Replacing the SSL certificate used by RHEV Manager for HTTPS connections

Author Name: Dan Yasny, Chris Negus
Editors: Allison Pranger
04/10/2012

This tech brief allows you to:

- Replace the existing RHEV-M set of certificates used to secure traffic to the services RHEV-M exposes via HTTPS with a signed certificate from a valid Certificate Authority.
- Prevent users from using a self-signed certificate and certificate authority in accessing the RHEV-M.

OVERVIEW

When you connect to a Red Hat Enterprise Virtualization Manager (RHEV-M) from a Web browser, the RHEV-M downloads a certificate to the browser that is signed by the RHEV-M. That certificate, which is automatically generated by the RHEV-M (the RHEV-M acts as a certificate authority), allows any HTTPS connection between the browser and the RHEV-M to be encrypted.

One drawback is that the first time a user accesses the RHEV-M, a message shows the connection is untrusted. The user must accept the certificate and the RHEV-M as a certificate authority. By replacing certificate information on the RHEV-M with certificates that are signed by a public certificate authority (such as Verisign or Thawte), a user can automatically validate the RHEV-M certificate without seeing that scary "Connection is Untrusted" or similar message.

This procedure describes how to get a certificate that is signed by an accepted Certificate Authority and replace the RHEV-M certificate with the commercially signed certificate. This certificate will apply to any HTTPS connections (Administrative Portal, User Portal, and so on) between the browser and the RHEV-M, with the certificate being validated without the user having to accept anything special.

> **NOTE**: This procedure was tested on a temporary SSL Test certificate from VeriSign. To test the procedure the same way, you could go to http://www.verisign.com/ssl/free-30day-trial/index.html. From that site, VeriSign provides a root CA cert, an intermediate cert, and the cert for your FQDN (emailed to you after CSR is approved). The certificate described here is a test certificate, with a single intermediate certificate and a separate CA. A normal VeriSign certificate chain contains the CA, two intermediate certificates and the private certificate.

PROCEDURE: REPLACING RHEV-M CERTIFICATES

Run the following procedure from a shell as the root user on the RHEV-M system.

1. **Create a new keystore on the RHEV-M**. Open a shell as root and run the **keytool** command to generate a Java keystore that includes information about your organization. Use any values you like for the options you give to **-alias** and **-keystore**, but remember those values, since you will need them later (along with the passwords you set).

```
# keytool -genkey -alias signedcert -keyalg RSA \
          -keystore signed.keystore -validity 3650
Enter keystore password: mypass
Re-enter new password: mypass
```

```
What is your first and last name?
  [Unknown]:  myrhevm.example.com
What is the name of your organizational unit?
  [Unknown]:  Labs
What is the name of your organization?
  [Unknown]:  Example.com Inc.
What is the name of your City or Locality?
  [Unknown]:  Raanana
What is the name of your State or Province?
  [Unknown]:  Centre
What is the two-letter country code for this unit?
  [Unknown]:  IL
Is CN=John Jones, OU=Labs, O=Example.com Inc., L=Raanana, ST=Centre, C=IL correct?
  [no]:  yes
Enter key password for <signedcert>
(RETURN if same as keystore password): mypass
Re-enter new password: mypass
```

A file containing the signed Java keystore (in this case, named **signed.keystore**) is saved to the current directory. Remember the values you used for ***signedcert***, ***signed.keystore***, and ***mypass***.

2. **Creating a Certificate Signing Request (CSR)**. Use the keytool to create a certificate signing request (**.csr**) file that you can send to the certificate authority (CA), such as Verisign (**www.verisign.com**) or Thawte (**http://www.thawte.com**). The certificate authority will create a certificate (**.crt**) from the certificate signing request file (**.csr**) you create.

```
# keytool -certreq -v -alias signedcert -file myrhevm.example.com.csr \
          -keypass mypass -keystore signed.keystore -storepass mypass
Certification request stored in file <myrhevm.example.com.csr>
Submit this to your CA
```

3. **Check CSR file**. To check the contents of the CSR file, type the following:

```
# openssl req -text < myrhevm.example.com.csr | less
Certificate Request:
 Data:
  Version: 0 (0x0)
  Subject: C=IL, ST=Centre, L=Raanana, O=Example Inc., OU=Labs, CN=myrhevm.example.com.csr
```

4. **Submit the CSR to the CA**. To submit the CSR file to Verisign, Thawte, or other Certificate Authority, follow the instructions they provide on their Web sites. You might email the file to the CA or paste the CSR into a Web page. If they ask you to paste the certificate into a Web page, you can use **cat** to display the file, then copy and paste the contents with your mouse.

```
# cat myrhevm.example.com.csr
-----BEGIN NEW CERTIFICATE REQUEST-----
MIIByjCCATMCAQAwgYkxCzAJBgNVBAYTAklMMQ4wDAYDVQQIEwVEYXJvbTEPMA0GA1UEBxMGQXNo
ZG9kMQwwCgYDVQQKEwNnc3MxHzAdBgNVBAsTFmdzcy5sYWIudGx2LnJlZGhhdC5jb20xKjAoBgNV
BAMTIXJoZXZtLWRvdDQuZ3NzLmxhYi50bHYucmVkaGF0LmNvbTCBnzANBgkqhkiG9w0BAQEFAAOB
jQAwgYkCgYEAhEa3BfIvU0f97yadCwMFvOSkZsYna+Da8jV1cQ5Ku/GhT3rI1Q+4aI8hACMsdNex
4pQSzOQBxiMncpl0vf8kb46T7OiyF7MBJnkyYaVDq+XS1q4iO4wQR3vjEooDNMdwow2m2oKd43Q7
0J5xgoz2XznrNiaRVUsHhmFB0EBtK1cCAwEAAaAAMA0GCSqGSIb3DQEBBQUAA4GBADIyTIF0tYv6
z0mkYns3KsMI76oeIMhe+95GusPhOXfbNMF8x9pW5ALTrbppCyLiPsrcaAkWJnmOSsNQflKqXfhg
ERwsQgwLJSx5B8gfZfzF0lAh1RrZhNG2LGdOCLnQawgIsCc/cwH0V9SxZQsnHAJa5hPJ2YFjaaOX
4+Lx+GFe
-----END NEW CERTIFICATE REQUEST-----
```

5. **Add the certificate chain to the keystore**. After you have submitted your CSR to the CA, you should receive the following (or links to the following):
   - The root CA cert (obtainable from VeriSign). Save it to the file name **root.cer**.
   - The intermediate cert (also obtainable from Verisign). Save it to the file name **intermediate.cer**.
   - The private cert for the RHEV-M server (VeriSign emails it after the CSR goes through). Save it to the file name **private.cer**.

   Assuming the **root.cer**, **intermediate.cer**, **private.cer** and **signed.keystore** files are in the current directory on the RHEV-M, run these commands to add information to **signed.keystore**:

```
# keytool -import -trustcacerts -alias root -keystore signed.keystore \
          -file root.cer
Enter keystore password: ********
...
Trust this certificate? [no]: yes
Certificate was added to keystore

# keytool -import -trustcacerts -alias intermediate -keystore signed.keystore \
          -file intermediate.cer
Enter keystore password: ********
...
Trust this certificate? [no]: yes
Certificate was added to keystore

# keytool -import -trustcacerts -alias signedcert -keystore signed.keystore \
          -file private.cer
Enter keystore password: ********
...
Trust this certificate? [no]: yes
Certificate was installed in keystore
```

NOTE: If you see "keytool error: java.lang.Exception: Failed to establish chain from reply" Then you are missing certificates in the signature chain.

6. **Check the keystore**. To check the contents of the **signed.keystore** file, type the following. It should contain fingerprints for the **root**, **intermediate** and **signedcert** certificates entered earlier:

```
# keytool -list -keystore signed.keystore
Enter keystore password:  ********
...
root, Apr 5, 2012, trustedCertEntry,
Certificate fingerprint (MD5): E0:19:F5:FC:C0:9A:53:0E:38:B7:DF:0D:02:40:D3:A1
intermediate, Apr 5, 2012, trustedCertEntry,
Certificate fingerprint (MD5): 71:13:D9:3A:CD:21:F2:EE:9F:59:17:8D:A6:F9:AE:14
signedcert, Apr 5, 2012, PrivateKeyEntry,
Certificate fingerprint (MD5): 6C:69:4D:24:DE:EC:47:30:76:7E:9D:6A:A3:90:60:00
```

7. **Stop the jbossas service**. Type the following to stop the jbossas service:

```
# /etc/init.d/jbossas stop
```

8. **Place the signed.keystore file in the correct location**. Copy the **signed.keystore** file to the **/etc/pki/rhevm** directory and set its ownership and permissions as follows:

```
# cp signed.keystore /etc/pki/rhevm/
# chmod 750 /etc/pki/rhevm/signed.keystore
# chown jboss:jboss /etc/pki/rhevm/signed.keystore
```

9. **Edit JBoss-Web configuration for the RHEV-M**. Open the **server.xml** in the /usr/share/jbossas/server/rhevm-slimmed/deploy/jbossweb.sar/ directory and change a line to contain the new keystoreFile, keystorePass and keyAlias entries:

```
# cd /usr/share/jbossas/server/rhevm-slimmed/deploy/jbossweb.sar/
# vim server.xml
```

The line you edit should appear near the end of the file. The values you change are highlighted:

```
<Connector protocol="HTTP/1.1" SSLEnabled="true" port="8443" address="$
{jboss.bind.address}" scheme="https" secure="true" clientAuth="false"
keystoreFile="/etc/pki/rhevm/signed.keystore" keystorePass="mypass"
keyAlias="signedcert" sslProtocol="TLS"/>
```

10. **Start the jbossas service**. Run the following command to start the jbossas service again.

```
# service jbossas start
```

11. **(Optional) Replace the ca.crt file**. If the root CA that signed the certificate is not globally known (like VeriSign), the root CA Certificate (in this case **root.cer**) should also be renamed as **ca.crt** and placed under **/var/lib/jbossas/server/rhevm-slimmed/deploy/ROOT.war/ca.crt**. Otherwise, a person accessing the Admin GUI through SSL it will not show the correct certificate. With the **root.cer** file in the current directory, here is what you do:

```
# cp root.cer /var/lib/jbossas/server/rhevm-slimmed/deploy/ROOT.war/ca.crt
# chmod 750  /var/lib/jbossas/server/rhevm-slimmed/deploy/ROOT.war/ca.crt
# chown jboss:jboss  /var/lib/jbossas/server/rhevm-slimmed/deploy/ROOT.war/ca.crt
```

NOTE: If you have a signed intermediate organizational certificate instead of an actual CA certificate from an authority, you need to specify the location of that certificate in the **/etc/pki/rhevm/cacert.template** file. In that file on the RHEV-M, change the **authorityInfoAccess** entry. For example, if your certificate were on certs.example.com, you could change the line to read:

authorityInfoAccess = caIssuers;URI:http://certs.example.com/ca.crt

12. **Test the new certificate**. Open an Internet Explorer Web browser and attempt to access the RHEV-M. For the example here, the URL would be **https://myrhevm.example.com:8443**. Try to access the Administrator Portal to make sure the certificates allow you to access that portal.