V3 Appliance SSL Certificate

Contents

Adding a SSL Certificate

If you have an existing keystore on an older Swivel appliance that you wish to repurpose on a newer Swivel appliance, then it is possible to import it using the menu in the new Swivel appliance. See the section Import existing Swivel keystore file.

If you don?t have an existing keystore to import, you can generate a new certificate to be signed by a Certificate Authority. See the section Generate new local certificate.

If you just want a self-signed certificate for testing purposes then you can either use the default self-signed certificate that ships with the appliance by default, or generate a new self-signed certificate with a custom Common Name (CN) / sitename e.g. acme.customersite.com. See the section Generate a self-signed certificate.

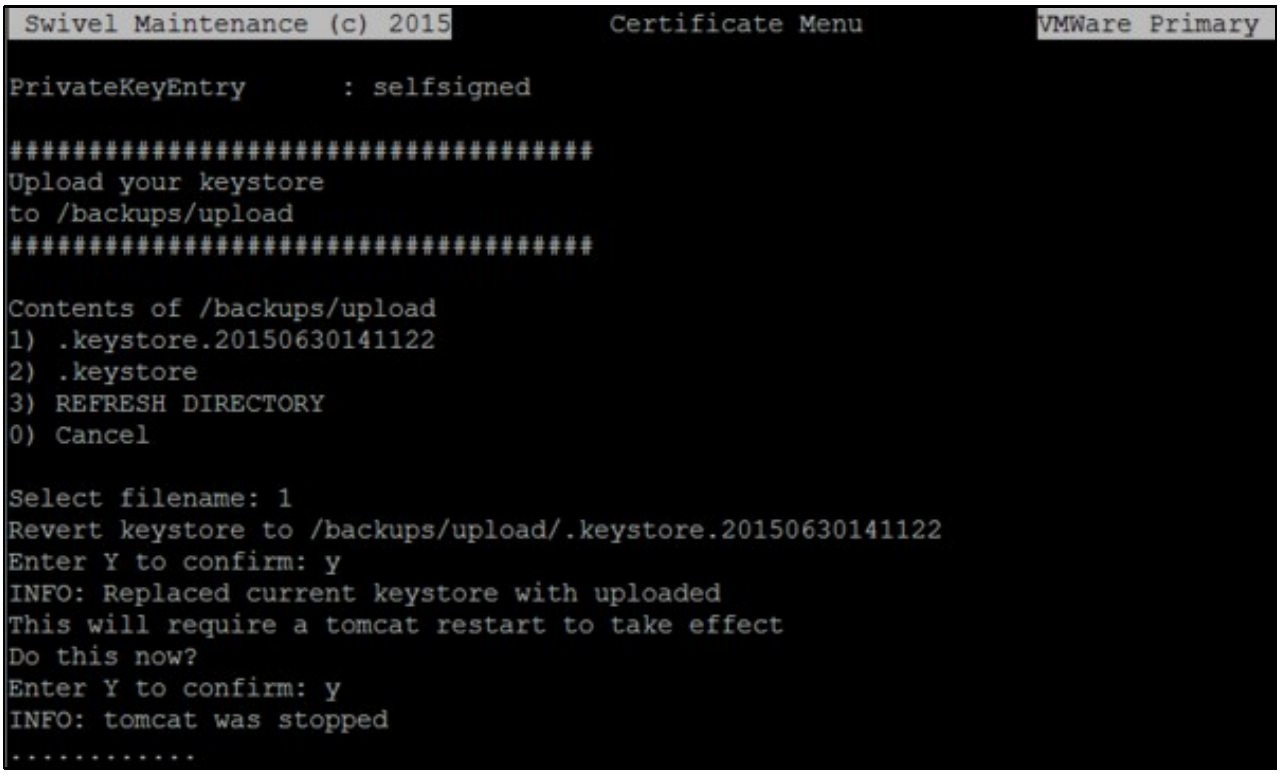
Prerequisites

- Public DNS record for the Swivel instance, usually resolving to a Public IP address;
- Certificate Authority to sign a Certificate Signing Request (CSR);
- Full appliance backup or copy of the /home/swivel/.keystore file, in case things don?t go to plan;

Import existing Swivel keystore file

A backup of the original keystore will be taken as part of the process, named with the date that you tried to replace the keystore. This can be restored using the ?Roll Back? option in the menu, provided that not too much time has passed.

Using WinSCP (see WinSCP How To Guide) copy the keystore you wish to import to /backups/upload. From the Certificate Menu, select ?Import / Roll Back to Previous Keystore?, then ?Import Keystore?



Select the keystore you wish to restore to and restart tomcat as prompted. The keystore will be renamed and given the appropriate permissions.

Generate new local certificate

If you don?t have an existing keystore to import, as per the above instructions, you can generate a new certificate.

In summary, the process is as follows:

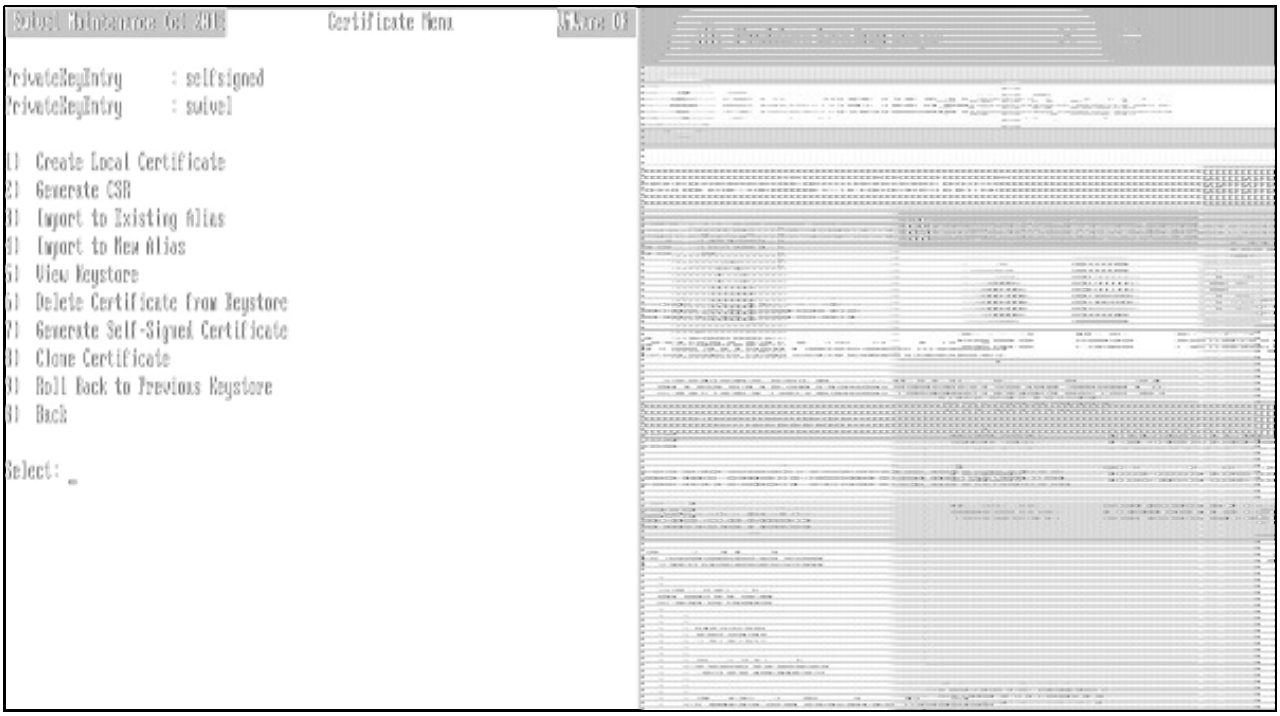
- Create a new local certificate on the appliance;
- Generate a Certificate Signing Request (CSR) for the new local certificate;
- Submit the CSR to your chosen Certificate Authority (CA) online, for signing;
- The CA will provide a response which contains the signed certificate, possibly some Intermediate *Certificates and a Root certificate;
- Import the CA Root Certificate to the Keystore;
- Import any CA Intermediate Certificates (there may be multiple Intermediates and they go by various names depending on the CA e.g. Primary and Secondary Intermediates);

The process is now described in detail:

Create Local certificate

From the Certificate Management menu, select the **Create Local Certificate** option.

A new alias of ?swivel? will appear in the certificate list as a ?PrivateKeyEntry?:



At this point, you can delete the ?PrivateKeyEntry? named ?selfsigned? that shipped with the appliance (using the **Delete Certificate from Keystore** option), otherwise this will conflict with the new alias that you have created.

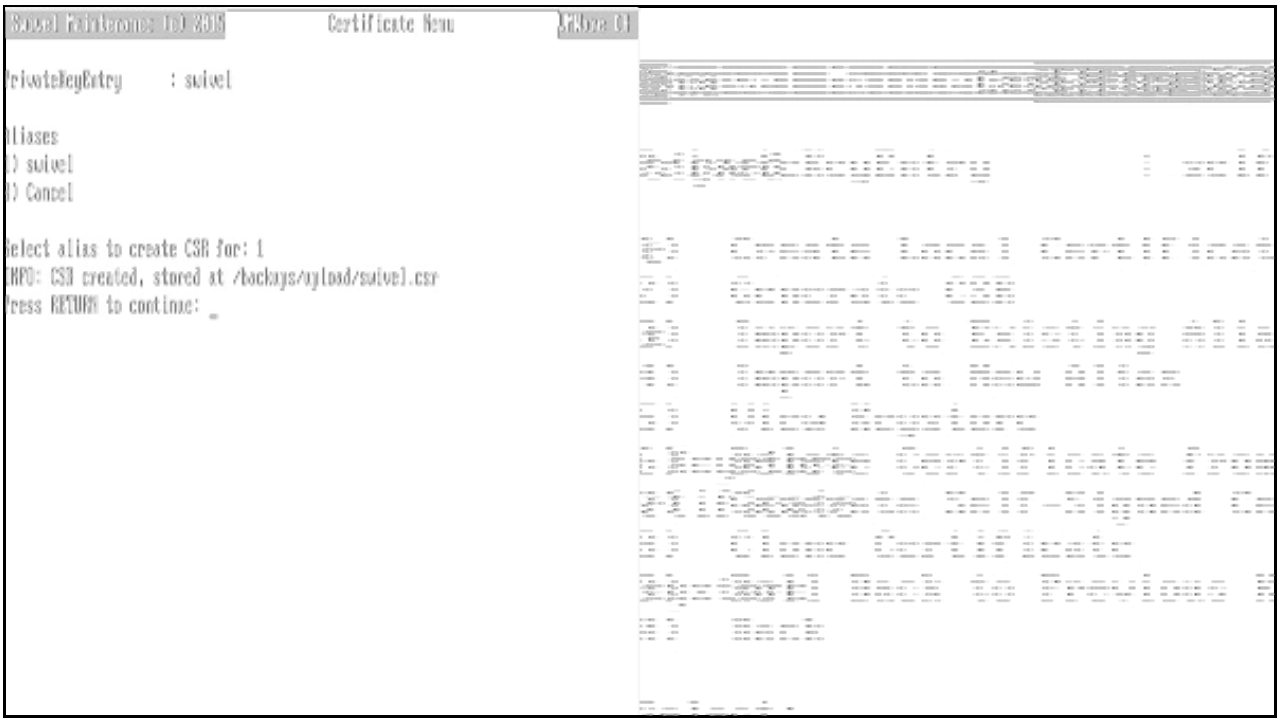
Generate Certificate Signing Request

Before generating a Certificate Signing Request (CSR), you must create a local certificate as detailed in the previous section.

To generate a CSR from the new local certificate, select the **Generate CSR** menu option from the Certificate Management menu screen.

Note: If you haven?t yet deleted the ?selfsigned? alias you will be prompted to delete it now.

Select the ?swivel? alias from the list:



The CSR will be created as a text file named swivel.csr. You can retrieve the file from the appliance using WinSCP (see WinSCP How To Guide). As the screenshot suggests, the location of the file is:

/backups/upload/swivel.csr

Submit the CSR to your chosen Certificate Authority (CA)

The contents of the ?.csr? file can be submitted to your chosen CA. Assuming that this is a commercial Certificate Authority, then usually they will respond within 24 hours.

You will usually have to paste the contents of the CSR file into the CA?s web page. Notepad or a similar text editor e.g. Notepad++ is most appropriate for copying the text.

The response, typically by email after purchase through a website, may include:

- Download links to the generic CA Root Certificates;
- Download links to the generic CA Intermediate Certificates;
- You may be provided with a ?certificate bundle? containing the CA Root and CA Intermediates bundled into one certificate - we would recommend avoiding this and instead importing the certificates as separate aliases, since bundles have not always been reliable;
- The unique, signed response of the certificate you generated, which you will import on top of the ?swivel? alias at the end of the import process;

Note: You may not receive Download Links by email or indeed any information pertaining to the Root and Intermediate certificates, from the CA. If this is the case, you will need to visit the CA website to obtain them prior to importing the signed response into the Keystore. This is because if they aren?t imported prior, you will receive an error ?Could not establish chain from reply?. Your signed response is essentially useless without them.

The Download Links provided by the Certificate Authority to the Root and Intermediate(s), may provide links to alternative Root and Intermediate certificates which do not apply to the SSL Certificate product you purchased. Hence you need to be careful to download only those Root and/or Intermediates that apply to the product you purchased in order for the import to be successful.

Import the Certificates

The certificates need to be imported in a specific order:

1. Import the Root certificate, with a unique alias name, e.g. ?root?. By importing this first, all the subsequent imports will be able to establish a chain from it. You may find that you are told that this is already imported into the ?system-wide keystore? which is separate to the keystore you are working with. That?s OK, but it?s probably wise to import it into this keystore anyway if prompted - so that the keystore and entire certificate chain is self contained should you migrate systems later;
2. Import the Intermediate certificate(s), with unique alias names e.g. ?intermediate1? and ?intermediate2?. There may only be one Intermediate depending on your Certificate Authority and how they operate. Once an Intermediate is imported, then your unique signed certificate will be able to establish a chain from it (and the Root certificate imported earlier);
3. Once the above certificates are in place, you can import the signed certificate onto the ?swivel? alias. The ?swivel? alias is the default alias name when you create a new local certificate. You should receive a success message at this point such as ?Certificate was added to keystore?. Alternatively you may receive a failure message such as ?Failed to establish chain from reply?. In this situation it is wise to check that you have imported the correct Root and Intermediate certificates prior, that are relevant to the SSL/TLS product you purchased. If you are not sure, contact your CA support who will be able to confirm.

Now you have imported the Certificates, see the section Apply the changes - Restart Tomcat to make the changes take effect.

Generate a self-signed certificate

If you need to create a self signed certificate, do not wish to use a commercial certificate signed by a Certificate Authority and have no need for the default self-signed certificate that ships with the appliance, then you can generate your own.

Prerequisites

- Prior to doing this you will need to delete the existing self-signed certificate.

See the section Delete the existing self-signed certificate.

- It?s advisable to ensure that you have taken a Backup first.

See the section Taking a Backup.

Delete the existing self-signed certificate

From the Certificate Menu, select the ?Delete Certificate from Keystore? option. Select the certificate aliases you want to delete until there are no more entries left.

Generate Self-Signed Certificate

From the Certificate Menu, select the ?Generate Self-Signed Certificate? option. You will be prompted to enter various attributes to create the certificate:

```
6)  Delete Certificate from Keystore
7)  Generate Self-Signed Certificate
8)  Clone Certificate
9)  Roll Back to Previous Keystore
0)  Back

Select: 7

Example:

Domain Name     : pinsafe.swivelsecure.com
Company Name    : Swivel Secure Ltd
Department      : IT Department
City            : Wetherby
County          : West Yorkshire
Country Code    : GB

Enter domain, or press RETURN to use default (Unknown): test.domain.com
Enter Company Name, or press RETURN to use default (Unknown): Example Company
Enter Department, or press RETURN to use default (Unknown): IT Department
Enter City, or press RETURN to use default (Unknown): London
Enter County, or press RETURN to use default (Unknown): Westminster
Enter Country, or press RETURN to use default (Unknown): GB
INFO: Certificate created, alias selfsigned
Press RETURN to continue: _
```

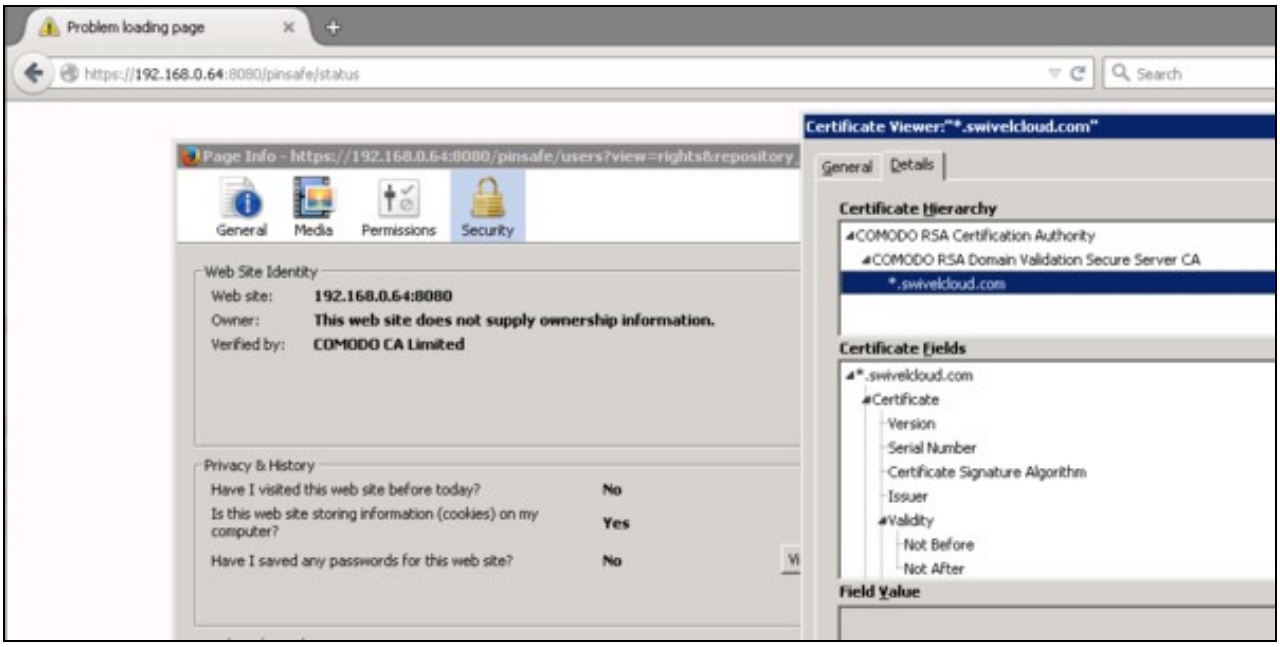
Once created, the certificate will appear as a PrivateKeyEntry alias named ?selfsigned?:

```
Swivel Maintenance (c) 2015          Certificate Menu          VMWare DR

PrivateKeyEntry      : selfsigned

1)  Create Local Certificate
2)  Generate CSR
3)  Import to Existing Alias
4)  Import to New Alias
5)  View Keystore
6)  Delete Certificate from Keystore
7)  Generate Self-Signed Certificate
8)  Clone Certificate
9)  Roll Back to Previous Keystore
0)  Back

Select: _
```

Now you have generated the Self-Signed Certificate, see the section Apply the changes - Restart Tomcat to make the changes take effect.

Apply the changes - Restart Tomcat

Once you restart the Tomcat service, the new keystore contents will be loaded. After the restart you can check the Certificate using a web browser. When the Swivel Core is operating in HTTPS mode, you can inspect the certificate padlock icon in the web browser address bar to reveal more information about the sitename or Common Name (CN). You should also be able to see the certificate chain/hierarchy when you view the certificate, containing all the certificates you imported (see the Certificate Hierarchy in the picture below).



Note: When you view the certificate in the browser, it?s wise to enter the actual Fully Qualified Domain Name into the browser address bar, matching the site name of the certificate.

As an example, assuming that the Public DNS record is active and resolving, and the sitename/Common Name (CN) of the certificate was core.swivelsecure.com then you would visit:

https://core.swivelsecure.com:8080/pinsafe

If you don?t use the sitename as above and instead use the local or public IP address as shown in the picture above, then this will cause the web browser to report that the certificate is ?invalid?. This is because the sitename of the certificate will not match the IP address in the address bar. Only until the hostname in the address bar matches the Common Name (CN) of the certificate, will you resolve the ?invalid? certificate issue.

If you still get an invalid certificate after eliminating the FQDN/CN mismatch issue above, then it is likely that you have not successfully established the certificate chain.

When you view the keystore you need to be sure that the alias for your unique certificate is definitely a PrivateKeyEntry and not TrustedCertEntry. If it is a TrustedCertEntry then it?s likely that you?ve somehow deleted the original Local Certificate that you generated, containing the private key - and have just imported the response from the Certificate Authority into a new alias. It is important that the response from the CA for your unique certificate it imported back on top of the PrivateKeyEntry in order to sign the Private Key. Root and Intermediate certificates can be a TrustedCertEntry without issue.

Troubleshooting

If you have any problems after the Import and Restart then see the Troubleshooting section of the SSL Guide on the Knowledgebase. A good place to start if Tomcat will not run, or the Swivel Core is inaccessible is to review the Catalina.out log file (/var/log/tomcat/catalina.out). Look towards the bottom of this file to see the latest errors since Tomcat was restarted. A common problem can be permissions on the .keystore file itself, especially when copied from another appliance.