VMware vRealize Automation 6.2

Hardening Guide

TECHNICAL WHITE PAPER

Table of Contents

The vRealize Automation 6.2 Security Posture

The security posture of vRealize Automation assumes a holistically secure environment, which involves system configuration, network configuration, organizational security policies, and best practices. The recommendations in this document are broken down into the following sections:

- Secure Deployment
- Secure Configuration
- Network Security
- Communication

The following major components are discussed:

- vRealize Appliance
- Identity Appliance
- Application Services Appliance
- IaaS Component

To ensure that your system is securely hardened, review the recommendations in this guide and assess them against your organization's security policies and risk exposure.

To familiarize yourself with the system and how the pieces operate together, see Foundations and Concepts in the VMware vRealize Automation 6.2 Documentation Center.

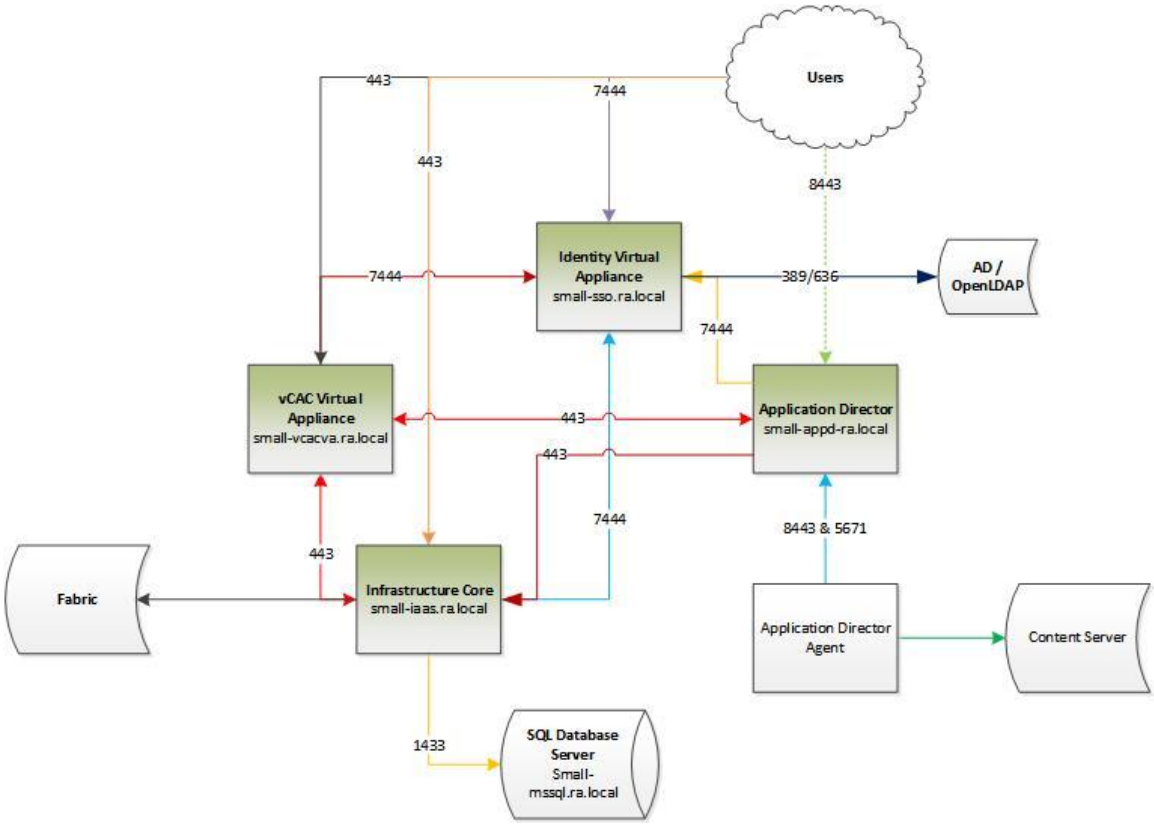


Figure 1. Minimal deployment footprint illustrating the components discussed in this document.

Secure Deployment

Verifying the Integrity of Installation Media

Verify the integrity of the installation media before you install the product.

Always verify the SHA1 hash after you download an ISO, offline bundle, or patch to ensure integrity and authenticity of the downloaded files. If you obtain physical media from VMware and the security seal is broken, return the software to VMware for a replacement.

After you download the media, use the MD5/SHA1 sum value to verify the integrity of the download. Compare the MD5/SHA1 hash output with the value posted on the VMware Web site. SHA1 or MD5 hash should match.

For more information about verifying the integrity of the installation media, see http://kb.vmware.com/kb/1537.

Hardening Infrastructure

Hardening the VMware vSphere Environment

vRealize Automation relies on a secure VMware vSphere environment to achieve the greatest benefits and a secured infrastructure.

Assess the VMware vSphere environment and verify that the appropriate level of vSphere hardening guidance is enforced and maintained.

For more guidance about hardening, see http://www.vmware.com/security/hardening-guides.html.

Verifying Hardening of the Infrastructure as a Service Host

Review the recommendations set out in the appropriate Windows hardening and secure best practice guidelines, and ensure that your Windows Server host is appropriately hardened. Not following the hardening recommendations might result in exposure to known security vulnerabilities from insecure components on Windows releases.

Currently supported versions are Windows Server 2008 R2, Windows Server 2012, and Windows Server 2012 R2.

Contact your Microsoft vendor about the correct guidance for hardening practices of Microsoft products.

Verifying Hardening Microsoft SQL Server

Review the recommendations set out in the appropriate Microsoft SQL Server hardening and secure best practice guidelines. Review all Microsoft security bulletins regarding the version of Microsoft SQL Server you are using. Not following the hardening recommendations might result in exposure to known security vulnerabilities from insecure components on Microsoft SQL Server versions.

To verify that your version is supported, see the vRealize Automation Support Matrix.

Contact your Microsoft vendor for guidance on hardening practices for Microsoft products.

Verifying Hardening Microsoft .NET

Review the recommendations set out in the appropriate .NET hardening and secure best practice guidelines. Review all Microsoft security bulletins regarding the version of Microsoft SQL Server you are using. Not following the hardening recommendations might result in exposure to known security vulnerabilities from insecure components on Microsoft .Net versions.

To verify that your version is supported, see the vRealize Automation Support Matrix.

Contact your Microsoft vendor for guidance on hardening practices for Microsoft products.

Verifying Hardening Microsoft Internet Information Services (IIS)

Review the recommendations set out in the appropriate Microsoft IIS hardening and secure best practice guidelines. Review all Microsoft security bulletins regarding the version of IIS you are using. Not following the hardening recommendations might result in exposure to known security vulnerabilities.

To verify that your version is supported, see the vRealize Automation Support Matrix.

Contact your Microsoft vendor for guidance on hardening practices for Microsoft products.

Ensuring that the VMware Supplied PostgreSQL is Hardened

vRealize Appliance and Application Services are distributed with hardened PostgreSQL. However, PostgreSQL hardening recommendations might change independently of the vRealize Automation release cycle. Review the most recent hardening recommendations. Not following the hardening recommendations might result in exposure to known security vulnerabilities on PostgreSQL.

VMware occasionally releases security advisories for various products. Being aware of these advisories can ensure that you have the safest underlying product and that the product is not vulnerable to known threats.

Review the recommendations set in the PostgreSQL hardening guides and best practices. For VMware supplied PostgreSQL, assess the vRealize Automation installation, patching, and upgrade history and ensure that the released VMware Security Advisories are followed and enforced.

For more information about the current VMware security advisories, see http://www.vmware.com/security/advisories/.

Supplying Your Own Hardened PostgreSQL for use with vRealize Automation Server

Review the recommendations set out in the appropriate PostgreSQL hardening and secure best practice guidelines. Review all PostgreSQL security bulletins regarding you version of PostgreSQL you are using. Not following the hardening recommendations may result in exposure to known security vulnerabilities from insecure components on PostgreSQL versions.

To verify that your version is supported, see the vRealize Automation Support Matrix.

For more information about PostgreSQL security alerts, see http://www.postgresql.org/support/security.html.

Reviewing Installed Software

Review software that is installed on hosts and evaluate its usage. Vulnerabilities in unused software might increase the risk to the system in unauthorized access and disruption of availability.

Do not install any software not required for the secure operation of the system on any of the vRealize Appliance hosts. Uninstall any unused or unrequired software.

Installing Unsupported Software

Installing unsupported, untested, or unapproved software on infrastructure products such as vRealize Automation is potentially dangerous. To minimize the threat to the infrastructure, do not install or use any third-party software that is not supported by VMware on VMware supplied hosts.

Assess the vRealize Automation deployment and inventory the installed products to verify that no unsupported software is installed.

For more information about the support policies for third-party products, see https://www.vmware.com/support/policies/thirdparty.html.

Using Third-Party Software

Do not use third-party software that is not supported by VMware. Verify that all third-party software is securely configured and patched in accordance with third-party vendor guidance.

Non-authenticity, insecurity, or unpatched vulnerabilities of third-party software that is installed on the host might put the operation of the system at risk to unauthorized access and disruption of availability. All software that is not supplied by VMware must be appropriately secured and patched.

If you must use third-party software that is not supported by VMware, consult the third-party vendor for secure configuration and patching requirements.

Following VMware Security Advisories and Applying Patches

VMware occasionally releases security advisories for various products. Being aware of these advisories can ensure that you have the safest underlying product and that the product is not vulnerable to known threats.

Assess the vRealize Automation and Application Services installation, patching, and upgrade history and verify that the released VMware Security Advisories are followed and enforced.

For more information about the current VMware security advisories, see http://www.vmware.com/security/advisories/.

Secure Configuration

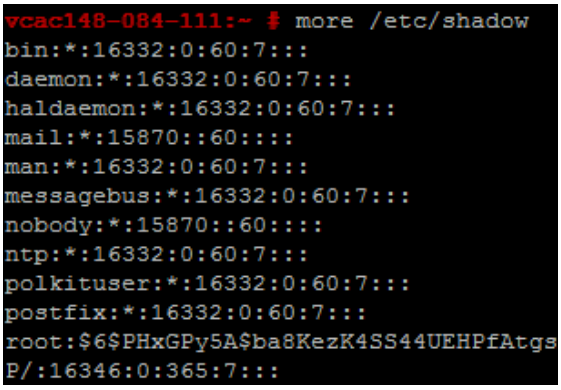
vRealize Appliance

Root Password

The root user password can be modified in the Admin tab of the VMware Appliance Management Infrastructure (VAMI) user interface.

To verify the hash of the root password, log in as root and run the following command:

```
# more /etc/shadow
```



If the account password starts with `$6$`, which is the standard hash for all hardened appliances, the password is using a sha512 hash. If the root password does not contain a sha512 hash, run the `passwd` command to change it.

All hardened appliances enable `enforce_for_root` for the pw_history module, found in `etc/pam.d/common-password`, so that the last five passwords are remembered by default. Old passwords are stored for each user in the `/etc/security/opasswd` file.

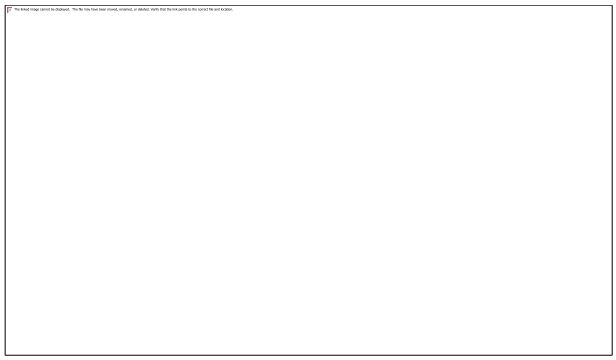
Password Expiry

All hardened appliances are set by default to create accounts with a 60-day password expiry. On most hardened appliances, the root account is set to a 365-day password expiry. It is highly recommended that you verify the expiry on all accounts to meet both security and operation requirements standards.

As part of your organization's compliance policies, a procedure should be implemented to ensure that administrators do not forget to change their passwords within the active period. If the root password expires, there is no method in the appliance to reinstate the root password. It is imperative that site-specific policies are implemented to prevent administrative and root passwords from expiring.

To verify the password expiry of all accounts, log in as root and run the following command:

```
# more /etc/shadow
```



The password expiry is the fifth field (fields are separated by a ‘: ‘column) of the shadow file. The root expiry is set in days.

To modify the expiry of the root account, log in as root and run the following command:

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

```
vcac148-084-111:~ # passwd -x 365 root
Password expiry information changed.
vcac148-084-111:~ #
```

The root password expiry is changed to 365 days. Use the same command to modify any user, substituting ‘root’ for the specific account, and replacing the number of days to meet the expiry standards of the organization.

Secure Shell, Administrative Accounts, and Console Access

For information, see Secure Shell, Administrative Accounts, and Console Access.

NTP Service

For critical time sourcing, disable host time synchronization and use NTP. NTP is recommended in production as a means to accurately track user actions and to realize potential malicious attacks and intrusion through accurate audit and log keeping.

The ntp daemon is included on the appliance, and is used to provide synchronized time services. The configuration file for NTP is located in `/etc/ntp.conf`.

You can enable the NTP service and add time servers in the Admin tab in the VAMI.

NTP Configuration

Verify the protection of the `/etc/ntp.conf` configuration file.

1. Set the file ownership to root:root.
2. Set the permissions to 0640.

To mitigate the risk of a Denial of Service amplification attack on the NTP service, ensure that the following lines appear in the `/etc/ntp.conf` file:

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

For information on NTP security notices, see http://support.ntp.org/bin/view/Main/SecurityNotice.

SSL/TLS

Ensure that the system is deployed with secure transmission channels.

Strong Protocols

Serious weaknesses have been identified with earlier SSL protocols, including SSLv2 and SSLv3. These protocols are no longer considered secure. The best practice for transport layer protection is to provide support for only the TLS protocols: TLS1.0, TLS 1.1 and TLS 1.2. vRealize Appliance disables SSLv2 and SSLv3 by default with the exception of RabbitMQ server where SSLv3 is not disabled by default. However, prior to production you should verify that SSLv2 and SSLv3 are disabled.

1. Verify that SSLv2 and v3 are disabled in apache2 https handler on the vRealize Appliance.

   Review the `/etc/apache2/vhosts.d/vcac.conf` file and verify that the following entry appears:

   ```
   SSLProtocol all -SSLv2 -SSLv3
   ```

2. Verify that SSLv2 and SSLv3 are disabled in the lighttpd https handler on the vRealize Appliance.

   Review the `/opt/vmware/etc/lighttpd/lighttpd.conf` file where `/opt/vmware/etc/lighttpd/conf.d/10-common.conf` settings are imported, and verify that the following entries appear:

   ```
   ssl.use-sslv2 = "disable"
   ssl.use-sslv3 = "disable"
   ```

   NOTE: When load balancing, ensure that all load balancers also have insecure protocols such as SSLv2 and SSLv3 disabled.

3. Verify that SSLv3 is disabled in the rabbitmq server on the vRealize Appliance.

   Review the file `/etc/rabbitmq/rabbitmq.config` and add the following line to ssl_options:

   ```
   {versions, ['tlsv1.2', 'tlsv1.1', tlsv1]}
   ```

Example:

In this example, the line you need to add appears in bold, and is the first line, and in the ssl_options section.

```
[
  {ssl, [{versions, ['tlsv1.2', 'tlsv1.1', tlsv1]}
        ]},
  {rabbit, [
     {ssl_listeners, [5672]},
     {tcp_listeners, []},
     {ssl_options, [{cacertfile,"/etc/rabbitmq/certs/ca/cacert.pem"},
                    {certfile,"/etc/rabbitmq/certs/server/cert.pem"},
                    {keyfile,"/etc/rabbitmq/certs/server/key.pem"},
                    {verify,verify_peer},
                    {fail_if_no_peer_cert,false},
                    {versions, ['tlsv1.2', 'tlsv1.1', tlsv1]},
                    ]},
     {heartbeat,30}
```

```
    ]}
].
```

4. Restart the RabbitMQ server by running the following command:

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

Default Supported Protocols

The vRealize Appliance supports the TLSv1.0 protocols by default.

Use Strong Ciphers

The encryption strength that is used in a TLS session is determined by the encryption cipher negotiated between the server and the browser. To make sure that only strong ciphers are selected, the server must be modified to disable the use of weak ciphers. Also, the ciphers should be configured in a suitable order. You should configure the server to support only strong ciphers and to use sufficiently large key sizes.

The following TLS ciphers are acceptable and enabled by default on the vRealize Appliance:

TLS_DHE_RSA_WITH_AES_128_CBC_SHA, TLS_DHE_RSA_WITH_AES_256_CBC_SHA, TLS_RSA_WITH_AES_128_CBC_SHA, TLS_RSA_WITH_AES_256_CBC_SHA

The preferred server cipher is TLS_DHE_RSA_WITH_AES_256_CBC_SHA.

Disable Weak Ciphers

Cipher suites that do not offer authentication such as NULL ciphersuites, aNULL or eNULL should be disabled. You should also disable anonymous Diffie-Hellman key exchange (ADH), export level ciphers (EXP, ciphers containing DES), key sizes smaller than 128 bits for encrypting payload traffic, the use of MD5 as a hashing mechanism for payload traffic, IDEA Cipher Suites, and RC4 cipher suites.

Disable Weak Ciphers in Apache

Disable the weak ciphers and enable strong ciphers on the vRealize Appliance in the apache2 https handler by reviewing the `/etc/apache2/vhosts.d/vcac.conf` file and ensuring that the following entry appears:

```
SSLCipherSuite HIGH:!aNULL!ADH:!EXP:!MD5:!3DES:!CAMELLIA:!PSK:!SRP:@STRENGTH
```

Disable Weak Ciphers in Lighttpd

Disable the weak ciphers and enable strong ciphers in the on the vRealize Appliance in the lighttpd https handler by reviewing the `/opt/vmware/etc/lighttpd/lighttpd.conf` file and ensuring that the following entry appears:

```
ssl.cipher-list = "TLSv1+HIGH: !SSLv2: !aNULL: !eNULL: !3DES: @STRENGTH"
```

Disable Weak Ciphers in RabbitMQ

To disable the weak ciphers and enable strong ciphers on the vRealize Appliance for the rabbitmq server, perform the following steps:

1. Evaluate the supported cipher suites by running the following command:

```
# /usr/sbin/rabbitmqctl eval 'ssl:cipher_suites().'
```

Example:

```
vcac_machine:~ # /usr/sbin/rabbitmqctl eval 'ssl:cipher_suites().'
[{dhe_rsa,aes_256_cbc,sha256},
 {dhe_dss,aes_256_cbc,sha256},
 {rsa,aes_256_cbc,sha256},
 {dhe_rsa,aes_128_cbc,sha256},
 {dhe_dss,aes_128_cbc,sha256},
 {rsa,aes_128_cbc,sha256},
 {dhe_rsa,aes_256_cbc,sha},
 {dhe_dss,aes_256_cbc,sha},
 {rsa,aes_256_cbc,sha},
 {dhe_rsa,'3des_ede_cbc',sha},
 {dhe_dss,'3des_ede_cbc',sha},
 {rsa,'3des_ede_cbc',sha},
 {dhe_rsa,aes_128_cbc,sha},
 {dhe_dss,aes_128_cbc,sha},
 {rsa,aes_128_cbc,sha},
 {rsa,rc4_128,sha},
 {rsa,rc4_128,md5},
 {dhe_rsa,des_cbc,sha},
 {rsa,des_cbc,sha}]
```

NOTE: The ciphers that are returned in this example represent only the ciphers that are supported. These ciphers are not used or advertised by the RabbitMQ server unless configured to do so in `rabbitmq.config`.

2. Select ciphers that are supported and that meet the security requirements for your organization.
   For example, if you want to allow the use of only rsa,aes_256_cbc,sha256 from the above list of available ciphers, perform the following step:
   - Review the file `/etc/rabbitmq/rabbitmq.config` and add the following line to ssl_options:

     ```
     {ciphers, [{rsa,aes_256_cbc,sha256}]}
     ```

3. Restart the RabbitMQ server by running the following command:

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

Example:

In this example, the line you need to add appears in bold, and is the last line in the ssl_options section.

```
[
  {ssl, [{versions, ['tlsv1.2', 'tlsv1.1', tlsv1]}
        ]},
[
  {ssl, [{versions, ['tlsv1.2', 'tlsv1.1', tlsv1]}
        ]},
  {rabbit, [
     {ssl_listeners, [5672]},
     {tcp_listeners, []},
     {ssl_options, [{cacertfile,"/etc/rabbitmq/certs/ca/cacert.pem"},
                    {certfile,"/etc/rabbitmq/certs/server/cert.pem"},
                    {keyfile,"/etc/rabbitmq/certs/server/key.pem"},
                    {verify,verify_peer},
                    {fail_if_no_peer_cert,false},
                    {versions, ['tlsv1.2', 'tlsv1.1', tlsv1]},
                    {ciphers, [{rsa,aes_256_cbc,sha256}]}
                   ]},
     {heartbeat,30}
   ]}
].
```

NOTE: Additional ciphers suites would be added: {ciphers, [{ciphersuite_1},{ciphersuite_2},{etc.}]}

Application Resources That Must Be Protected

Follow these procedures to ensure that application resources are protected.

1. Verify that the files that have the SUID and GUID bits set are well defined by running the following command:

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

   The following list should appear:

```
385054   36 -rwsr-xr-x   1 root     shadow      35688 Jan 13  2012 /sbin/unix_chkpwd ,
 385127   12 -rwsr-xr-x   1 root     shadow      10736 Dec 16  2011 /sbin/unix2_chkpwd ,
100469  816 -r-xr-sr-x   1 root     mail       829672 Feb 18  2010 /usr/sbin/sendmail ,
 335926   88 -rwsr-xr-x   1 root     shadow      85952 Feb  1  2012 /usr/bin/gpasswd ,
335927   20 -rwsr-xr-x   1 root     root        19416 Feb  1  2012 /usr/bin/newgrp ,
335924   84 -rwsr-xr-x   1 root     shadow      77848 Feb  1  2012 /usr/bin/chsh ,
335922   92 -rwsr-xr-x   1 root     shadow      86200 Feb  1  2012 /usr/bin/chage ,
```

```
335925   20 -rwsr-xr-x   1 root     shadow      19320 Feb  1  2012 /usr/bin/expiry ,

335928   84 -rwsr-xr-x   1 root     shadow      81856 Feb  1  2012 /usr/bin/passwd ,

335923   88 -rwsr-xr-x   1 root     shadow      82472 Feb  1  2012 /usr/bin/chfn ,

335951   40 -rwsr-x---   1 root     trusted     40432 Apr  2 23:33 /usr/bin/crontab ,

94272  232 -rwsr-xr-x   1 root     root       230072 Apr 30 08:38 /usr/bin/sudo ,

148679   16 -rwxr-sr-x   1 root     polkituser    14856 Oct 26  2012
/usr/lib/PolicyKit/polkit-read-auth-helper ,

148676   20 -rwxr-sr-x   1 root     polkituser    19008 Oct 26  2012
/usr/lib/PolicyKit/polkit-explicit-grant-helper ,

148677   20 -rwxr-sr-x   1 root     polkituser    19208 Oct 26  2012
/usr/lib/PolicyKit/polkit-grant-helper ,

148682   24 -rwsr-xr-x   1 polkituser root        23176 Oct 26  2012
/usr/lib/PolicyKit/polkit-set-default-helper ,

148681   24 -rwxr-sr-x   1 root     polkituser    23160 Oct 26  2012
/usr/lib/PolicyKit/polkit-revoke-helper ,

148678   12 -rwsr-x---   1 root     polkituser    10744 Oct 26  2012
/usr/lib/PolicyKit/polkit-grant-helper-pam ,

91452   28 -rwsr-xr-x   1 root     root        26897 Mar 18 21:30 /usr/lib64/pt_chown ,

196623   40 -rwsr-xr-x   1 root     root        40048 Apr 15  2011 /bin/ping ,

196664   40 -rwsr-xr-x   1 root     root        40016 Mar 18 23:59 /bin/su ,

196712   72 -rwsr-xr-x   1 root     root        69208 Apr 29 21:20 /bin/umount ,

196624   36 -rwsr-xr-x   1 root     root        35792 Apr 15  2011 /bin/ping6 ,

196711  100 -rwsr-xr-x   1 root     root        94776 Apr 29 21:20 /bin/mount ,

499900   48 -rwsr-x---   1 root     messagebus    47912 Oct 15  2012 /lib64/dbus-1/dbus-
daemon-launch-helper
```

2. Verify that all of the files on the VA have an owner, by running the following command.

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

   If no results appear, then all files have an owner.

3. Verify that none of the files are world writable files, by reviewing permissions of all files on the VA to ensure that note of the files include the permission xx6.

4. Verify that the correct files are owned only by the vcac user, by running the following command.

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

   If no results appear, then all correct files are owned only by the vcac user.

5. Verify that the correct files are writeable only by the vcac user, as shown in the following list.

   ```
   /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
/var/log/vcac
/var/lib/vcac/*
/var/cache/vcac/*
```

6. Verify that the readable only by the vcac user, as shown in the following list.

```
/etc/vcac/*
/var/log/vcac/*
/var/lib/vcac/*
/var/cache/vcac/*
```

7. Verify that the correct files are owned only by the vco user, as shown in the following list.

```
/etc/vco/*
/var/log/vco/*
/var/lib/vco/*
/var/cache/vco/*
```

8. Verify that the correct files are writeable only by the vco user, as shown in the following list.

```
/etc/vco/*
/var/log/vco/*
/var/lib/vco/*
/var/cache/vco/*
```

9. Verify that the correct files are readable only by the vco user, as shown in the following list.

```
/etc/vco/*
/var/log/vco/*
/var/lib/vco/*
/var/cache/vco/*
```

PostgreSQL

Client Authentication Configuration

The client authentication configuration settings can be found in `/storage/db/pgdata/pg_hba.conf`.

Configuring the system for local "trust" authentication allows any local user to connect as any PostgreSQL user, including the database super user without a password. To provide defense in depth and if you do not have significant trust in all local user accounts, use another authentication method. The method md5 is recommended since it sends encrypted passwords.

```
# TYPE  DATABASE        USER            ADDRESS                 METHOD

# "local" is for Unix domain socket connections only
local   all             all                                     trust
# IPv4 local connections:
host    all             all             127.0.0.1/32            md5
# IPv6 local connections:
host    all             all             ::1/128                 md5
# Allow replication connections from localhost, by a user with the
# replication privilege.
#local   replication     postgres                                trust
#host    replication     postgres        127.0.0.1/32            md5
#host    replication     postgres        ::1/128                 md5
host    all         all             0.0.0.0/0       md5
host    replication  all            0.0.0.0/0       md5
```

Figure 2. pg_hba.conf example.

If you edit the `pg_hba.conf` file, before any changes can take effect, you must restart the postgres server by running the following commands:

```
# cd /opt/vmware/vpostgres/9.2/bin

# su postgres

# ./pg_ctl restart –D /storage/db/pgdata/ -m fast
```

vcac User

vRealize Automation embedded Postgres database includes a default Postgres user account named vcac. You must change the vcac account password when using the embedded vRealize Automation database. After the vRealize Automation system is configured, perform the following steps:

To change the vcac user password:

1. ssh in to the vRealize Appliance that is designated for postgres DB.
2. Enter **`ssh root@10.20.136.161`**.
3. Enter **`su Postgres /opt/vmware/vpostgres/9.2/bin/psql`** to start the psql prompt.
4. Change the password for vcac user:

```
ALTER ROLE vcac with PASSWORD 'your-complex-password';
```

Postgres user

You might want to disable users from accessing Postgres as the default postgres user. You can configure this in `/storage/db/pgdata/pg_hba.conf`.

Distributing Postgres

If you choose to distribute your Postgres DB, you need to ensure that this appliance is also appropriately secured.

To verify that your version is supported, see the vRealize Automation Support Matrix.

When choosing to use the Postgres DB on the vRealize Appliance as a separate DB, you need to make changes to secure it. Follow all hardening recommendations for the vRealize Appliance and enable only the services and ports required for running the Postgres DB.

Disable Configuration Modes

When installing, configuring, or maintaining vRealize Automation you might want to change some configurations and settings to enable troubleshooting and debugging of your installation. You should catalog and audit each of the changes you make to ensure that these are properly secured following their required use. Do not put in to production if you are not sure that your configuration changes are correctly secured.

Session Timeout

The session timeout on user inactivity is 30 minutes by default. If you want to adjust this session time out to conform to your organization's security policy, make the following changes to the `web.xml` file.

1. Edit `/usr/lib/vcac/server/webapps/vcac/WEB-INF/web.xml`.
   Find `session-config` and set the session-timeout value.

```
<!-- 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>
```

2. Restart the Apache server.

```
/etc/init.d/apache2 stop

/etc/init.d/apache2 start
```

Identity Appliance

Root Password

The root user password can be modified in the Admin tab of the VMware Appliance Management Infrastructure (VAMI) user interface.

To verify the hash of the root password, log in as root and run the following command:

```
# more /etc/shadow
```

```
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$ba8KezK4SS44UEHPfAtgs
P/:16346:0:365:7:::
```

If the account password starts with `$6$`, which is the standard hash for all hardened appliances, the password is using a

sha512 hash. If the root password does not contain a sha512 hash, run the `passwd` command to change it.

All hardened appliances enable `enforce_for_root` for the pw_history module, found in `etc/pam.d/common-password`, so that the last five passwords are remembered by default. Old passwords are stored for each user in the `/etc/security/opasswd` file.

Password Expiry

All hardened appliances are set by default to create accounts with a 60-day password expiry. On most hardened appliances, the root account is set to a 365-day password expiry. It is highly recommended that you verify the expiry on all accounts to meet both security and operation requirements standards.

As part of your organization's compliance policies, a procedure should be implemented to ensure that administrators do not forget to change their passwords within the active period. If the root password expires, there is no method in the appliance to reinstate the root password. It is imperative that site-specific policies are implemented to prevent administrative and root passwords from expiring.

To verify the password expiry of all accounts, log in as root and run the following command:

```
# more /etc/shadow
```

```
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$ba8KezK4SS44UEHPfAtgsB6iy1nJYbBb
zRbYR1mvJiX.pJpub0AEpP/:16346:0:365:7:::
sshd:!:16332:0:60:7:::
```

The password expiry is the fifth field (fields are separated by a ': 'column) of the shadow file. The root expiry is set in days.

To modify the expiry of the root account, log in as root and run the following command:

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

```
vcac148-084-111:~ # passwd -x 365 root
Password expiry information changed.
vcac148-084-111:~ #
```

The root password expiry is changed to 365 days. Use the same command to modify any user, substituting 'root' for the specific account, and replacing the number of days to meet the expiry standards of the organization.

Secure Shell, Administrative Accounts, and Console Access

For information, see Secure Shell, Administrative Accounts, and Console Access.

NTP Service

For critical time sourcing, disable host time synchronization and use NTP. NTP is recommended in production as a means to accurately track user actions and to realize potential malicious attacks and intrusion through accurate audit and log keeping.

The ntp daemon is included on the appliance, and is used to provide synchronized time services. The configuration file

for NTP is located in `/etc/ntp.conf`.

You can enable the NTP service and add time servers under the Admin tab in the VAMI.

NTP Configuration

Verify the protection of the `/etc/ntp.conf` configuration file.

1. Set the file ownership to root:root.
2. Set the permissions to 0640.

To mitigate the risk of a Denial of Service amplification attack on the NTP service, ensure that the following lines appear in the `/etc/ntp.conf` file:

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

For information on NTP security notices, see http://support.ntp.org/bin/view/Main/SecurityNotice.

Strong Protocols

Serious weaknesses have been identified with earlier SSL protocols, including SSLv2 and SSLv3. These protocols are no longer considered secure. The best practice for transport layer protection is to provide support only for the TLS protocols: TLS1.0, TLS 1.1 and TLS 1.2.

Do not use SSLv2 as it is not secure.

Do not use SSLv3 as it is not secure.

The Identity Appliance disables SSLv2 and SSLv3 by default. However, prior to production you should verify that SSLv2 and SSLv3 are disabled.

1. Verify that SSLv2 and SSLv3 are disabled on the Identity Appliance in the tomcat https handler:

   Review the `/usr/lib/vmware-sts/conf/server.xml` file, find the connector configuration for https por, and verify that the following attribute appears:

   ```
   sslEnabledProtocols = "TLSv1,TLSv1.1,TLSv1.2"
   ```

2. Verify that SSLv2 and SSLv3 is disabled on the Identity Appliance in the lighttpd https handler:

   Review the `/opt/vmware/etc/lighttpd/conf.d/10-common.conf` file (these settings are imported into `/opt/vmware/etc/lighttpd/lighttpd.conf`), and verify that the following entries appear:

   ```
   ssl.use-sslv2                 = "disable"
   ssl.use-sslv3                 = "disable"
   ```

NOTE: When load balancing, ensure that all load balancers also have insecure protocols such as SSLv2 and SSLv3 disabled.

Default Supported Protocols

The Identity Appliance supports the TLSv1.0, TLSv1.1, TLSv1.2 protocols by default.

Use Strong Ciphers

The encryption strength that is used in a TLS session is determined by the encryption cipher negotiated between the server and the browser. To make sure that only strong ciphers are selected, the server must be modified to disable the use of weak ciphers. In addition, the ciphers should be configured in a suitable order. You should configure the server to support only strong ciphers and to use sufficiently large key sizes.

The following TLS ciphers are acceptable and should be enabled by default on the Identity Appliance.

For lighttpd:

TLS_DHE_RSA_WITH_AES_128_CBC_SHA, TLS_DHE_RSA_WITH_AES_256_CBC_SHA, TLS_RSA_WITH_AES_128_CBC_SHA, TLS_RSA_WITH_AES_256_CBC_SHA

Preferred server cipher - TLS_DHE_RSA_WITH_AES_256_CBC_SHA

For the tomcat http handler:

TLS_RSA_WITH_AES_128_CBC_SHA, TLS_RSA_WITH_AES_256_CBC_SHA

Preferred server cipher - TLS _RSA_WITH_AES_256_CBC_SHA

Disable Weak Ciphers

Cipher suites that do not offer authentication such as NULL ciphersuites, aNULL or eNULL should be disabled. You should also disable anonymous Diffie-Hellman key exchange (ADH), export level ciphers (EXP, ciphers containing DES), key sizes smaller than 128 bits for encrypting payload traffic, the use of MD5 as a hashing mechanism for payload traffic, IDEA Cipher Suites, and RC4 cipher suites.

Disable Weak Ciphers in Tomcat

Add or remove ciphers in the Identity Appliance in the tomcat https handler by reviewing the `/usr/lib/vmware-sts/conf/server.xml` file, finding the connector configuration for https port, and reviewing the following attribute:

```
ciphers="TLS_RSA_WITH_AES_128_CBC_SHA,TLS_RSA_WITH_AES_256_CBC_SHA"
```

Disable Weak Ciphers in Lighttpd

Disable the weak ciphers and enable strong ciphers in the on the vRealize Appliance in the lighttpd https handler by reviewing the `/opt/vmware/etc/lighttpd/lighttpd.conf` file and verifying that the following entry appears:

```
ssl.cipher-list = "TLSv1+HIGH: !SSLv2: !aNULL: !eNULL: !3DES: @STRENGTH"
```

Disable Configuration Modes

When installing, configuring, or maintaining vRealize Automation you might want to change some configurations and settings to enable troubleshooting and debugging of your installation. You should catalog and audit each of the changes you make to ensure that these are properly secured following their required use. Do not put in to production if you are not sure that your configuration changes are correctly secured.

Application Services Appliance

Root Password

To change the root password at the command line, enter the command `passwd` at the root shell of the appliance.

```
vcac148-084-111:~ # passwd
Changing password for root.
New password:
BAD PASSWORD: it is based on a dictionary word
BAD PASSWORD: is too simple
Retype new password:
Password changed.
vcac148-084-111:~ #
```

The root user bypasses the pam_cracklib module password complexity check, which is found in `etc/pam.d/common-password`. It is imperative that you manually verify that that the root password meets your organization's corporate password complexity requirements.

To verify the hash of the root password, log in as root and run the following command:

```
# more /etc/shadow
```

```
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$ba8KezK4SS44UEHPfAtgs
P/:16346:0:365:7:::
```

If the account password starts with 6, which is the standard hash for all hardened appliances, the password is using a sha512 hash. If the root password does not contain a sha512 hash, run the `passwd` command to change it.

All hardened appliances enable `enforce_for_root` for the pw_history module (found in `etc/pam.d/common-password`), so the last five passwords are remembered by default. Old passwords are stored for each user in the `/etc/security/opasswd` file.

Secure Shell, Administrative Accounts, and Console Access

For information, see Secure Shell, Administrative Accounts, and Console Access.

NTP Service

For critical time sourcing, disable host time synchronization and use NTP. NTP is recommended in production as a means to accurately track user actions and to realize potential malicious attacks and intrusion through accurate audit and log keeping.

The ntp daemon is included on the appliance, and is used to provide synchronized time services. The configuration file for NTP is located in `/etc/ntp.conf`.

You can enable the NTP service and add timeservers under the Admin tab in the VAMI.

NTP Configuration

Verify the protection of the `/etc/ntp.conf` configuration file.

1. Set the file ownership to root:root.
2. Set the permissions to 0640.

To mitigate the risk of a Denial of Service amplification attack on the NTP service, verify that the following lines appear in the `/etc/ntp.conf` file:

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

For information on NTP security notices, see http://support.ntp.org/bin/view/Main/SecurityNotice.

SSL/TLS

Ensure that the system is deployed with secure transmission channels.

Strong Protocols

Serious weaknesses have been identified with earlier SSL protocols, including SSLv2 and SSLv3. These protocols are no longer considered secure. The best practice for transport layer protection is to provide support only for the TLS protocols: TLS1.0, TLS 1.1 and TLS 1.2.

Do not use SSLv2 as it is not secure.

Do not use SSLv3 as it is not secure.

The Application Services Appliance disables SSLv2 and SSLv3 by default. However, prior to production you should verify that SSLv2 and SSLv3 are disabled.

Verify that SSLv2 and SSLv3 are disabled on the Application Services Appliance in the tomcat https handler by reviewing the `/home/darwin/tcserver/darwin/conf/server.xml` file, find the connector configuration for https port 8443, and verify that the following attribute appears:

```
sslEnabledProtocols = "TLSv1,TLSv1.1,TLSv1.2"
```

Default Supported Protocols

The vRealize Appliance supports the TLSv1.0 protocol by default.

Use Strong Ciphers

The encryption strength that is used in a TLS session is determined by the encryption cipher negotiated between the server and the browser. To make sure that only strong ciphers are selected, the server must be modified to disable the use of weak ciphers. Also, the ciphers should be configured in a suitable order. You should configure the server to only support strong ciphers and to use sufficiently large key sizes.

The following TLS ciphers are acceptable and enabled by default on the vRealize Appliance:

TLS_DHE_RSA_WITH_AES_128_CBC_SHA, TLS_DHE_RSA_WITH_AES_256_CBC_SHA, TLS_RSA_WITH_AES_128_CBC_SHA, TLS_RSA_WITH_AES_256_CBC_SHA

The preferred server cipher is TLS_DHE_RSA_WITH_AES_256_CBC_SHA.

Disable Weak Ciphers

Cipher suites that do not offer authentication such as NULL ciphersuites, aNULL or eNULL should be disabled. You should also disable anonymous Diffie-Hellman key exchange (ADH), export level ciphers (EXP, ciphers containing DES), key sizes smaller than 128 bits for encrypting payload traffic, the use of MD5 as a hashing mechanism for payload traffic, IDEA Cipher Suites, and RC4 cipher suites.

Reviewing Ciphers

Review the `/home/darwin/tcserver/darwin/conf/server.xml` file, find the connector configuration for https port 8443, and verify the cipher suites listed in the following attribute: ciphers="…."

Ensure that the ciphers listed are not weak, see Disable Weak Ciphers.

Unrequired Software Components

Application Services contains a content server that has unpatched example content. To prevent misuse, this needs to be secured prior to production. Consult the Application Services installation documentation for the correct setup of a secure content server.

Disable Configuration Modes

When installing, configuring, or maintaining vRealize Automation you might want to change some configurations and settings to enable troubleshooting and debugging of your installation. You should catalog and audit each of the changes you make to ensure that these are properly secured following their required use. Do not put in to production if you are not sure that your configuration changes are correctly secured.

Infrastructure as a Service Component

Windows Time Service

For critical time sourcing, you should disable host time synchronization and use authorized time servers. This is recommended in production as a means to accurately track user actions and to identify potential malicious attacks and intrusion through accurate auditing and logging.

SSL/TLS

Ensure that the system is deployed with secure transmission channels.

Strong Protocols

Serious weaknesses have been identified with earlier SSL protocols, including SSLv2 and SSLv3. These protocols are no longer considered secure. The best practice for transport layer protection is to provide support only for the TLS protocols: TLS1.0, TLS 1.1 and TLS 1.2.

Do not use SSLv2 as it is not secure.

Do not use SSLv3 as it is not secure.

Prior to production, you should verify that SSLv2 and SSLv3 are disabled. To disable SSLv2 and SSLv3 on the Infrastructure Services component, you must disable these in Internet Information Services (IIS) on the Windows host machine. See Microsoft KB for information on how to disable SSLv3 in IIS: https://support.microsoft.com/kb/187498/en-us.

NOTE: When load balancing, ensure that all load balancers also have insecure protocols such as SSLv2 and SSLv3 disabled.

Use Strong Ciphers

The encryption strength that is used in a TLS session is determined by the encryption cipher negotiated between the server and the browser. To make sure that only strong ciphers are selected, the server must be modified to disable the use of weak ciphers. In addition, the ciphers should be configured in a suitable order. You should configure the server to support only strong ciphers and to use sufficiently large key sizes.

The following TLS ciphers are acceptable for communicating with the other vRealize components:

TLS_DHE_RSA_WITH_AES_128_CBC_SHA, TLS_DHE_RSA_WITH_AES_256_CBC_SHA
TLS_RSA_WITH_AES_128_CBC_SHA, TLS_RSA_WITH_AES_256_CBC_SHA

Disable Weak Ciphers

Cipher suites that do not offer authentication such as NULL ciphersuites, aNULL or eNULL should be disabled. You should also disable anonymous Diffie-Hellman key exchange (ADH), export level ciphers (EXP, ciphers containing DES), key sizes smaller than 128 bits for encrypting payload traffic, the use of MD5 as a hashing mechanism for payload traffic, IDEA Cipher Suites, and RC4 cipher suites.

Disable Configuration Modes

When installing, configuring, or maintaining vRealize Automation you might want to change some configurations and settings to enable troubleshooting and debugging of your installation. You should catalog and audit each of the changes you make to ensure that these are properly secured following their required use. Do not put in to production if you are not sure that your configuration changes are correctly secured.

Verify Host Server's Secure Baseline

You can use the Microsoft Baseline Security Analyzer (MBSA) to quickly identify that your server has the latest updates or hotfixes. You can use MBSA to install any missing security patches from Microsoft to keep your server up-to-date with Microsoft security recommendations. You can download this tool from Microsoft.

Verify Host Server Is Securely Configured

You can use the Windows Security Configuration Wizard (SCW) and the Microsoft Security Compliance Manager toolkit to verify that the host server is securely configured.

You can start SCW from the administrative tools from you Windows server. This tool can identify the roles of your server and the installed features including networking, Windows firewalls, and registry settings. Compare the report with the latest hardening guidance delivered from the relevant Microsoft Security Compliance Manager for your Windows server. Based on the results, you can fine tune security settings for each feature such as network services, account settings, and Windows firewalls, and apply the settings to your server.

Application Resources That Must Be Protected

Review the files in the following table against your installation. In this table where a directory is listed, this means that every file in that directory or in its sub-directories should have these same settings, unless stated otherwise elsewhere in the table.

Directory or File	Groups or Users	Full Control	Modify	Read & Execute	Read	Write
`VMware\vCAC\Agents\<agent_name>\logs`	SYSTEM	X	X	X	X	X
	Administrator	X	X	X	X	X

	Administrators	X	X	X	X	X
VMware\vCAC\Agents\<agent_name>\temp	SYSTEM	X	X	X	X	X
	Administrator	X	X	X	X	X
	Administrators	X	X	X	X	X
VMware\vCAC\Agents\	SYSTEM	X	X	X	X	X
	Administrators	X	X	X	X	X
	Users			X	X	
VMware\vCAC\Distributed Execution Manager\	SYSTEM	X	X	X	X	X
	Administrators	X	X	X	X	X
	Users			X	X	
VMware\vCAC\Distributed Execution Manager\DEM\Logs	SYSTEM	X	X	X	X	X
	Administrator	X	X	X	X	X
	Administrators	X	X	X	X	X
VMware\vCAC\Distributed Execution Manager\DEO\Logs	SYSTEM	X	X	X	X	X
	Administrator	X	X	X	X	X
	Administrators	X	X	X	X	X
VMware\vCAC\Management Agent\	SYSTEM	X	X	X	X	X
	Administrators	X	X	X	X	X
	Users			X	X	
VMware\vCAC\Server\	SYSTEM	X	X	X	X	X
	Administrators	X	X	X	X	X
	Users			X	X	
VMware\vCAC\Web API	SYSTEM	X	X	X	X	X
	Administrators	X	X	X	X	X
	Users			X	X	

Additional Secure Configuration Activities

Verify Server User Account Settings

Verify that for local and domain user accounts and settings that there are no unnecessary user accounts. Any user account not related to the functioning of the application should be restricted to those accounts required for administration, maintenance, and troubleshooting. Remote access from domain user accounts should be restricted to the minimal required to maintain the server and be strictly controlled and activity audited.

Delete and Disable Unnecessary Applications

Delete all unnecessary applications from the host servers. Each additional and unnecessary application increases the risk of exposure due to their unknown or unpatched vulnerabilities.

Disable All Unnecessary Ports and Services

Verify the host server's Windows firewall for the list of open ports. Block all of the ports that are not listed as a minimum requirement for the IaaS component in the ports and protocols section of this document, and are not required. In addition, audit the services running against your host server and disable those that are not required.

Network Security and Secure Communication

Network Settings for VMware Appliances

Prevent User Control

Manipulating network interfaces can result in bypassing network security mechanisms or denial of service. The ability to change network interface settings should be restricted to privileged users. You should ensure that network interfaces are not configured for user control.

Verify user control settings, by running the following command:

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

Make sure that each interface is set to NO.

TCP Backlog Queue Size

To provide mitigation for TCP denial or service attacks, the TCP backlog queue sizes should be set to a default size.

Verify that the system sets a default TCP backlog queue size, by running the following command:

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

The recommended default setting is 1280.

To set the default TCP backlog queue size, add the following entry to the `/etc/sysctl.conf` file:

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

Deny ICMPv4 Echoes to Broadcast Address

Responses to broadcast Internet Control Message Protocol (ICMP) echoes provides an attack vector for amplification attacks and can assist malicious agents in network mapping.

Verify that the system is not sending responses to ICMP echo requests to the broadcast address, by running the following command:

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

If this value is not 1, then add the following entry to the `/etc/sysctl.conf` file to disable the echo broadcast address:

```
net.ipv4.icmp_echo_ignore_broadcasts=1
```

Disable IPv4 Proxy ARP

IPv4 Proxy ARP allows a system to send responses to ARP requests on one interface on behalf of hosts connected to another interface. If enabled, addressing information might be leaked between the attached network segments.

If IPv4 Proxy ARP is not required, verify that the system disables Proxy ARP by running the following command:

```
# grep [01] /proc/sys/net/ipv4/conf/*/proxy_arp|egrep "default|all"
```

If both values are not zero, add the following entries to the `/etc/sysctl.conf` file to set the value to 0:

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

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

Ignore IPv4 ICMP Redirect Messages

A malicious ICMP redirect message can allow a man-in-the-middle attack to occur. Routers use ICMP redirect messages to tell hosts that a more direct route exists for a destination. These messages modify the host's route table and are unauthenticated.

Verify that the system ignores IPv4 ICMP redirect messages by running the following command:

```
# grep [01] /proc/sys/net/ipv4/conf/*/accept_redirects|egrep "default|all"
```

If the value for both are not set to zero, add the following entries to the `/etc/sysctl.conf` file to set the value to 0:

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

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

Ignore IPv6 ICMP Redirect Messages

A malicious ICMP redirect message can allow a man-in-the-middle attack to occur. Routers use ICMP redirect messages to tell hosts that a more direct route exists for a destination. These messages modify the host's route table and are unauthenticated.

Verify that the system ignores IPv6 ICMP redirect messages by running the following command:

```
# grep [01] /proc/sys/net/ipv6/conf/*/accept_redirects|egrep "default|all"
```

If the value for both are not set to zero, add the entries to the `/etc/sysctl.conf` file to set the value to 0:

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

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

Deny IPv4 ICMP Redirects

ICMP redirect messages are used by routers to inform servers that a more direct route exists for a particular destination. These messages contain information from the system's route table that could reveal portions of the network topology.

Verify that the system denies IPv4 ICMP redirects by running the following command:

```
# grep [01] /proc/sys/net/ipv4/conf/*/send_redirects|egrep "default|all"
```

If the value for both are not set to zero, add the following entries to the `/etc/sysctl.conf` file to set the value to 0:

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

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

Log IPv4 Martian Packets

Martian packets are packets that contain addresses that are known by the system to be invalid. Logging these messages allows the identification of misconfigurations or attacks in progress.

Verify that the logs IPv4 martian packets by running the following command:

```
# grep [01] /proc/sys/net/ipv4/conf/*/log_martians|egrep "default|all"
```

If the value for both are not set to one, add the following entries to the `/etc/sysctl.conf` file to set the value to 1:

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

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

IPv4 Reverse Path Filtering

Reverse-path filtering provides protection against spoofed source addresses by causing the system to discard packets that have source addresses for which the system has no route, or if the route does not point towards the interface on which the packet arrived. Reverse-path filtering should be used whenever possible. Depending on the role of the system, reverse-path filtering might cause legitimate traffic to be discarded and, therefore, should be used in a more permissive mode or not at all.

Verify that the system uses IPv4 reverse path filtering by running the following command:

```
# grep [01] /proc/sys/net/ipv4/conf/*/rp_filter|egrep "default|all"
```

If the value for both are not set to one, add the following entries to the `/etc/sysctl.conf` file to set the value to 1:

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

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

Deny IPv4 forwarding

If the system is configured for IP forwarding and is not a designated router, it could be used to bypass network security by providing a path for communication not filtered by network devices.

Verify that the system denies IPv4 forwarding by running the following command:

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

If the value is not set to zero, add the following entry to the /etc/sysctl.conf file to set the value to 0:

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

Deny IPv6 Forwarding

If the system is configured for IP forwarding and is not a designated router, it could be used to bypass network security by providing a path for communication not filtered by network devices.

Verify that the system denies IPv6 forwarding by running the following command:

```
# grep [01] /proc/sys/net/ipv6/conf/*/forwarding|egrep "default|all"
```

If the value for both are not set to zero, add the following entries to the `/etc/sysctl.conf` file to set the value to 0:

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

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

IPv4 TCP Syncookies

A TCP SYN flood attack might cause a denial of service by filling a system's TCP connection table with connections in the SYN_RCVD state. Syncookies are used to not track a connection until a subsequent ACK is received, verifying that the initiator is attempting a valid connection and is not a flood source. This technique does not operate in a fully standards-compliant manner, but is only activated when a flood condition is detected, and allows defence of the system while continuing to service valid requests.

Verify that the system uses IPv4 TCP syncookies by running the following command:

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

If the value is not set to one, add the following entry to the `/etc/sysctl.conf` file to set the value to 1:

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

Deny IPv6 Router advertisements

A feature of IPv6 is how systems can configure their networking devices using information from the network automatically. From a security perspective, manually configuring important configuration information is preferable to accepting it from the network in an unauthenticated fashion.

Verify that the system denies the acceptance of router advertisements and ICMP redirects unless necessary by running the following command:

```
# grep [01] /proc/sys/net/ipv6/conf/*/accept_ra|egrep "default|all"
```

If the value for both are not set to zero, add the following entries to the `/etc/sysctl.conf` file to set the value to 0:

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

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

Deny IPv6 Router Solicitations

The router solicitations setting determines how many router solicitations are sent when bringing up the interface. If addresses are statically assigned, there is no need to send any solicitations.

Verify that system denies IPv6 router solicitations unless necessary by running the following command:

```
# grep [01] /proc/sys/net/ipv6/conf/*/router_solicitations|egrep "default|all"
```

If the value for both are not set to zero, add the following entries to the `/etc/sysctl.conf` file to set the value to 0:

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

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

Deny IPv6 Router Preference in Router Solicitations

The router preference in solicitations setting determines router preferences. If addresses are statically assigned, there is no need to receive any router preference for solicitations.

Verify that the system denies IPv6 router solicitations unless necessary by running the following command:

```
# grep [01] /proc/sys/net/ipv6/conf/*/accept_ra_rtr_pref|egrep "default|all"
```

If the value for both are not set to zero, add the following entries to the `/etc/sysctl.conf` file to set the value to 0:

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

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

Deny IPv6 Router Prefix

The accept ra pinfo setting controls whether the system will accept prefix info from the router. If addresses are statically assigned, there is no need to receive any router prefix information.

Verify that the system denies IPv6 router prefix information unless necessary by running the following command:

```
# grep [01] /proc/sys/net/ipv6/conf/*/accept_ra_pinfo|egrep "default|all"
```

If the value for both are not set to zero, add the following entries to the `/etc/sysctl.conf` file to set the value to 0:

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

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

Deny IPv6 Router Advertisement Hop Limit Settings

The accept_ra_defrtr setting controls whether the system will accept Hop Limit settings from a router advertisement. Setting it to 0 prevents a router from changing your default IPv6 Hop Limit for outgoing packets.

Verify that the system denies IPv6 router hop limit settings unless necessary by running the following command:

```
# grep [01] /proc/sys/net/ipv6/conf/*/accept_ra_defrtr|egrep "default|all"
```

If the value for both are not set to zero, add the following entries to the `/etc/sysctl.conf` file to set the value to 0:

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

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

Deny IPv6 Router Advertisement Autoconf Settings

The autoconf setting controls whether router advertisements can cause the system to assign a global unicast address to an interface.

Verify that the system denies IPv6 router autoconf settings unless necessary by running the following command:

```
# grep [01] /proc/sys/net/ipv6/conf/*/autoconf|egrep "default|all"
```

If the value for both are not set to zero, add the following entries to the `/etc/sysctl.conf` file to set the value to 0:

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

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

Deny IPv6 Neighbor Solicitations

The dad_transmits setting determines how many neighbor solicitations to send out per address (global and link-local) when bringing up an interface to ensure the desired address is unique on the network.

Verify that the system denies IPv6 neighbor solicitations unless necessary by running the following command:

```
# grep [01] /proc/sys/net/ipv6/conf/*/dad_transmits|egrep "default|all"
```

If the value for both are not set to zero, add the following entries to the `/etc/sysctl.conf` file to set the value to 0:

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

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

Restrict IPv6 Max Addresses

The max addresses setting determines how many global unicast IPv6 addresses can be assigned to each interface. The default is 16, but it should be set to exactly the number of statically configured global addresses required.

Verify that the system restricts IPv6 max address setting to one unless more are necessary by running the following command:

```
# grep [1] /proc/sys/net/ipv6/conf/*/max_addresses|egrep "default|all"
```

If the value for both are not set to one, add the following entries to the `/etc/sysctl.conf` file to set the value to 1:

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

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

Network Settings for IaaS Component

See the Secure Configuration for Infrastructure as a Service Component section of this document.

Ports and Protocols

It is recommended that you disable all non-essential ports and protocols. The following table lists the minimum incoming and outgoing ports required for vRealize Automation to operate.

vRealize Automation Appliance

Minimum Required Incoming Ports

PORT	PROTOCOL	COMMENTS
111	TCP, UDP	RPC
443	TCP	Access to the vRealize Automation console and API calls
5480	TCP	Access to the virtual appliance web management interface
5488,5489	TCP	Internal. Used by vRealize Automation Appliance for updates
5672	TCP	RabbitMQ messaging
8230, 8280, 8281	TCP	Internal vCenter Orchestrator instance

Optional Incoming Ports

PORT	PROTOCOL	COMMENTS
22	TCP	Optional. SSH. The SSH service listening on port 22, or any other port, should be disabled in a production environment, and port 22 should be closed.
80	TCP	Optional. Redirects to 443.

Minimum Required Outgoing Ports

PORT	PROTOCOL	COMMENTS
25,587	TCP, UDP	SMTP for sending outbound notification emails
53	TCP, UDP	DNS
67, 68, 546, 547	TCP, UDP	DHCP
110, 995	TCP, UDP	POP for receiving inbound notification emails
143, 993	TCP, UDP	IMAP for receiving inbound notification emails
443	TCP	IaaS Manager Service over HTTPS
7444	TCP	Communication with SSO service over HTTPS

Optional Outgoing Ports

PORT	PROTOCOL	COMMENTS
80	TCP	Optional. For fetching software updates. Updates can be downloaded separately and applied.
123	TCP, UDP	Optional. For connecting directly to NTP instead of using host time.
5433	TCP, UDP	Optional. For communicating with standalone PostgreSQL database.
8281	TCP	Optional. For communicating with an external vCenter Orchestrator instance.

Identity Appliance

Minimum Required Incoming Ports

PORT	PROTOCOL	COMMENTS
5480	TCP	Access to virtual appliance web management interface
7444	TCP	SSO service over HTTPS

Optional Incoming Ports

PORT	PROTOCOL	COMMENTS
22	TCP	Optional. SSH. The SSH service listening on port 22, or any other port, should be disabled in a production environment, and port 22 should be closed.

Application Services Appliance

Minimum Required Incoming Ports

PORT	PROTOCOL	COMMENTS
8443	TCP	HTTPS for App Services UI
5671		AMQP over SSL for RabbitMQ
80	TCP	HTTP (used to host OOB demo content, agent binary, and CLI binary). Do not use OOB content in production environments.

Optional Incoming Ports

PORT	PROTOCOL	COMMENTS
22	TCP	Optional. SSH. The SSH service listening on port 22, or any other port, should be disabled in a production environment, and port 22 should be closed.

Infrastructure as a Service Components

Minimum Required Incoming Ports

COMPONENT	PORT	PROTOCOL	COMMENTS
Manager Service	443*	TCP	Communication with IaaS components and vRealize Automation Appliance over HTTPS

* Any virtualization hosts managed by proxy agents must also have TCP port 443 open for incoming traffic.

Minimum Required Outgoing Ports

COMPONENT	PORT	PROTOCOL	COMMENTS
All	53	TCP, UDP	DNS
All		TCP, UDP	DHCP
Manager Service	443	TCP	Communication with vRealize Automation Appliance over HTTPS
Web site	443	TCP	Communication with Manager Service over HTTPS
Distributed Execution Managers	443	TCP	Communication with Manager Service over HTTPS
Proxy Agents	443	TCP	Communication with Manager Service and virtualization hosts over HTTPS

Guest Agent	443	TCP	Communication with Manager Service over HTTPS
Manager Service, Web site	1433	TCP	MSSQL

Optional Outgoing Ports

COMPONENT	PORT	PROTOCOL	COMMENTS
All	123	TCP, UDP	Optional. NTP

Auditing and Logging

The detailed implementation of auditing and logging is outside the scope of this document. However, the following recommendations should be taken into consideration.

Remote logging to a central log host provides a secure store for logs. By gathering log files onto a central host, you can more easily monitor the environment with a single tool. You can also do aggregate analysis and searching to look for such things as coordinated attacks on multiple entities within the infrastructure. Logging to a secure, centralized log server can helps prevent log tampering and also provides a long-term audit record.

Ensure Remote Logging Server is Secure

Ensuring that the remote logging server is authorized and secure is very important. Often, once an attacker has managed to breach the security of your host machine, they will search for and endeavor to tamper with logs to cover their tracks and maintain their control without being discovered.

Use an Authorized NTP Server

By ensuring that all systems use the same relative time source, including the relevant localization offset, and that the relative time source can be correlated to an agreed-upon time standard such as Coordinated Universal Time (UTC), you can make it simpler to track and correlate an intruder's actions when reviewing the relevant log files. Incorrect time settings can make it difficult to inspect and correlate log files to detect attacks, and can make auditing inaccurate.

- Use at the least three NTP servers from outside time sources.

OR

- Configure a few local NTP servers on a trusted network that in turn obtain their time from at least three outside time source.

Secure Shell, Administrative Accounts, and Console Access

For remote connections, all hardened appliances include the Secure Shell (SSH).

It is highly recommended that you disable Secure Shell (SSH) in a production environment. If it is required for diagnosing or troubleshooting, you should enable it only for that particular purpose and only while it is needed. Make sure that you follow your organization's security policies.

SSH is an interactive command line environment that is available for making remote connections to the vRealize Appliance. Access using SSH requires high-privileged user account credentials. The activities performed from SSH generally bypass the RBAC and audit controls of the vRealize Appliance. SSH should be turned on only when needed to troubleshoot problems that cannot be resolved by using other procedures.

Depending on your vSphere configuration, you can choose to enable or disable SSH when you deploy your OVF. You can also disable SSH in the Admin tab of the VAMI user interface.

If you enable SSH, it needs to be appropriately protected against attack and enabled only for as long as is required.

A simple test is to try opening a connection by using SSH. If the connection opens and requests credentials, then SSH is enabled and is available for making connections.

To enable or disable SSH on the vRealize Appliance, go to the Web Management Console and click the Admin tab.

SSH root User

Do not allow SSH as root. Because appliances do not include default user accounts, the root account might still be able to directly log in by using SSH. To meet the compliance standards for non-repudiation, the SSH server on all hardened appliances comes preconfigured with the **AllowGroups wheel** entry to restrict SSH access to the secondary group wheel.

For separation of duties, the **AllowGroups wheel** entry can be modified in `/etc/ssh/sshd_config` to use another group such as sshd. The wheel group is enabled with the pam_wheel module for su access, so members of the wheel group are allowed to su-root, where the root password is required. Group separation provides a method for users to SSH to the appliance, but not to su to root. Do not remove or modify other entries in the **AllowGroups** field, which ensures proper appliance functionality. After making a change, you must restart the SSH daemon by running the command: `# service sshd restart`.

Before you remove the root SSH access, create local administrative accounts that can be used as ssh and/or are members of the secondary wheel group.

1. Create a local account on the appliance.

   a. Log in as root and enter the following command:

   `# useradd -d /home/vcacuser vcacuser -g users -G wheel -m`

   Where wheel is the group specified in AllowGroups for ssh access. To add multiple secondary groups, use –G wheel,sshd.

   `# passwd` *`username`*

2. Switch to the user to provide a new password so that password complexity checking is enforced.

   `# su - username`

   `username``@hostname:~>passwd`

If the password complexity is met, the password change is successful. If the password complexity is not met, the password reverts to the original password. Rerun the command to set a compliant password for the user.

After login accounts to allow SSH remote access and wheel access (su-root) are created, the root account can be removed from SSH direct login.

Before disabling direct root access, test that authorized administrators can access SSH by using AllowGroups, and that they can su to root by using the wheel group.

To remove direct login to SSH, modify the `/etc/ssh/sshd_config` file and make the following replacements:

- Replace **(#)PermitRootLogin yes** with **PermitRootLogin no**
- Enter `# service sshd restart` to restart the sshd service.

Alternatively, to enable or disable ssh login as root on the vRealize Appliance, navigate to the Web Management Console (VAMI), click the Admin tab, and select or deselect the **Administrator SSH login enabled** check box.

SSH Restricting Access

SSH access should also be restricted with the proper entries to limit access. All VMware virtual appliances include the tcp_wrappers package to allow tcp supported daemons to control the network subnets that can access the libwrapped daemons. By default, the `/etc/hosts.allow` file contains a generic entry, `Sshd: ALL : ALLOW`, that allows all access to the secure shell.

The generic entry should be changed for production environments to include only the localhost entries and the management networn subnet for secure operations, as shown in the following example:

```
Sshd:127.0.0.1 : ALLOW

Sshd: [::1] : ALLOW

Sshd: 10.0.0. :ALLOW
```

In this example, all localhost connections and connections made by clients on the 10.0.0.0 subnet are allowed.

Make sure to add all of the representation with which a machine can be identified, for example, hostname, IP add, FQDN, loopback, and so on.

SSH Key File Permissions

The following SSH key file permissions must be maintained:

Verify that the public host key files, located in `/etc/ssh/*key.pub`, are owned by root, group owned by root, and have permissions set to 0644. (-rw-r--r--)

Verify that the private host key files, located in `/etc/ssh/*key`, are owned by root, group owned by root, and have permissions set to 0600. (-rw-------)

SSH Port

By default, the SSHD listens on Port 22. You can choose not to change this. But if there were attackers looking for possible connections to break into, they would first be looking for the most common ports, like 22. Changing the port number considerably reduces the number of automated attacks performed by systematic attackers or Zombie Computers. On the other hand, changing the port number forces the configuration of this alternative port on all of the clients that want to connect to you. To change the listening port number, log in as root and edit `/etc/ssh/sshd_config` to change: #Port 22 to Port <preferred port number>. Then restart the sshd service by running `# service sshd restart`.

NOTE: Remember to update your firewall settings accordingly to allow the new port.

SSH Server Configuration

Where possible, the vRealize Appliance ships with a default hardened state, and many of these recommendations are already set. The following recommendations are provided to harden the server and client service in the global options section of the configuration file.

Open the server configuration file from `/etc/ssh/sshd_config` and verify the following settings:

- Ensure that the daemon is configured to run version 2 of the protocol by using the entry **Protocol 2**.
- Ensure that the CBC ciphers are not used by verifying that Ciphers aes256-ctr, aes128-ctr are the only ciphers listed.
- Ensure that the TCP forwarding is disabled by using the entry **AllowTCPForwarding no**.
- Ensure that the Gateway Ports are disabled by using the entry **GatewayPorts no**.
- Ensure that X11 forwarding is disabled by using the entry **X11Forwarding no**.
- Restrict the use of the SSH service by using the **AllowGroups** field, and specifying a group to allow access and add members to the secondary group for users permitted to use the service.
- Ensure that GSSAPI authentication is disabled by using the entry **GSSAPIAuthentication no** if unused.
- Ensure that Kerberos authentication is disabled by using the entry **KerberosAuthentication no** if unused.
- Ensure that local variables are set to "**disabled by commenting out**" or "**enabled for only LC_* or LANG variables**". This can be verified by finding the property named **AcceptEnv** and its value as mentioned earlier.
- Ensure that tunnels are disabled by using the entry **PermitTunnel no**.
- Ensure that network sessions are limited to a single session by using the entry **MaxSessions 1**.
- Restrict per user concurrent connection to 1, from root or any other user. `/etc/security/limits.conf` also needs to be changed.
- Ensure that strict mode checking is enabled by using the entry **StrictModes yes**.
- Ensure that privilege separation is enabled by using the entry **UsePrivilegeSeparation yes**.
- Ensure that rhosts RSA authentication is disabled by using the entry **RhostsRSAAuthentication no**.
- Ensure that compression is either delayed or disabled by using the entry **Compression delayed** or **Compression no**.
- Ensure that '**Message authentication code**' is used by using the entry **MACs hmac-sha1**.
- Ensure that disabling users to bypass access restriction by using the entry **PermitUserEnvironment no**.

If possible, restrict the use of the SSH server to a management subnet in the `/etc/hosts.allow` file.

SSH Client Configuration

Verify that the following settings are made in the global options section of the configuration file for the SSH client, which is located in `/etc/ssh/ssh_config`.

- Ensure that the client is configured to run version 2 of the protocol by using the entry Protocol 2.
- Ensure that the client Gateway Ports are disabled by using the entry **GatewayPorts no**.
- Ensure that the GSSAPI authentication is disabled by using the entry **GSSAPIAuthentication no**.
- Ensure that only locale variables are set by finding the SendEnv global option and verify that only LC_* or LANG variables are provided.
- Ensure CBC ciphers are not used by verifying that Ciphers aes256-ctr, aes128-ctr are the only listed ciphers.
- Ensure that only message authentication codes are used in the entry MACs hmac-sha1.

Disabling Direct Logins as root

By default, the hardened appliances allow direct login to root via the console. After administrative accounts are created for non-repudiation and tested for wheel access (su-root), direct root logins can be disabled by editing the `/etc/securetty` file as root and replacing the tty1 entry with console.

VMware, Inc. 3401 Hillview Avenue Palo Alto CA 94304 USA Tel 877-486-9273 Fax 650-427-5001 www.vmware.com