコルがデフォルトでロードされないようにします。潜在的な攻撃者がシステムのセキュリティを侵害するためにこのプロトコルを活用する可能性があります。

必要でなければ、AppleTalk プロトコル モジュールはロードしないようにしてください。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、ホストの攻撃対象領域が増加します。権限のないローカル プロセスがこのプロトコルを使用してソケットを開くことにより、システムでプロトコル ハンドラが動的にロードされる場合があります。

手順

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

2 このファイルに行 `install appletalk /bin/false` が含まれることを確認します。

3 ファイルを保存して閉じます。

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。潜在的な攻撃者がシステムのセキュリティを侵害するためにこのプロトコルを活用する可能性があります。

絶対に必要でなければ、DECnet プロトコル モジュールはロードしないようにしてください。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、ホストの攻撃対象領域が増加します。権限のないローカル プロセスがこのプロトコルを使用してソケットを開くことにより、システムでプロトコル ハンドラが動的にロードされる場合があります。

手順

1 テキスト エディタで DECnet プロトコルの `/etc/modprobe.d/modprobe.conf` ファイルを開きます。

2 このファイルに行 `install decnet /bin/false` が含まれることを確認します。

3 ファイルを保存して閉じます。

Firewire モジュールのセキュリティ保護

vRealize アプライアンスで、Firewire モジュールがデフォルトでロードされないようにします。潜在的な攻撃者がシステムのセキュリティを侵害するためにこのプロトコルを活用する可能性があります。

必要でなければ、Firewire モジュールはロードしないようにしてください。

手順

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

2 このファイルに行 `install ieee1394 /bin/false` が含まれることを確認します。

3 ファイルを保存して閉じます。

カーネル メッセージのログ

/Etc/sysctl.conf ファイル内の kernel.printk は、カーネルのログ出力を指定します。

次の 4 つの値を指定します。

- console loglevel. コンソールに出力されるメッセージの最低優先順位。
- default loglevel. 固有のログ レベルのないメッセージの最低レベル。
- コンソール ログ レベルの出力可能な最低レベル。
- コンソール ログ レベルのデフォルト値。

値ごとに 8 つのエントリから指定可能です。

- define KERN_EMERG "<0>" /* system is unusable */
- define KERN_ALERT "<1>" /* action must be taken immediately */
- define KERN_CRIT "<2>" /* critical conditions */
- define KERN_ERR "<3>" /* error conditions */
- define KERN_WARNING "<4>" /* warning conditions */
- define KERN_NOTICE "<5>" /* normal but significant condition */
- define KERN_INFO "<6>" /* informational */
- define KERN_DEBUG "<7>" /* debug-level messages */

kernel.printk の値を **3 4 1 7** に設定して、/etc/sysctl.conf ファイル内に行 kernel.printk=3 4 1 7 が存在することを確認します。

End Point Operations Management エージェント

End Point Operations Management エージェントは、エージェントベースの検出および監視機能を vRealize Operations Manager に追加します。

End Point Operations Management エージェントは、ホスト上に直接インストールされるため、信頼性が End Point Operations Management サーバと同じレベルとは限りません。したがって、エージェントが安全にインストールされていることを確認する必要があります。

End Point Operations Management エージェントを実行するためのセキュリティ上のベスト プラクティス

ユーザー アカウントの使用時には、セキュリティ上のベスト プラクティスに従う必要があります。

- サイレント インストールの場合は、AGENT_HOME/conf/agent.properties ファイルに保存されたすべての認証情報とサーバ証明書のサムプリントを削除します。
- End Point Operations Management エージェント登録用に特別に予約された vRealize Operations Manager ユーザー アカウントを使用します。詳細については、vRealize Operations Manager ヘルプのトピック「vRealize Operations Manager でのロールと権限」を参照してください。

- インストールの完了後に、エージェント登録に使用した vRealize Operations Manager ユーザー アカウントを無効にします。エージェント管理アクティビティ用のユーザー アクセスを有効にする必要があります。詳細については、vRealize Operations Manager ヘルプのトピック「vRealize Operations Manager でのユーザーとグループの構成」を参照してください。
- エージェントを実行するシステムがセキュリティ侵害を受けた場合、vRealize Operations Manager ユーザー インターフェイスを使用してエージェント リソースを削除することによって、エージェント証明書を破棄できます。詳細については、セクション「エージェントの破棄」を参照してください。

エージェント機能に最低限必要な権限

サービスをインストールおよび変更するための権限が必要です。実行中のプロセスを検出する場合、エージェントの実行に使用するユーザー アカウントには、プロセスとプログラムにアクセスする権限も必要になります。Windows オペレーティング システムのインストールの場合、サービスをインストールおよび変更する権限が必要です。Linux のインストールでは、RPM インストーラを使用してエージェントをインストールする場合は、エージェントをサービスとしてインストールする権限が必要です。

エージェントを vRealize Operations Manager サーバに登録するために必要な最低限の権限は、システム内のオブジェクトに対する割り当てが一切ない、Agent Manager のロールが付与されているユーザー用の権限です。

Linux ベースのプラットフォームにおけるファイルと権限

End Point Operations Management エージェントのインストール後は、エージェントをインストールしたユーザーがエージェントの所有者になります。

End Point Operations Management エージェントをインストールするユーザーが TAR ファイルを解凍するか、RPM をインストールするときに、所有者に対するインストール ディレクトリとファイルの権限（600、700 など）が設定されます。

注： ZIP ファイルを展開する場合は、権限が正しく適用されない可能性があります。権限が正しいことを確認してください。

エージェントによって作成および書き込みが行われるすべてのファイルには 700 の権限が割り当てられ、所有者はエージェントを実行するユーザーになります。

表 3-1. Linux のファイルと権限

ディレクトリまたはファイル	権限	グループまたはユーザー	読み取り	書き込み	実行
エージェント ディレクトリ/bin	700	所有者	可	可	可
		グループ	不可	不可	不可
		すべて	不可	不可	不可
エージェント ディレクトリ/conf	700	所有者	可	可	可
		グループ	不可	不可	不可
		すべて	不可	不可	不可
エージェント ディレクトリ/log	700	所有者	可	可	不可
		グループ	不可	不可	不可

表 3-1. Linux のファイルと権限 （続き）

ディレクトリまたはファイル	権限	グループまたはユーザー	読み取り	書き込み	実行
		すべて	不可	不可	不可
エージェント ディレクトリ/data	700	所有者	可	可	可
		グループ	不可	不可	不可
		すべて	不可	不可	不可
エージェント ディレクトリ/bin/ep-agent.bat	600	所有者	可	可	不可
		グループ	不可	不可	不可
		すべて	不可	不可	不可
エージェント ディレクトリ/bin/ep-agent.sh	700	所有者	可	可	可
		グループ	不可	不可	不可
		すべて	不可	不可	不可
エージェント ディレクトリ/conf/* （conf ディレクトリ内のすべてのファイル）	600	所有者	可	可	可
		グループ	不可	不可	不可
		すべて	不可	不可	不可
エージェント ディレクトリ/log/* （log ディレクトリ内のすべてのファイル）	600	所有者	可	可	不可
		グループ	不可	不可	不可
		すべて	不可	不可	不可
エージェント ディレクトリ/data/* （data ディレクトリ内のすべてのファイル）	600	所有者	可	可	不可
		グループ	不可	不可	不可
		すべて	不可	不可	不可

Windows ベースのプラットフォームにおけるファイルと権限

End Point Operations Management エージェントの Windows ベースのインストールでは、エージェントをインストールするユーザーには、サービスをインストールおよび変更する権限が必要です。

End Point Operations Management エージェントをインストールした後、インストール フォルダ（すべてのサブディレクトリおよびファイルを含む）には、SYSTEM、管理者グループ、およびインストール ユーザーのみがアクセス可能になるようにします。`ep-agent.bat` を使用して End Point Operations Management エージェントをインストールする場合、セキュリティ強化プロセスが成功したことを確認します。エージェントをインストールするユーザーには、エラー メッセージのメモを取ることをお勧めします。セキュリティ強化プロセスが失敗した場合、ユーザーはこれらの権限を手動で適用できます。

表 3-2. Windows のファイルと権限

ディレクトリまたはファイル	グループまたはユーザー	フル コントロール	変更	読み取りと実行	読み取り	書き込み
<agent directory>/bin	SYSTEM	可	-	-	-	-
	システム管理者	可	-	-	-	-

表 3-2. Windows のファイルと権限 （続き）

ディレクトリまたはファイル	グループまたはユーザー	フル コントロール	変更	読み取りと実行	読み取り	書き込み
	インストール ユーザー	可	-	-	-	-
	ユーザー		-	-	-	-
<agent directory>/conf	SYSTEM	可	-	-	-	-
	システム管理者	可	-	-	-	-
	インストール ユーザー	可	-	-	-	-
	ユーザー		-	-	-	-
<agent directory>/log	SYSTEM	可	-	-	-	-
	システム管理者	可	-	-	-	-
	インストール ユーザー	可	-	-	-	-
	ユーザー		-	-	-	-
<agent directory>/data	SYSTEM	可	-	-	-	-
	システム管理者	可	-	-	-	-
	インストール ユーザー	可	-	-	-	-
	ユーザー		-	-	-	-
<agent directory>/bin/hq-agent.bat	SYSTEM	可	-	-	-	-
	システム管理者	可	-	-	-	-
	インストール ユーザー	可	-	-	-	-
	ユーザー		-	-	-	-
<agent directory>/bin/hq-agent.sh	SYSTEM	可	-	-	-	-
	システム管理者	可	-	-	-	-
	インストール ユーザー	可	-	-	-	-
	ユーザー		-	-	-	-
<agent directory>/conf/* （conf ディレクトリ内のすべてのファイル）	SYSTEM	可	-	-	-	-
	システム管理者	可	-	-	-	-
	インストール ユーザー	可	-	-	-	-
	ユーザー		-	-	-	-
<agent directory>/log/*	SYSTEM	可	-	-	-	-
	システム管理者	可	-	-	-	-

表 3-2. Windows のファイルと権限 （続き）

ディレクトリまたはファイル	グループまたはユーザー	フル コントロール	変更	読み取りと実行	読み取り	書き込み
(log ディレクトリ内のすべてのファイル)	インストール ユーザー	可	-	-	-	-
	ユーザー		-	-	-	-
<agent directory>/data/* (data ディレクトリ内のすべてのファイル)	SYSTEM	可	-	-	-	-
	システム管理者	可	-	-	-	-
	インストール ユーザー	可	-	-	-	-
	ユーザー		-	-	-	-

エージェント ホストで開かれるポート

エージェント プロセスは、構成可能な 2 つのポート 127.0.0.1:2144 および 127.0.0.1:32000 でコマンドをリッスンします。これらのポートは任意に割り当て可能であるため、正確なポート番号は異なることがあります。エージェントは、外部インターフェイス上のポートを開きません。

表 3-3. 最低限必要なポート

ポート	プロトコル	方向	コメント
443	TCP	発信	HTTP、TCP、または ICMP での発信接続のためにエージェントによって使用されます。
2144	TCP	リッスン	内部専用。構成可能。エージェントと、それをロードおよび構成するコマンド ラインの間のプロセス間通信に使用されます。エージェント プロセスはこのポート上でリッスンします。 **注：** ポート番号は任意に割り当てることができ、異なる場合があります。
32000	TCP	リッスン	内部専用。構成可能。エージェントと、それをロードおよび構成するコマンド ラインの間のプロセス間通信に使用されます。エージェント プロセスはこのポート上でリッスンします。 **注：** ポート番号は任意に割り当てることができ、異なる場合があります。

エージェントの破棄

エージェントが実行するシステムがセキュリティ侵害を受けた場合など、何らかの理由でエージェントを破棄する必要がある場合は、システムからエージェント リソースを削除できます。以降の要求では、確認に失敗します。

vRealize Operations Manager のユーザー インターフェイスを使用してエージェント リソースを削除することにより、エージェント証明書を破棄できます。詳細については、エージェント リソースの削除 を参照してください。

システムが再び安全になったら、エージェントを復元できます。詳細については、エージェント リソースの復元 を参照してください。

エージェント リソースの削除

vRealize Operations Manager を使用してエージェント リソースを削除することにより、エージェント証明書を破棄できます。

前提条件

以前に記録されたメトリック データを使用してリソースの継続性を維持するには、リソースの詳細に表示される End Point Operations Management エージェント トークンを記録しておきます。

手順

1. vRealize Operations Manager ユーザー インターフェイスの [インベントリ] ページに移動します。
2. アダプタ タイプ ツリーを開きます。
3. EP Ops アダプタ リストを開きます。
4. [EP Ops Agent - *HOST_DNS_NAME*] を選択します。
5. [Edit Object] をクリックします。
6. エージェント トークン文字列であるエージェント ID を記録します。
7. [Edit Object] ダイアログ ボックスを閉じます。
8. [EP Ops Agent - *HOST_DNS_NAME*] を選択し、[Delete Object] をクリックします。

エージェント リソースの復元

システムの安全な状態がリカバリされた場合、破棄されたエージェントを復元できます。これによってエージェントは、履歴データを損失することなく、引き続き同じリソースについてレポートできます。このためには、エージェント リソースを削除する前に記録されたのと同じトークンを使用して新しい End Point Operations Management トークン ファイルを作成する必要があります。「エージェント リソースの削除」のセクションを参照してください。

前提条件

- End Point Operations Management トークン文字列が記録されていることを確認します。
- vRealize Operations Manager サーバからエージェント リソースを削除する前に記録されたリソース トークンを使用します。
- エージェントの管理権限を保有していることを確認します。

手順

1. エージェントを実行するユーザーでエージェント トークン ファイルを作成します。

   例として、123-456-789 トークンを含むトークン ファイルを作成するコマンドを実行します。

   - Linux：

     ```
     echo 123-456-789 > /etc/epops/epops-token
     ```

   - Windows：

     ```
     echo 123-456-789 > %PROGRAMDATA%\VMware\Ep Ops Agent\epops-token
     ```

   この例では、トークン ファイルは該当するプラットフォームでのデフォルトのトークンの場所に書き込まれます。

2 新しいエージェントをインストールし、vRealize Operations Manager サーバに登録します。トークン ファイルに書き込まれたトークンがエージェントによってロードされたことを確認します。

このアクションを実行するには、エージェントの管理権限が必要です。

エージェントの証明書の取消と更新

再発行フローは、setup コマンド ライン引数を使用して、エージェントから開始されます。すでに登録されているエージェントが setup コマンド ラインの引数 ep-agent.sh setup を使用し、必要な認証情報を入力すると、新しい registerAgent コマンドがサーバに送信されます。

サーバは、エージェントがすでに登録されていることを検出すると、別のエージェント リソースを作成せずに、そのエージェントに新しいクライアント証明書を送信します。エージェント側では、古い証明書が新しい証明書に置き換えられます。サーバ証明書の変更で ep-agent.sh setup コマンドを実行する場合、新しい証明書を信頼するように求めるメッセージが表示されます。この処理をサイレントに実行するには、agent.properties ファイルに新しいサーバ証明書のサムプリントを指定してから ep-agent.sh setup コマンドを実行します。

前提条件

エージェントの権限を管理して、証明書の取消と更新を行います。

手順

- Linux ベースのオペレーティング システムでは、エージェントのホストで ep-agent.sh setup コマンドを実行します。Windows ベースのオペレーティング システムでは、ep-agent.bat setup コマンドを実行します。

  サーバの証明書が変更されていることをエージェントが検出すると、メッセージが表示されます。新しい証明書が有効で信頼できるものであれば、新しい証明書を受け入れます。

End Point Operations Management エージェントのパッチ適用と更新

必要に応じて、vRealize Operations Manager リリースとは別に、新しい End Point Operations Management エージェント バンドルを利用できます。

End Point Operations Management エージェントには、パッチやアップデートは提供されません。最新のセキュリティ修正を含むエージェントの最新のバージョンをインストールする必要があります。クリティカルなセキュリティ修正は、VMware のセキュリティに関する注意のガイダンスに従って伝えられます。セキュリティに関する注意のトピックを参照してください。

その他の安全な構成アクティビティ

サーバのユーザー アカウントを確認し、不要なアプリケーションをホスト サーバから削除します。不要なポートをブロックし、不要なホスト サーバ上で実行されているサービスを無効にします。

サーバ ユーザー アカウント設定の確認

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と設定について、不必要なユーザー アカウントが存在していないか確認することをお勧めします。

アプリケーションの実行に関連しないユーザー アカウントは、管理、メンテナンス、トラブルシューティングに必要なアカウントのみに制限します。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小限のアクセスのみに制限します。これらのアカウントを厳格に管理および監査してください。

不要なアプリケーションを削除し、無効にする

不要なアプリケーションをホスト サーバから削除します。不要なアプリケーションが増えるほど、不明の脆弱性やパッチが適用されていない脆弱性によるセキュリティ侵害のリスクが高まります。

不要なポートおよびサービスを無効にする

トラフィックを許可するオープン ポートのリストについては、ホスト サーバのファイアウォールを確認してください。

このドキュメントのポートおよびプロトコルの構成セクションで vRealize Operations Manager の最小要件としてリストされていないすべてのポート（不要なポート）をブロックします。さらに、ホスト サーバ上で実行されているサービスを監査し、不要なサービスを無効にします。

ネットワーク セキュリティと安全な通信

4

セキュリティ上のベスト プラクティスとして、VMware 仮想アプライアンスおよびホスト マシンのネットワーク通信設定を確認および編集します。また、vRealize Operations Manager の受信ポートと発信ポートの構成は最小限にする必要があります。

この章には、次のトピックが含まれています。

仮想アプリケーション インストール用のネットワーク設定の構成

VMware 仮想アプライアンスとホスト マシンが安全で不可欠な通信のみを許可するようにするには、ネットワーク通信設定を確認および編集します。

TCP バックログのキュー サイズの設定

セキュリティ上のベスト プラクティスとして、VMware アプライアンス ホスト マシン上でデフォルトの TCP バックログのキュー サイズを構成します。TCP サービス拒否攻撃を緩和するために、TCP バックログ キューの適切なデフォルト サイズを設定します。推奨されるデフォルト設定は 1280 です。

手順

1. それぞれの VMware アプライアンス ホスト マシン上で `# cat /proc/sys/net/ipv4/tcp_max_syn_backlog` コマンドを実行します。
2. TCP バックログのキュー サイズの設定
   a. テキスト エディタで `/etc/sysctl.conf` ファイルを開きます。
   b. ファイルに次のエントリを追加することにより、デフォルトの TCP バックログ キュー サイズを設定します。

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

   c. 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 エコーを拒否する

インターネット制御メッセージ プロトコル (ICMP) エコーのブロードキャストに応答すると、増幅攻撃に攻撃経路を知らせ、悪意のあるエージェントがネットワーク マッピングを行いやすくなる可能性があります。ICMPv4 エコーを無視するようにシステムを構成すると、このような攻撃から保護できます。

手順

1. `# cat /proc/sys/net/ipv4/icmp_echo_ignore_broadcasts` コマンドを実行して、システムが ICMP ブロードキャスト アドレス エコー要求を送信していないことを確認します。
2. ICMPv4 ブロードキャスト アドレス エコー要求を拒否するようにホスト システムを構成します。
   a. テキスト エディタで `/etc/sysctl.conf` ファイルを開きます。
   b. このエントリの値が 1 に設定されていない場合は、`net.ipv4.icmp_echo_ignore_broadcasts=1` エントリを追加します。
   c. 変更内容を保存し、ファイルを閉じます。

IPv4 プロキシ ARP を無効にするようにホスト システムを構成する

IPv4 プロキシ ARP を使用すると、システムは、あるインターフェイスに接続されているホストの代理として別のインターフェイス上で ARP 要求への応答を送信できます。権限のない情報共有を防止するには、IPv4 プロキシ ARP を無効にする必要があります。接続されているネットワーク セグメント間のアドレス指定情報の漏えいを防止するには、この設定を無効にします。

手順

1. `# grep [01] /proc/sys/net/ipv4/conf/*/proxy_arp|egrep "default|all"` コマンドを実行して、プロキシ ARP が無効にされているかどうかを確認します。
2. IPv4 プロキシ ARP を無効にするようにホスト システムを構成します。
   a. テキスト エディタで `/etc/sysctl.conf` ファイルを開きます。
   b. 値が `0` に設定されていない場合は、エントリを追加するか、それに合わせて既存のエントリを更新します。値を `0` に設定します。

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

   c. 行った変更内容を保存し、ファイルを閉じます。

IPv4 ICMP リダイレクト メッセージを無視するようにホスト システムを構成する

セキュリティ上のベスト プラクティスとして、ホスト システムで IPv4 インターネット制御メッセージ プロトコル (ICMP) リダイレクト メッセージが無視されることを確認します。悪意のある ICMP リダイレクト メッセージが使用されると、中間者攻撃が発生する可能性があります。ルータは、ICMP リダイレクト メッセージを使用して、特定の転送先へのより直接的なルートが存在することをホストに通知します。これらのメッセージは、ホストのルート テーブルを変更しますが、認証を受けません。

手順

1 ホスト システム上で `# grep [01] /proc/sys/net/ipv4/conf/*/accept_redirects|egrep "default|all"` コマンドを実行して、ホスト システムで IPv4 リダイレクト メッセージが無視されるかどうかをチェックします。

2 IPv4 ICMP リダイレクト メッセージを無視するようにホスト システムを構成します。

   a `/etc/sysctl.conf` ファイルを開きます。

   b 値が `0` に設定されていない場合は、次のエントリをファイルに追加するか、それに合わせて既存のエントリを更新します。値を `0` に設定します。

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

   c 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 メッセージを無視するようにホスト システムを構成する

セキュリティ上のベスト プラクティスとして、ホスト システムで IPv6 インターネット制御メッセージ プロトコル (ICMP) リダイレクト メッセージが無視されることを確認します。悪意のある ICMP リダイレクト メッセージが使用されると、中間者攻撃が発生する可能性があります。ルータは、ICMP リダイレクト メッセージを使用して、特定の転送先へのより直接的なルートが存在することをホストに知らせます。これらのメッセージは、ホストのルート テーブルを変更しますが、認証を受けません。

手順

1 ホスト システム上で `# grep [01] /proc/sys/net/ipv6/conf/*/accept_redirects|egrep "default|all"` コマンドを実行し、IPv6 リダイレクト メッセージが無視されるかどうかをチェックします。

2 IPv6 ICMP リダイレクト メッセージを無視するようにホスト システムを構成します。

   a `/etc/sysctl.conf` を開いて、IPv6 リダイレクト メッセージを無視するようにホスト システムを構成します。

   b 値が `0` に設定されていない場合は、次のエントリをファイルに追加するか、それに合わせて既存のエントリを更新します。値を `0` に設定します。

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

   c 変更内容を保存し、ファイルを閉じます。

IPv4 ICMP リダイレクトを拒否するようにホスト システムを構成する

セキュリティ上のベスト プラクティスとして、ホスト システムで IPv4 インターネット制御メッセージ プロトコル (ICMP) リダイレクトが拒否されることを確認します。ルータは、ICMP リダイレクト メッセージを使用して、特定の転送先への直接のルートが存在することをサーバに通知します。このメッセージには、システムのルート テーブルからの情報が含まれており、ネットワーク トポロジの各部分が明らかになる可能性があります。

手順

1 ホスト システム上で `# grep [01] /proc/sys/net/ipv4/conf/*/send_redirects|egrep "default|all"` を実行して、IPv4 ICMP リダイレクトが拒否されることを確認します。

2 IPv4 ICMP リダイレクトを拒否するようにホスト システムを構成します。

   a `/etc/sysctl.conf` ファイルを開いて、ホスト システムを構成します。

   b 値が `0` に設定されていない場合は、次のエントリをファイルに追加するか、それに合わせて既存のエントリを更新します。値を `0` に設定します。

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

   c 変更内容を保存し、ファイルを閉じます。

IPv4 の出所不明パケットをログに記録するためのホスト システムの構成

セキュリティ上のベスト プラクティスとして、ホスト システムが IPv4 の出所不明パケットをログに記録していることを確認します。出所不明パケットには、無効であると認識されているアドレスが含まれます。メッセージをログに記録するようにホスト システムを構成すると、不適切な構成や進行中の攻撃を特定できます。

手順

1 `# grep [01] /proc/sys/net/ipv4/conf/*/log_martians|egrep "default|all"` コマンドを実行して、ホストが IPv4 の出所不明パケットをログに記録しているかどうかを確認します。

2 IPv4 の出所不明パケットをログに記録するようにホスト システムを構成します。

   a `/etc/sysctl.conf` ファイルを開いて、ホスト システムを構成します。

   b 値が 1 に設定されていない場合は、次のエントリをファイルに追加するか、それに合わせて既存のエントリを更新します。値を 1 に設定します。

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

   c 変更内容を保存し、ファイルを閉じます。

IPv4 リバース パス フィルタリングを使用するようにホスト システムを構成する

セキュリティ上のベスト プラクティスとして、IPv4 リバース パス フィルタリングを使用するようにホスト マシンを構成します。リバース パス フィルタリングは、ルートがないソース アドレスを持つパケット、またはルートが送信元のインターフェイスの方を指していないパケットをシステムに破棄させることで、偽装されたソース アドレスから保護します。

可能であれば必ず、リバース パス フィルタリングを使用するようにシステムを構成します。システム ロールによっては、リバース パス フィルタリングによって正当なトラフィックが破棄されることがあります。このような場合、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必要が生じることがあります。

手順

1 ホスト システム上で `# grep [01] /proc/sys/net/ipv4/conf/*/rp_filter|egrep "default|all"` コマンドを実行して、システムで IPv4 リバース パス フィルタリングが使用されているかどうかをチェックします。

2 IPv4 リバース パス フィルタリングを使用するようにホスト システムを構成します。

a `/etc/sysctl.conf` ファイルを開いて、ホスト システムを構成します。

b 値が 1 に設定されていない場合は、次のエントリをファイルに追加するか、それに合わせて既存のエントリを更新します。値を 1 に設定します。

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

c 変更内容を保存し、ファイルを閉じます。

IPv4 転送を拒否するようにホスト システムを構成する

セキュリティ上のベスト プラクティスとして、ホスト システムで IPv4 転送が拒否されることを確認します。指定されたルータでないシステムが IP 転送を行うように構成されている場合、このシステムを使用して、ネットワーク デバイスでフィルタ処理されない通信のパスを提供することにより、ネットワーク セキュリティがバイパスされることがあります。

手順

1 `# cat /proc/sys/net/ipv4/ip_forward` コマンドを実行して、ホストで IPv4 転送が拒否されるかどうかを確認します。

2 IPv4 転送を拒否するようにホスト システムを構成します。

a `/etc/sysctl.conf` を開いて、ホスト システムを構成します。

b 値が `0` に設定されていない場合は、次のエントリをファイルに追加するか、同じように既存のエントリを更新します。値を `0` に設定します。

```
net.ipv4.ip_forward=0
```

c 変更内容を保存し、ファイルを閉じます。

IPv4 ソース ルーティングされたパケットの転送を拒否するようにホスト システムを構成する

ソース ルーティングされたパケットを使用すると、パケットのソースが、ルータで構成されている内容とは異なるパスに沿ってルータがパケットを転送するように指示できるようになります。これを使用して、ネットワークのセキュリティ対策がバイパスされることがあります。

IPv4 転送が有効になっていて、システムがルータとして機能している場合など、この要件は、ソース ルーティングされたトラフィックの転送にのみ適用されます。

手順

1 `# grep [01] /proc/sys/net/ipv4/conf/*/accept_source_route|egrep "default|all"` コマンドを実行して、システムで IPv4 ソース ルーティングされたパケットが使用されないかどうかを確認します。

2 IPv4 ソース ルーティングされたパケットの転送を拒否するようにホスト システムを構成します。

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

   b 値が `0` に設定されていない場合は、`net.ipv4.conf.all.accept_source_route=0` と `et.ipv4.conf.default.accept_source_route=0` が `0` に設定されていることを確認します。

   c ファイルを保存して閉じます。

IPv6 転送を拒否するようにホスト システムを構成する

セキュリティ上のベスト プラクティスとして、ホスト システムで IPv6 転送が拒否されることを確認します。指定されたルータでないシステムが IP 転送を行うように構成されている場合、このシステムを使用して、ネットワーク デバイスでフィルタ処理されない通信のパスを提供することにより、ネットワーク セキュリティがバイパスされることがあります。

手順

1 `# grep [01] /proc/sys/net/ipv6/conf/*/forwarding|egrep "default|all"` コマンドを実行して、ホストで IPv6 転送が拒否されるかどうかを確認します。

2 IPv6 転送を拒否するようにホスト システムを構成します。

   a `/etc/sysctl.conf` を開いて、ホスト システムを構成します。

   b 値が `0` に設定されていない場合は、次のエントリをファイルに追加するか、それに合わせて既存のエントリを更新します。値を `0` に設定します。

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

   c 変更内容を保存し、ファイルを閉じます。

IPv4 TCP SYN Cookie を使用するためのホスト システムの構成

セキュリティ上のベスト プラクティスとして、ホスト システムで IPv4 Transmission Control Protocol (TCP) SYN Cookie を使用していることを確認します。TCP SYN フラッディング攻撃は、システムの TCP 接続テーブルを SYN_RCVD 状態の接続で満たすことにより、サービス妨害を引き起こす可能性があります。SYN Cookie は、後続の ACK を受信してイニシエータが有効な接続を試みておりフラッディング ソースではないことが確認されるまで、接続の追跡を行わないようにする手法です。

この手法は、標準準拠の方法では完全には動作しませんが、フラッディング条件の検出時にのみ有効化され、有効な要求の処理を継続しながらシステムを保護できます。

手順

1 `# cat /proc/sys/net/ipv4/tcp_syncookies` コマンドを実行して、ホスト システムで IPv4 TCP SYN Cookie を使用しているかどうかを確認します。

2 IPv4 TCP SYN Cookie を使用するためのホスト システムの構成

a /etc/sysctl.conf を開いて、ホスト システムを構成します。

b 値が 1 に設定されていない場合は、次のエントリをファイルに追加するか、同じように既存のエントリを更新します。値を 1 に設定します。

```
net.ipv4.tcp_syncookies=1
```

c 変更内容を保存し、ファイルを閉じます。

IPv6 ルータ通知を拒否するようにホスト システムを構成する

セキュリティ上のベスト プラクティスとして、ホスト システムで、必要でない限りルータ通知とインターネット制御メッセージ プロトコル (ICMP) リダイレクトの受け入れが拒否されることを確認します。IPv6 を使用すると、システムがネットワークからの情報を自動的に使用してネットワーク デバイスを構成できます。セキュリティ上の観点から、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むしろ、手動で設定することをお勧めします。

手順

1 ホスト システム上で # grep [01] /proc/sys/net/ipv6/conf/*/accept_ra|egrep "default|all" を実行して、システムで、必要でない限りルータ通知とインターネット制御メッセージ プロトコル (ICMP) リダイレクトの受け入れが拒否されることを確認します。

2 IPv6 ルータ通知を拒否するようにホスト システムを構成します。

a /etc/sysctl.conf ファイルを開きます。

b 値が 0 に設定されていない場合は、次のエントリをファイルに追加するか、それに合わせて既存のエントリを更新します。値を 0 に設定します。

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

c 変更内容を保存し、ファイルを閉じます。

IPv6 ルータ要請を拒否するようにホスト システムを構成する

セキュリティ上のベスト プラクティスとして、ホスト システムで、必要でない限り IPv6 ルータ要請が拒否されることを確認します。ルータ要請設定では、インターフェイスを構築するとき、近隣要請を何件送信するかを指定します。アドレスが静的に割り当てられる場合、要請を送信する必要はありません。

手順

1 # grep [01] /proc/sys/net/ipv6/conf/*/router_solicitations|egrep "default|all" コマンドを実行して、必要でない限りホスト システムで IPv6 ルータ要請が拒否されるかどうかを確認します。

2 IPv6 ルータ要請を拒否するようにホスト システムを構成します。

a /etc/sysctl.conf を開きます。

b 値が 0 に設定されていない場合は、次のエントリをファイルに追加するか、それに合わせて既存のエントリを更新します。値を 0 に設定します。

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

c 変更内容を保存し、ファイルを閉じます。

ルータ要請で IPv6 ルータ プリファレンスを拒否するようにホスト システムを構成する

セキュリティ上のベスト プラクティスとして、ホスト システムで、必要でない限り IPv6 ルータ要請が拒否されることを確認します。要請設定のルータ プリファレンスにより、ルータ 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、要請のルータ プリファレンスを受信する必要はありません。

手順

1 ホスト システム上で # grep [01] /proc/sys/net/ipv6/conf/*/accept_ra_rtr_pref|egrep "default|all" を実行して、ホスト システムで IPv6 ルータ要請が拒否されるかどうかを確認します。

2 ルータ要請で IPv6 ルータ プリファレンスを拒否するようにホスト システムを構成します。

a /etc/sysctl.conf ファイルを開きます。

b 値が 0 に設定されていない場合は、次のエントリをファイルに追加するか、それに合わせて既存のエントリを更新します。値を 0 に設定します。

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

c 変更内容を保存し、ファイルを閉じます。

IPv6 ルータ プリフィックスを拒否するようにホスト システムを構成する

セキュリティ上のベスト プラクティスとして、ホスト システムで、必要でない限り IPv6 ルータ プリフィックス情報が拒否されることを確認します。accept ra pinfo 設定は、システムでルータからのプリフィックス情報を受け入れるかどうかを制御します。アドレスが静的に割り当てられる場合、システムでルータ 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 # grep [01] /proc/sys/net/ipv6/conf/*/accept_ra_pinfo|egrep "default|all" を実行して、そのシステムで IPv6 ルータ プリフィックス情報が拒否されるかどうかを確認します。

2 IPv6 ルータ プリフィックスを拒否するようにホスト システムを構成します。

a /etc/sysctl.conf ファイルを開きます。

b 値が 0 に設定されていない場合は、次のエントリをファイルに追加するか、それに合わせて既存のエントリを更新します。値を 0 に設定します。

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

c 変更内容を保存し、ファイルを閉じます。

IPv6 ルータ通知のホップ制限設定を拒否するためのホスト システムの構成

セキュリティ上のベスト プラクティスとして、必要な場合を除き、ホスト システムがルータ通知からの IPv6 ルータ通知のホップ制限設定を拒否していることを確認します。accept_ra_defrtr 設定は、システムがルータ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 0 に設定すると、ルータ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なくなります。

手順

1 # grep [01] /proc/sys/net/ipv6/conf/*/accept_ra_defrtr|egrep "default|all" コマンドを実行して、ホスト システムが IPv6 ルータのホップ制限設定を拒否することを確認します。

2 値が 0 に設定されていない場合は、IPv6 ルータ通知のホップ制限設定を拒否するようにホスト システムを構成します。

a /etc/sysctl.conf ファイルを開きます。

b 値が 0 に設定されていない場合は、次のエントリをファイルに追加するか、それに合わせて既存のエントリを更新します。値を 0 に設定します。

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

c 変更内容を保存し、ファイルを閉じます。

IPv6 ルータ通知 autoconf 設定を拒否するようにホスト システムを構成する

セキュリティ上のベスト プラクティスとして、ホスト システムで IPv6 ルータ通知 autoconf 設定が拒否されることを確認します。autoconf 設定は、ルータ通知によってシステムから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ることができるかどうかを制御します。

手順

1 # grep [01] /proc/sys/net/ipv6/conf/*/autoconf|egrep "default|all" コマンドを実行して、ホスト システムで IPv6 ルータ通知 autoconf 設定が拒否されるかどうかを確認します。

2 値が 0 に設定されていない場合は、IPv6 ルータ通知 autoconf 設定を拒否するようにホスト システムを構成します。

a /etc/sysctl.conf ファイルを開きます。

b 値が 0 に設定されていない場合は、次のエントリをファイルに追加するか、それに合わせて既存のエントリを更新します。値を 0 に設定します。

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

c 変更内容を保存し、ファイルを閉じます。

IPv6 近隣要請を拒否するようにホスト システムを構成する

セキュリティ上のベスト プラクティスとして、ホスト システムで、必要でない限り IPv6 近隣要請が拒否されることを確認します。目的のアドレスがネットワーク上で一意となるようにインターフェイスを構築するとき、グローバルおよびリンク ローカルを含むアドレスごとに近隣要請を何件送出するかを dad_transmits 設定で指定します。

手順

1 # grep [01] /proc/sys/net/ipv6/conf/*/dad_transmits|egrep "default|all" コマンドを実行して、ホスト システムで IPv6 近隣要請が拒否されるかどうかを確認します。

2 値が 0 に設定されていない場合は、IPv6 近隣要請を拒否するようにホスト システムを構成します。

a /etc/sysctl.conf ファイルを開きます。

b 値が 0 に設定されていない場合は、次のエントリをファイルに追加するか、それに合わせて既存のエントリを更新します。値を 0 に設定します。

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

c 変更内容を保存し、ファイルを閉じます。

IPv6 の最大アドレス数を制限するためのホスト システムの構成

セキュリティ上のベスト プラクティスとして、ホストで割り当てることができる IPv6 アドレスの最大数が制限されていることを確認します。この最大アドレス数の設定によって、各インターフェイスに割り当てることができるグローバル ユニキャスト IPv6 アドレスの数が決定されます。デフォルトは 16 ですが、静的に構成されるグローバル アドレスとして必要な数を設定する必要があります。

手順

1 # grep [1] /proc/sys/net/ipv6/conf/*/max_addresses|egrep "default|all" コマンドを実行して、ホスト システムで割り当てることができる IPv6 アドレスの最大数が制限されているかどうかを確認します。

2 これらの値が 1 に設定されていない場合は、割り当てることができる IPv6 アドレスの最大数を制限するようにホスト システムを構成します。

a /etc/sysctl.conf ファイルを開きます。

b 次のエントリをファイルに追加するか、またはこれらの既存のエントリを更新します。値を 1 に設定します。

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

c 変更内容を保存し、ファイルを閉じます。

ポートおよびプロトコルの構成

セキュリティ上のベスト プラクティスとして、不可欠ではないポートおよびプロトコルをすべて無効にします。

重要なシステム コンポーネントが本番環境で動作するために最小限必要な受信および送信ポートを vRealize Operations Manager コンポーネントで構成します。

最低限のデフォルト受信ポート

セキュリティ上のベスト プラクティスとして、vRealize Operations Manager が本番で稼働するのに必要な受信ポートを構成します。

表 4-1. 最低限必要な受信ポート

ポート	プロトコル	コメント
443	TCP	vRealize Operations Manager ユーザー インターフェイスおよび vRealize Operations Manager 管理者インターフェイスへのアクセスに使用されます。
123	UDP	プライマリ ノードへの Network Time Protocol (NTP) 同期のために vRealize Operations Manager によって使用されます。
5433	TCP	高可用性が有効になっているときに、グローバル データベース (vPostgreSQL) を複製するために、プライマリ ノードおよびレプリカ ノードによって使用されます。
7001	TCP	Cassandra によって、ノード間クラスタ通信をセキュリティ保護するために使用されます。 このポートはインターネットに公開しないでください。このポートをファイアウォールに追加します。
9042	TCP	Cassandra によって、ノード間でのクライアント関連通信をセキュリティ保護するために使用されます。 このポートはインターネットに公開しないでください。このポートをファイアウォールに追加します。
6061	TCP	クライアントが Gemfire ロケータに接続して分散型システムでサーバに対する接続情報を取得するときに使用します。また、クライアント（からの要求）を最も負荷の少ないサーバに送信するためにサーバの負荷を監視する目的にも使用されます。

表 4-1. 最低限必要な受信ポート （続き）

ポート	プロトコル	コメント
10000-10010	TCP と UDP	ユニキャスト UDP メッセージングとピアツーピア分散システムにおける TCP 障害検出に使用される GemFire Server 短期ポート範囲。
20000-20010	TCP と UDP	ユニキャスト UDP メッセージングとピアツーピア分散システムにおける TCP 障害検出に使用される GemFire ロケータ短期ポート範囲。

表 4-2. オプションの受信ポート

ポート	プロトコル	コメント
22	TCP	任意。Secure Shell (SSH)。運用環境では、ポート 22 や他のポート上での SSH サービスのリッスンは無効にし、ポート 22 は閉じる必要があります。
80	TCP	任意。443 にリダイレクトします。
3091-3101	TCP	Horizon View がインストールされている場合に、Horizon View から vRealize Operations Manager のデータにアクセスするために使用されます。

暗号

ここでは、受信、ノード間、および送信の接続に基づいて分類される暗号化スイートのリストを示します。暗号化スイートのリストはカンマ区切りのリストです。

vRealize Operations Manager への受信接続

表 4-3. 受信接続の暗号

名前	暗号
構成されている暗号化スイート	
Apache の暗号	ECDHE-RSA-AES256-GCM-SHA384、ECDHE-RSA-AES128-GCM-SHA256、ECDHE-RSA-AES256-SHA384、ECDHE-RSA-AES128-SHA256、ECDHE-RSA-AES256-SHA、ECDHE-RSA-AES128-SHA、AES256-GCM-SHA384、AES128-GCM-SHA256、AES256-SHA256、AES128-SHA256、AES256-SHA、AES128-SHA
構成可能：OS の暗号スイート リストへの Apache リレーを検索するには、CLI コマンド `openssl ciphers –v` を実行します。	

vRealize Operations Manager ノードのノード間接続

表 4-4. vRealize Operations Manager ノードのノード間接続

名前	暗号
構成されている暗号化スイート	
inter_cluster プロトコル：TLSv1.2、TLSv1.1、TLSv1	TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA、 TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA、 TLS_ECDH_ECDSA_WITH_AES_256_CBC_SHA、 TLS_ECDH_RSA_WITH_AES_256_CBC_SHA、 TLS_DHE_RSA_WITH_AES_256_CBC_SHA、 TLS_DHE_DSS_WITH_AES_256_CBC_SHA、 TLS_DH_RSA_WITH_AES_256_CBC_SHA、 TLS_DH_DSS_WITH_AES_256_CBC_SHA、 TLS_RSA_WITH_AES_256_CBC_SHA、 TLS_RSA_WITH_AES_128_CBC_SHA、 TLS_DHE_DSS_WITH_AES_128_CBC_SHA、 TLS_DH_RSA_WITH_AES_128_CBC_SHA、 TLS_DH_DSS_WITH_AES_128_CBC_SHA

vRealize Operations Manager からの送信接続

構成済みの送信接続の暗号化スイートは、次の 3 つの種類に分類されます。

- アダプタから送信元
- 認証ソース
- 送信プラグイン

表 4-5. アダプタから送信元

名前	暗号
adapter_to_source プロトコル：TLSv1.2、TLSv1.1、TLSv1	TLS_ECDH_RSA_WITH_AES_256_GCM_SHA384, TLS_ECDH_ECDSA_WITH_AES_256_GCM_SHA384, TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384, TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384, TLS_RSA_WITH_AES_256_GCM_SHA384, TLS_ECDH_RSA_WITH_AES_128_GCM_SHA256, TLS_ECDH_ECDSA_WITH_AES_128_GCM_SHA256, TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256, TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256, TLS_RSA_WITH_AES_128_GCM_SHA256, TLS_ECDH_RSA_WITH_AES_256_CBC_SHA384, TLS_ECDH_ECDSA_WITH_AES_256_CBC_SHA384, TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384, TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384, TLS_RSA_WITH_AES_256_CBC_SHA256, TLS_ECDH_RSA_WITH_AES_256_CBC_SHA, TLS_ECDH_ECDSA_WITH_AES_256_CBC_SHA, TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA, TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA, TLS_RSA_WITH_AES_256_CBC_SHA, TLS_ECDH_RSA_WITH_AES_128_CBC_SHA256, TLS_ECDH_ECDSA_WITH_AES_128_CBC_SHA256, TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256, TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256, TLS_RSA_WITH_AES_128_CBC_SHA256, TLS_ECDH_RSA_WITH_AES_128_CBC_SHA, TLS_ECDH_ECDSA_WITH_AES_128_CBC_SHA, TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA, TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA, TLS_RSA_WITH_AES_128_CBC_SHA, TLS_DHE_RSA_WITH_AES_256_CBC_SHA, TLS_DHE_DSS_WITH_AES_256_CBC_SHA, TLS_DH_RSA_WITH_AES_256_CBC_SHA, TLS_DH_DSS_WITH_AES_256_CBC_SHA, TLS_DHE_DSS_WITH_AES_128_CBC_SHA, TLS_DH_RSA_WITH_AES_128_CBC_SHA, TLS_DH_DSS_WITH_AES_128_CBC_SHA

表 4-6. 認証ソース

名前	暗号
LDAP プロトコル：TLSv1.2、TLSv1.1、TLSv1	TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA、 TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA、 TLS_ECDH_ECDSA_WITH_AES_256_CBC_SHA、 TLS_ECDH_RSA_WITH_AES_256_CBC_SHA、 TLS_DHE_RSA_WITH_AES_256_CBC_SHA、 TLS_DHE_DSS_WITH_AES_256_CBC_SHA、 TLS_DH_RSA_WITH_AES_256_CBC_SHA、 TLS_DH_DSS_WITH_AES_256_CBC_SHA、 TLS_RSA_WITH_AES_256_CBC_SHA、 TLS_RSA_WITH_AES_128_CBC_SHA、 TLS_DHE_DSS_WITH_AES_128_CBC_SHA、 TLS_DH_RSA_WITH_AES_128_CBC_SHA、 TLS_DH_DSS_WITH_AES_128_CBC_SHA
csp プロトコル：TLSv1.2、TLSv1.1、TLSv1	TLS_ECDH_RSA_WITH_AES_256_GCM_SHA384, TLS_ECDH_ECDSA_WITH_AES_256_GCM_SHA384, TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384, TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384, TLS_RSA_WITH_AES_256_GCM_SHA384, TLS_ECDH_RSA_WITH_AES_128_GCM_SHA256, TLS_ECDH_ECDSA_WITH_AES_128_GCM_SHA256, TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256, TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256, TLS_RSA_WITH_AES_128_GCM_SHA256, TLS_ECDH_RSA_WITH_AES_256_CBC_SHA384, TLS_ECDH_ECDSA_WITH_AES_256_CBC_SHA384, TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384, TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384, TLS_RSA_WITH_AES_256_CBC_SHA256, TLS_ECDH_RSA_WITH_AES_256_CBC_SHA, TLS_ECDH_ECDSA_WITH_AES_256_CBC_SHA, TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA, TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA, TLS_RSA_WITH_AES_256_CBC_SHA, TLS_ECDH_RSA_WITH_AES_128_CBC_SHA256, TLS_ECDH_ECDSA_WITH_AES_128_CBC_SHA256, TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256, TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256, TLS_RSA_WITH_AES_128_CBC_SHA256, TLS_ECDH_RSA_WITH_AES_128_CBC_SHA, TLS_ECDH_ECDSA_WITH_AES_128_CBC_SHA, TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA, TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA, TLS_RSA_WITH_AES_128_CBC_SHA, TLS_DHE_RSA_WITH_AES_256_CBC_SHA, TLS_DHE_DSS_WITH_AES_256_CBC_SHA, TLS_DH_RSA_WITH_AES_256_CBC_SHA, TLS_DH_DSS_WITH_AES_256_CBC_SHA, TLS_DHE_DSS_WITH_AES_128_CBC_SHA, TLS_DH_RSA_WITH_AES_128_CBC_SHA, TLS_DH_DSS_WITH_AES_128_CBC_SHA

表 4-6. 認証ソース （続き）

名前	暗号
vIDM	暗号化スイートは特に設定されていません。
sso_util	暗号化スイートは特に設定されていません。

表 4-7. 送信プラグイン

email_sender	暗号化スイートは特に設定されていません。
rest_sender	暗号化スイートは特に設定されていません。

vRealize Operations Manager システムでの監査とロギング

5

セキュリティ上のベスト プラクティスとして、vRealize Operations Manager システムでの監査とロギングを設定します。

監査とロギングの詳細な実装は、このドキュメントの対象外です。

中央のログ ホストにリモート ロギングを行うことで、ログを安全に保存できます。ログ ファイルを中央のホストに集めると、環境を単一のツールで簡単に監視できます。集計分析を実行し、インフラストラクチャ内の複数のエンティティに対する組織的攻撃を検索することもできます。安全な統合ログ サーバにロギングすると、ログの改ざんを防止でき、長期的な監査レコードを確保することもできます。

この章には、次のトピックが含まれています。

- リモート ロギング サーバのセキュリティ保護
- 公認の NTP サーバの使用
- クライアント ブラウザに関する考慮事項

リモート ロギング サーバのセキュリティ保護

セキュリティ上のベスト プラクティスとして、権限のあるユーザーのみがリモート ロギング サーバを構成でき、リモート ロギング サーバが安全であることを確認します。

ホスト マシンのセキュリティに違反する攻撃者は、自らの足跡を隠し、発見されることなくコントロールを維持するために、ログ ファイルを検索して改ざんを試みることがあります。

公認の NTP サーバの使用

すべてのホスト システムが同じ相対タイム ソースを使用しており、適切な現地オフセットを使用していることを確認します。協定世界時 (UTC) などの合意された時間標準に相対タイム ソースを関連付けることができます。

該当するログ ファイルを確認することで、侵入者のアクションを簡単に追跡し、関連付けることができます。設定時刻が正しくないと、攻撃検出のためにログ ファイルを検査して関連付けることが困難になり、また監査が不正確になる可能性があります。タイム ソースの外部にある 3 つ以上の NTP サーバを使用する方法、または 3 つ以上の外部タイム ソースから時刻を取得する複数のローカル NTP サーバを信頼済みネットワーク上に構成する方法があります。

クライアント ブラウザに関する考慮事項

セキュリティ上のベスト プラクティスとして、信頼できないクライアントやパッチが適用されていないクライアント、あるいはブラウザの拡張機能を使用するクライアントから vRealize Operations Manager を使用しないでください。