SSL Certificate PINsafe Appliance How to Guide

Contents

Overview

Swivel virtual or hardware appliances ship with a self-signed certificate which prompts users to accept a security warning when the Turing image is presented within a web browser. When a Swivel system goes into production a signed certificate should be installed onto the virtual or hardware appliance. This article describes how to install a valid certificate using the Swivel virtual or hardware appliance CMI menu.

A certificate request can be made from the Swivel virtual or hardware appliance or an existing certificate can be used by importing the private AND the public key.

If you do not have an virtual or hardware appliance, but instead a software only installation of Swivel, please see our Generate CSR for Tomcat How to Guide.

Also see the SSL Solutions guide.

Applying for certificates may take some time so we advise that renewals are carried out in good time before current certificates expire.

Prerequisites

- Swivel virtual or hardware appliance version 2.x with Console Management Interface (CMI)
- DNS name for the Swivel instance, usually the public IP address
- Configuration of the Swivel virtual or hardware appliance for basic settings
- Certificate Authority to sign a certificate signing request
- Please read and understand these instructions before attempting to install a certificate
- Ensure backups of the keystore exist

Considerations

HA virtual or hardware Appliances using VIP

For HA virtual or hardware appliances, if a VIP (Virtual IP) is being is used then the certificate must to be bound to a hostname that is used on both Swivel servers. When you've setup a signed-certificate on one virtual or hardware appliance then the keystore can be copied to the other virtual or hardware appliance.

NAT address and Certificates

Where the Swivel virtual or hardware appliance or HA VIP is behind a NAT, the DNS entry used as for the IP address of the NAT is usually used as the hostname for the certificate and with a Swivel HA VIP, that certificate is imported into the secondary by transferring the /home/swivel/.keystore file to the standby see below for details on this.

virtual or hardware Appliance Webmin

The other virtual or hardware appliance web based interface, Webmin, uses a separate certificate for SSL communications. Since this is rarely used and then only for administrative purposes by trained administrators, the built-in self-signed certificate is utilised.

Keystore and Connector Ports

The keystore used by each port is defined as a Connector element within the Tomcat server.xml file. You can view this file from Webmin (https://<Swivel_server>:10000). Go to Servers -> Swivel and select "Edit Tomcat config file". There should be 3 "<Connector ..." entries, one each for ports 8080, 8443 and 8181 (the last is for internal use only). The entries for port="8080" and "8443" should both have the same value for keystoreFile, which should be "/home/swivel/.keystore". If one of these is different, it needs to be changed.

Key format

Keystores are by default JKS on the Swivel virtual or hardware appliance. Keys are by default generated in DER and expecting X.509 formatted responses. However you can import PKCS12 keystores as long as you can obtain the password from the certificate authority to be able to convert/import it into a JKS.

VeriSign Certificates

VeriSign certificates generally require to import the Primary and Secondary intermediates (instead of root and intermediate) and remove the existing root and intermediate before importing the response onto the private key

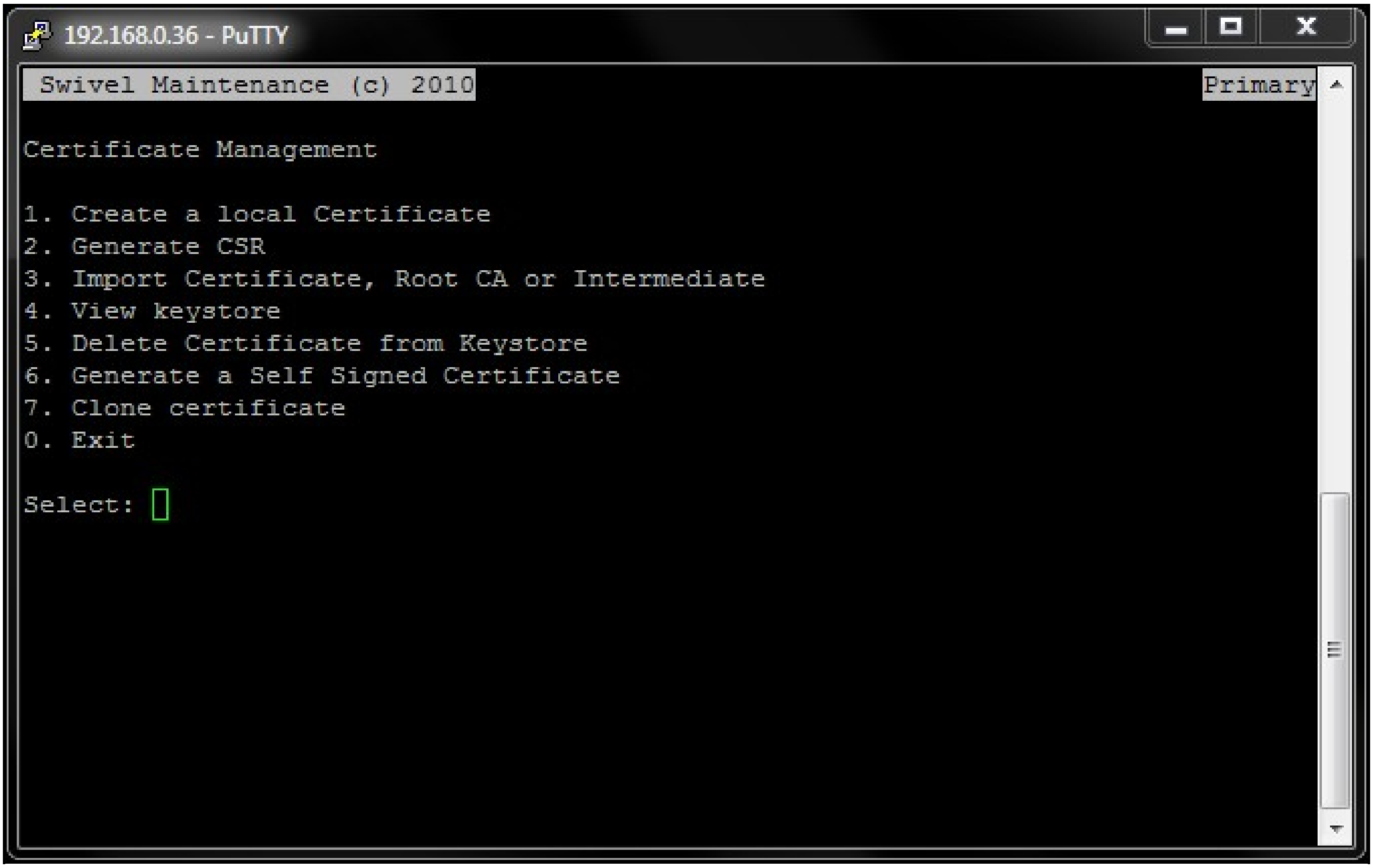
Internal Certificate Authority

See SSL Internal Certificate Authority

Certificate Management CMI menu option

You can find the Certificate Management menu on the Swivel virtual or hardware appliance, by initiating an SSH connection. For information on how to do this, see our PuTTY How To Guide.

Once you've logged into the CMI, you can find the Certificate Management menu under the Tomcat menu and the screen should look something like this (screen shot taken from a Swivel Primary HA appliance):



```
 Swivel Maintenance (c) 2010                                      Primary

Certificate Management

1. Create a local Certificate
2. Generate CSR
3. Import Certificate, Root CA or Intermediate
4. View keystore
5. Delete Certificate from Keystore
6. Generate a Self Signed Certificate
7. Clone certificate
0. Exit

Select:
```

Procedure Summary

The normal procedure in getting a commercially signed certificate for a Swivel virtual or hardware appliance is to:

- Create a self-signed private/public key pair;
- Generate a CSR (Certificate Signing Request) from the self-signed certificate you created, usually alias swivel;
- Import the root or intermediate certificates from the CA (Certificate Authority), usually it is imported with the alias swivel;
- Import the response from the CA to replace the public key.

Note: Do not import the certificates as a bundle, but rather each certificate (root, intermediate, response) needs to be imported individually.

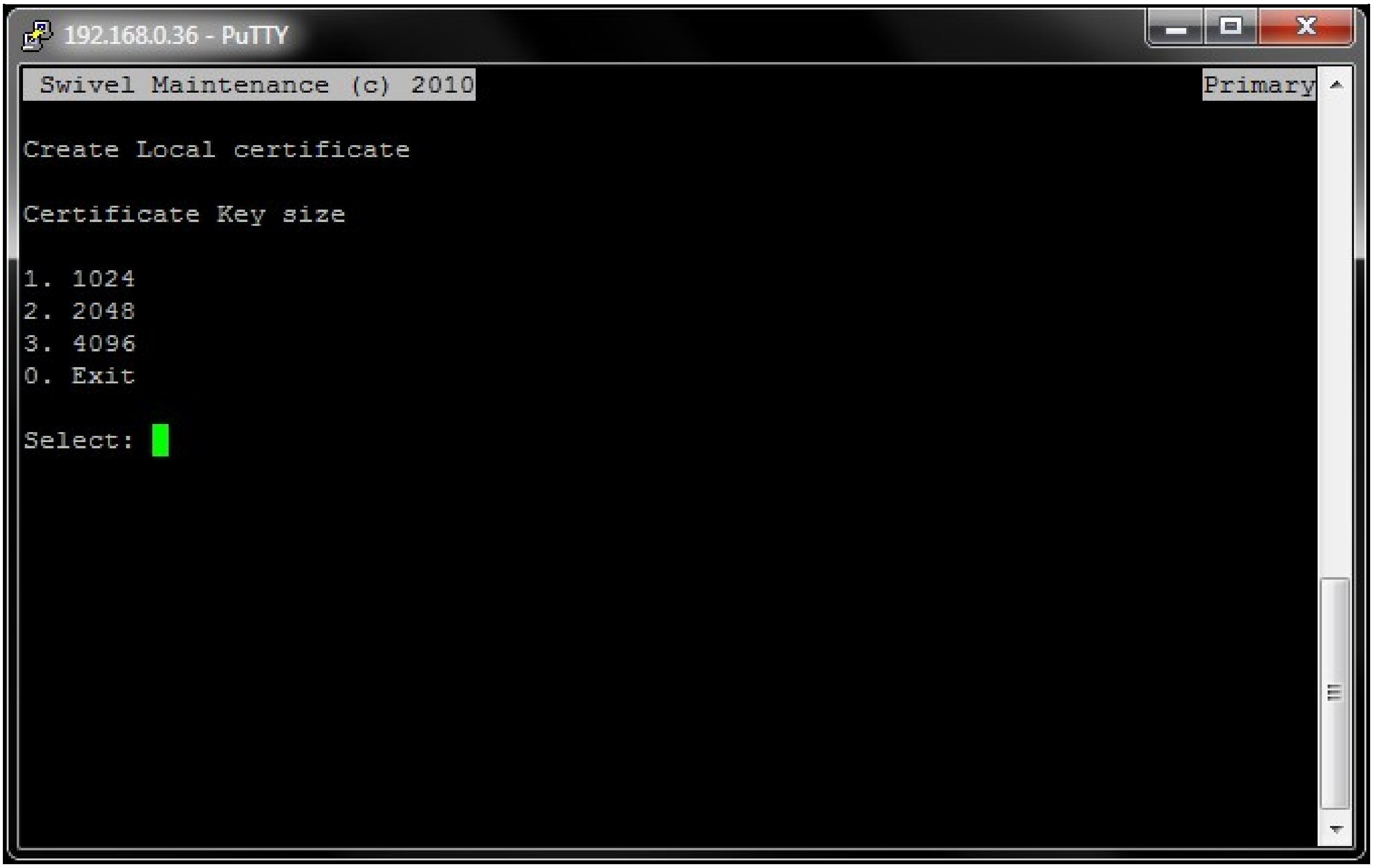
The following article sections detail how this is done via the Certificate Management CMI menu option.

1. Create a local certificate

From the Certificate Management menu, select the Create a local Certificate option.

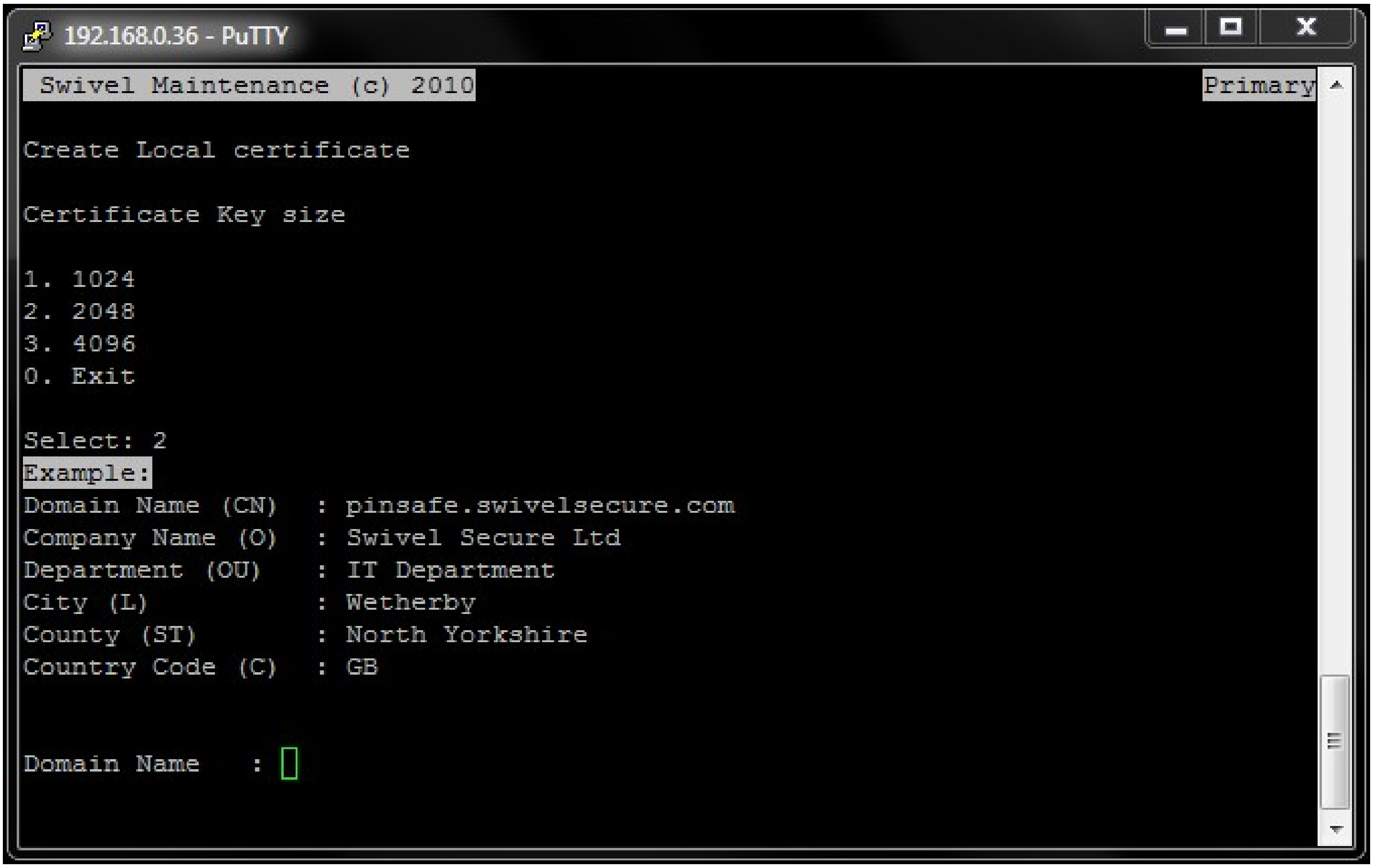
Note: With a certificate renewal you just need to Generate a CSR on the existing cert, without creating a new cert. You will need to enter the alias of the existing keypair when selecting the cert to generate a csr for. **Usually this is "swivel"**.

Here is a screen shot of the first screen you encounter when selecting the Create a local Certificate option. At the time of writing this article, most CAs (Certificate Authorities) require at least 2048 bit key size.



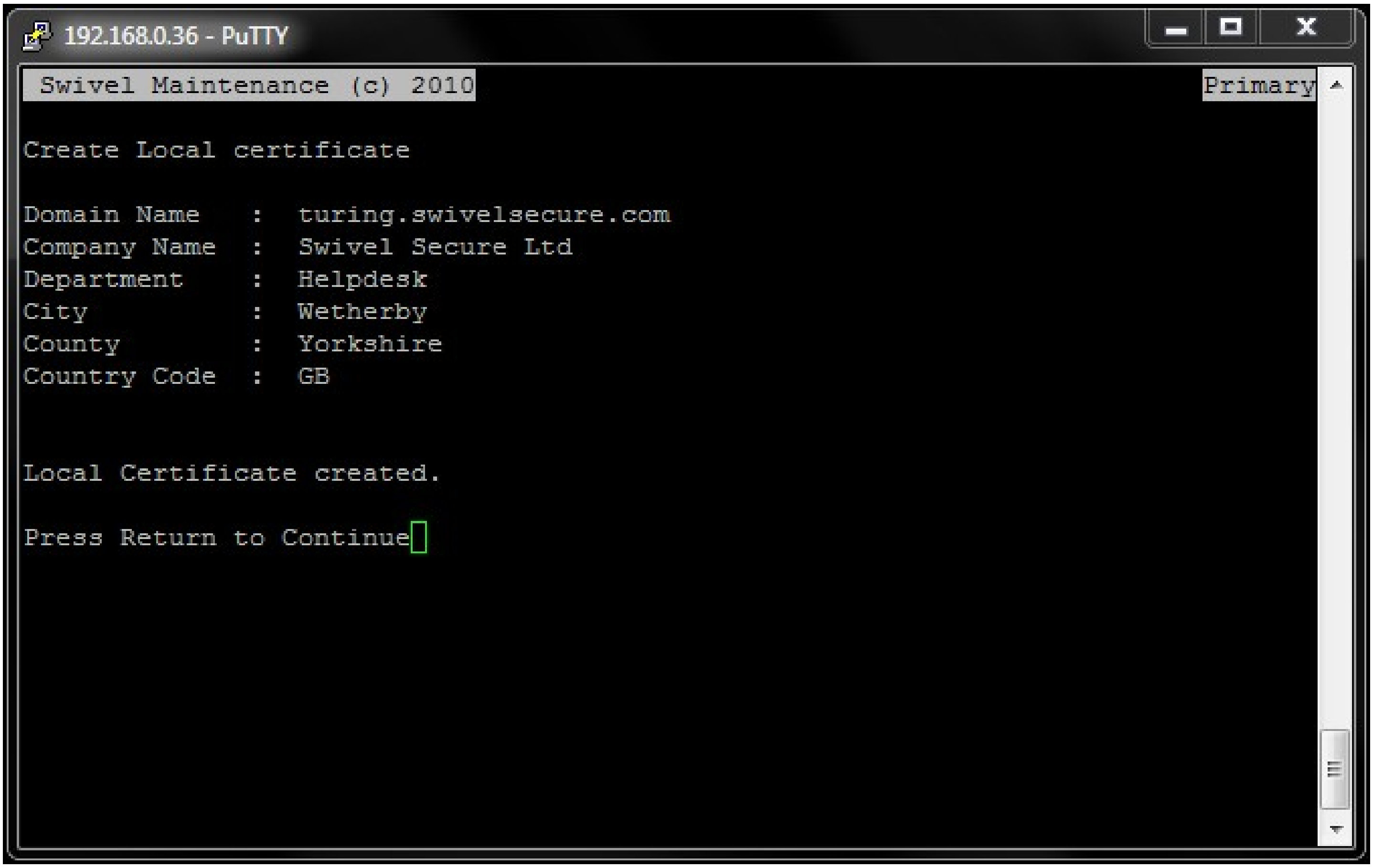
```
 Swivel Maintenance (c) 2010                                        Primary

Create Local certificate

Certificate Key size

1. 1024
2. 2048
3. 4096
0. Exit

Select:
```

You are next prompted to provide information on the site-name (URL of the Swivel Turing image that the certificate will be securing), company name and location information.



```
 Swivel Maintenance (c) 2010                                          Primary

Create Local certificate

Certificate Key size

1. 1024
2. 2048
3. 4096
0. Exit

Select: 2
Example:
Domain Name (CN)   : pinsafe.swivelsecure.com
Company Name (O)   : Swivel Secure Ltd
Department (OU)    : IT Department
City (L)           : Wetherby
County (ST)        : North Yorkshire
Country Code (C)   : GB


Domain Name    : 
```

Once all of the information has been entered (including correct country code) you will be presented with the information for review and confirmation that the certificate has been created.



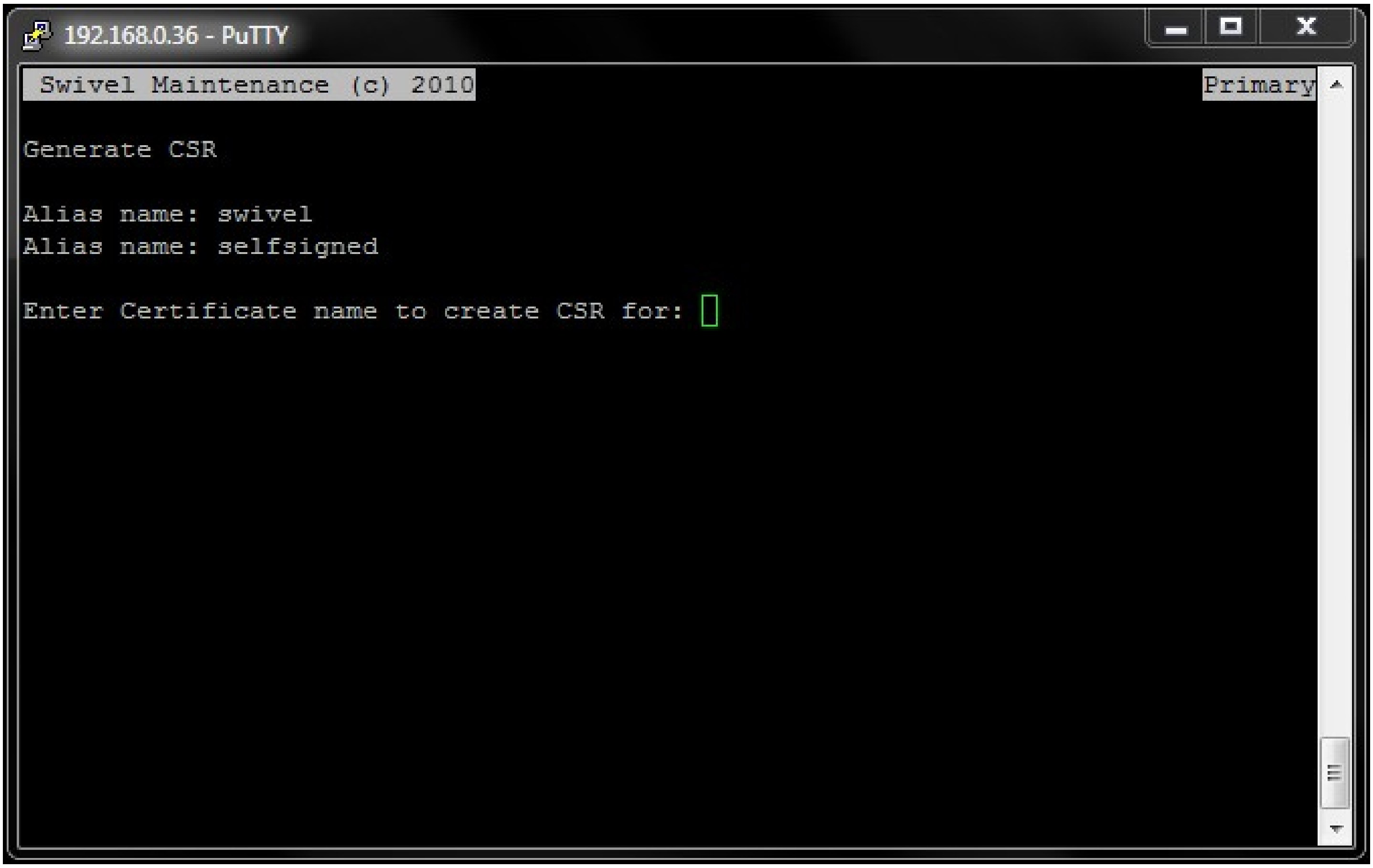
```
 Swivel Maintenance (c) 2010                                          Primary

Create Local certificate

Domain Name    :  turing.swivelsecure.com
Company Name   :  Swivel Secure Ltd
Department     :  Helpdesk
City           :  Wetherby
County         :  Yorkshire
Country Code   :  GB


Local Certificate created.

Press Return to Continue
```

Now you have created a local Certificate, you can generate a CSR (Certificate Signing Request) from it.

2. Generate a CSR

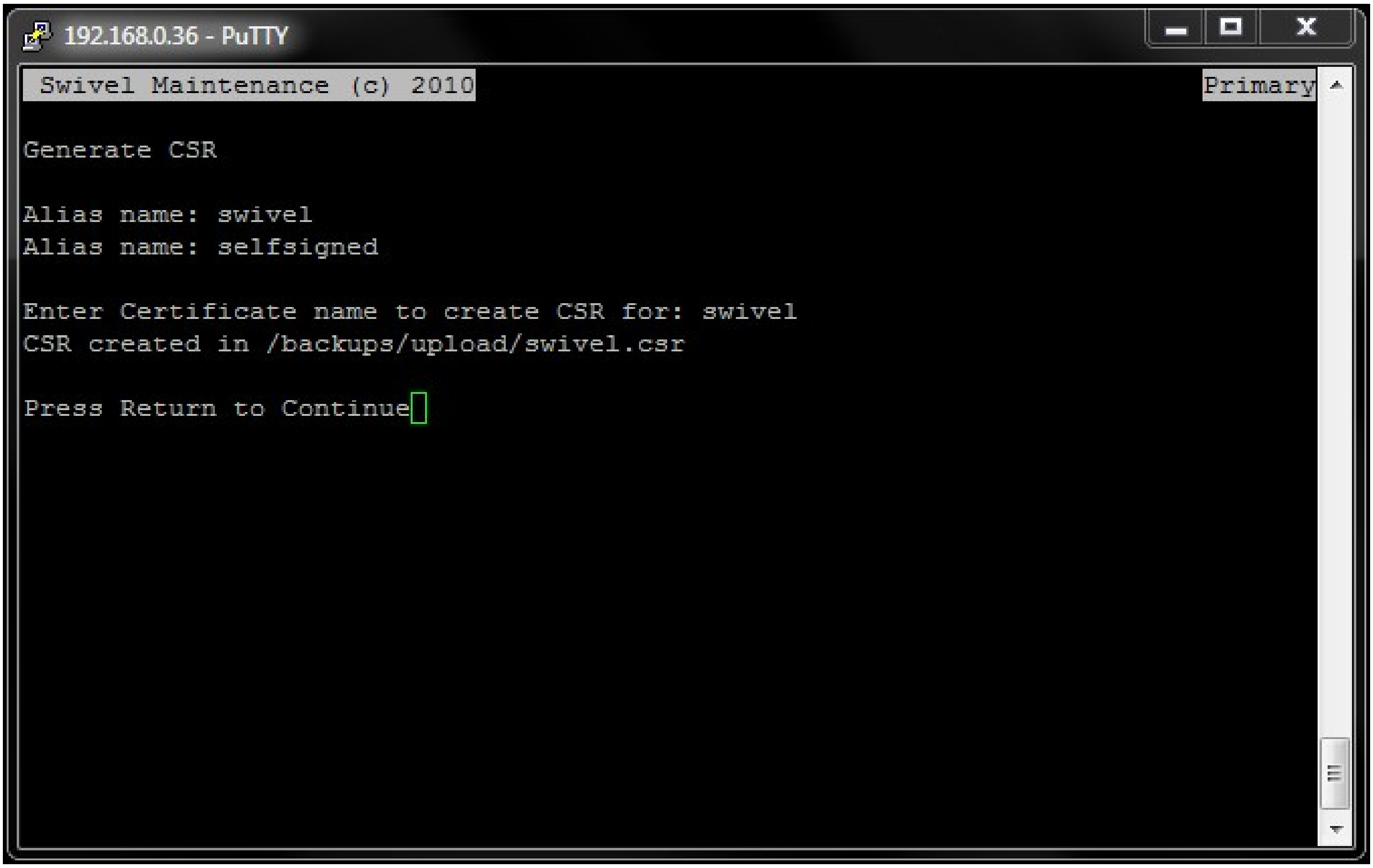
Before performing this step, please ensure you have successfully created a local certificate, as detailed in the previous section.

To generate a CSR (Certificate Signing Request) from the new local certificate, select the Generate CSR menu option from the Certificate Management menu screen. You will be presented with the following screen.



```
 Swivel Maintenance (c) 2010                                        Primary

Generate CSR

Alias name: swivel
Alias name: selfsigned

Enter Certificate name to create CSR for:
```

You need to enter the alias swivel as shown in the next screen shot, as this is the alias used by the Certificate Management software when you create a local certificate.



```
 Swivel Maintenance (c) 2010                                              Primary

Generate CSR

Alias name: swivel
Alias name: selfsigned

Enter Certificate name to create CSR for: swivel
CSR created in /backups/upload/swivel.csr

Press Return to Continue
```

You now need to copy the certificate signing request file you just generated, from the virtual or hardware appliance, so that you may send it to your Certificate Signing Authority. For more information on how to copy files to and from the Swivel virtual or hardware appliance, see the Copying appliance files How to Guide.

The CSR is created under /backups/upload as <certificate alias>.csr where <certificate alias> in this case would be swivel, e.g. swivel.csr

Note: ensure that a backup is made of the .keystore file as the CSR needs the keystore from which the request was made. The CMI does this automatically, but if any command line work is carried out such as using keytool, then the keystore is not backed up.

3. Apply for the Certificate

Request the certificate from the certificate provider via their website. You should receive an email response once the certificate has been signed. You need to **look for "Tomcat" or "Java" keystore compatible formats**. This is fairly crucial, otherwise you will create further complications translating the certificate to the correct format and this may require command line work.

4. Import the Certificate

Copy Certificates to the appliance

Copy the intermediate certificates and certificates you require, to the virtual or hardware appliance, under /backups/upload. See Copying appliance files How to Guide.

Import Intermediate certificates

Next, on the Certificate Management menu of the virtual or hardware appliance, first import the intermediate certificates. Ensure they are given aliases that do not already exist in the keystore, any unique name can be used.

Import the certificate

After the intermediates, import the new certificate, making sure you use the alias that was used to generate the CSR, i.e. usually **"swivel"**. It is important that you import the intermediate certificates BEFORE the response, or the response will not be imported correctly.

Note: **If you get an error at this point, such as a warning ; "Please delete the existing certificate, before generating a new one", DO NOT delete the old alias**. If the alias has been deleted see *trustedCertEntry* in Troubleshooting below. This will likely be caused by a known bug in some CMI versions, check the CMI certificate software version. The workaround to this bug is to login the the command line via the Advanced Options menu and enter the following command:

```
keytool -importcert -keystore /home/swivel/.keystore -alias swivel -file /backups/upload/response.txt -trustcacerts
```

If you have to specify the path to keytool then:

```
 /usr/java/jre1.6.0_18/bin/keytool -importcert -keystore /home/swivel/.keystore -alias swivel -file /backups/upload/response.txt -trustcacert
```

Where:

- "swivel" is the alias you were attempting to import the response onto, but failed due to the known bug - note that this is the default alias for a local certificate;
- "/backups/upload/response.txt" is the location of the response file you are trying to import. Replace this with the actual filename of your response file.

5. Check that the certificate is valid

Check the certificate using the View keystore option by selecting the view certificates option. The length of the certificate chain should equal the number of certificates you have installed (including intermediates) plus one (for the root certificate). Also, the certificate type should be "privateKeyEntry", not "trustedCertEntry".

Example:

Alias Name: swivel

Creation date: 01-Feb-2013

Entry type PrivateKeyEntry

Certificate chain length: 4

Intermediate certificates will be listed as "trustedCertEntry"

Example:

Alias Name: intermediatecert

Creation date: 01-Feb-2013

Entry type trustedCertEntry

6. Delete the old selfsigned or old certificate alias if it exists

Ensure a backup is made, for Swivel virtual or hardware appliances /home/swivel/.keystore.

If there is still an alias "selfsigned" in the keystore, delete it, or Swivel may use that instead of the new certificate. Use the delete certificate option for this. Do not delete the local certificate created in step 1.

Where a new alias has been created the previous certificate will need to be deleted, this is important, as if the previous alias relates to an expired or invalid certificate this may be loaded first rather than the correct certificate.

7. Restart Tomcat

Check the currently loaded certificate through the administration console or by a web request (see below), then restart Tomcat to register the new certificate, and check again to ensure that. View the certificate information to ensure it has the correct expiry date.

https://hostname:8080/pinsafe/SCImage?username=test

https://hostname:8443/proxy/SCImage?username=test

Exporting certificate from Primary to Standby Server

This is useful where a VIP is used, and the certificate can be bound to the hostname. The procedure actually involves copying the keystore file to the Standby virtual or hardware appliance, not exporting the keypair.

The keystore file location should be listed within your /usr/local/tomcat/conf/server.xml file, but the default keystore file location is:

/home/swivel/.keystore

(it's a hidden file with the '.' prefix)

The method would involve:

- Take a backup of the existing /home/swivel/.keystore file on the Primary

- Make a note of the permissions assigned to the file, by default they are swivel:swivel 600

- Copy in the Primary .keystore file to the same location on the Standby. See Copying appliance files How to Guide

- Run the following commands to ensure the permissions are set to their

defaults:

chmod 600 /home/swivel/.keystore

chown swivel:swivel /home/swivel/.keystore

- Restart Tomcat

Moving from one virtual or hardware appliance to another

You can't import the response on an virtual or hardware appliance that doesn't have the private key corresponding to the original request. However, you can simply copy the entire keystore from the working appliance to the other one, see the instructions above for Exporting certificate from Primary to Standby Server.

Importing certificates generated on other virtual or hardware appliances and servers

It is possible to import a certificate that was generated from another server, but only if the private key is imported as well. This requires the certificate to be exported from the other server complete with the private key, normally as a .pfx file. This is, however, a much more complicated procedure. See SSL Solutions.

Certificates using a CSR from a different appliance or servers

If the CSR was generated from a different virtual or hardware appliance or server, and not from the Swivel virtual or hardware appliance, then things get complicated. The simplest solution is to request a new certificate from the certificate authority, using a CSR generated as above, it is possible to export the private key from the server that you generated the original request from, but this is more complicated.

Known Issues

* in alias name

The CMI cannot handle "*" characters in the alias name. However, the alias does not have to match the subject of the certificate, so the workaround is to use a different name for the alias without a *.

Keysize

The CMI Certs version 0.7 and earlier does not contain keysize options. This is available in 0.8 onwards. Upgrade the CMI or issue the keysize through the command line.

keytool -genkey -keysize 2048 -alias <YourAlias> -keyalg RSA -keystore /home/swivel/.keystore

Where <YourAlias> is whatever you would like the alias to be called e.g. swivel, it must be an alias that is not already used.

When prompted for the First and Last Name, use the FQDN.

Once the alias has been created a CSR can be made through the CMI or using:

```
keytool -certreq -keystore <keystoreFile> -alias <certificateAlias> -file <CSRFileName>
```

Replace the names in brackets with the appropriate values (<CSRFileName> is the name of the output file for the CSR request).

You can now send off the generated CSR to your certificate authority. Make sure that you request a Tomcat or Java-compatible format.

Note: ensure that a backup is made of the .keystore file as the CSR needs the keystore from which the request was made.

Troubleshooting

Check the Tomcat logs, particularly the Catalina.out file, see Troubleshooting Files FAQ.

Is the single Channel image coming through on other Web Browsers (IE, Firefox, Chrome).

New Certificate not working

Has Tomcat been restarted after installing the certificate? The certificate file is only read when Tomcat starts.

If Importing Server Certificates from Another Machine, ensure that the private key as well as the public key is imported

Verify the certificates on your virtual or hardware appliance from the appliance console. Log on via SSH or on the console. Go to the Tomcat option, then Certificate Management, then View Keystore. This will show a list of installed certificates. Select the required certificate, and check the details. In particular, check the Entry Type. If it is "trustedCertEntry", then only the public key has been imported. If there is an alias of "swivel" or "selfsigned" that is not one that you have installed, this may need to be deleted. If the type of this certificate is PrivateKeyEntry, it may be used in preference to one imported, in this case, you should delete all PrivateKeyEntry certificates except the one you imported.

A self signed certificate may still exist, if there are two certs in the keystore, the first certificate that is found will be presented. Ensure a backup is made, for Swivel virtual or hardware appliances /home/swivel/.keystore.

Verify the permissions on the keystore

Was the Intermediate certificate imported first

Virtual or hardware appliance and certificate management version

Which appliance version and certificate management software is running, check the versions screen within the CMI menu?

Certificate Alias

If the CSR was generated by the Swivel virtual or hardware appliance, but it is still not treated as a private key entry, then you may not have imported it with the same alias as the original self-signed certificate. If this is the case, try re-importing the certificate using the same alias as you used to request the certificate. Otherwise, we recommend starting again, using a self-signed certificate generated from the Swivel keystore.

trustedCertEntry

The certificate alias "swivel" is showing as a trustedCertEntry. That means it doesn't have an associated private key.

There are two possible reasons for this:

a. The CSR for this certificate was not generated from the Swivel virtual or hardware appliance

b. The original local certificate was deleted before installing the new certificate

The CMI makes backups of the keystore before it makes any changes, allowing a roll back to an earlier keystore. The backups are located under /backups. Restoring the keystore requires command line access.

```
cd /home/swivel
```

```
ls -al
```

This lists the folder contents, most of which will be backups of the keystore (starting with .keystore). Identify the correct one from the date and time, or the file size (the file immediately after the correct one will be slightly smaller, as the certificate has been deleted). To confirm the contents of a keystore, use the following command:

```
keytool -list -keystore <filename>
```

The keystore password can be obtained from SwivelSecure Support.

Look for a store that contains the alias "swivel" and has the type "PrivateKeyEntry".

Once the correct keystore has been located, make a copy of it:

```
cp <existing_filename> <new_filename>
```

Import the certificate response into the keystore as follows:

```
keytool -importcert -alias swivel -keystore <keystore_filename> -file <response_filename>
```

If this is successful, rename the file .keystore:

```
mv .keystore .keystore.saved
```

And then rename the modified keystore to .keystore:

```
mv <keystore_name> .keystore
```

Restart Tomcat in order to register the new keystore and check the certificates by connecting to the Swivel Administration Console.

Keystore ownership

Ensure that the .keystore file has ownership and group as swivel, to view the file:

```
ls -la /home/swivel/.keystore
```

to change the ownership

```
chown swivel:swivel /home/swivel/.keystore
```

Webmin still uses a self signed certificate

Importing the certificate does not affect webmin, only Tomcat.

Certificate not working for port 8443

If the certificate is not working for port 8443, it may be that you haven't imported it properly, or it may be a self-signed certificate that is being used instead.

Error Messages

keytool -importcert -keystore /home/swivel/.keystore -alias swivel -file /backups/upload/swivel.txt -trustcacerts Enter keystore password: keytool error: java.lang.Exception: Failed to establish chain from reply

The certificate has been imported without the required intermediate certificate. Import the intermediate bundle or individual intermediate certificates first (primary and secondary intermediates may be required), before importing the certificate response. Import the intermediate(s)/root e.g. intermediate or root depending on what you're importing (example below).

```
  keytool -importcert -keystore /home/swivel/.keystore -alias intermediate -file intermediate.txt -trustcacerts
```

Once the intermediates/root certificate has been imported you can then attempt to import the response again and you should get a success message.

bash: keytool: command not found

This error is seen when keytool cannot be found in the users path. This will be part of the Java path, and will depend upon the Java Version, Example: /usr/java/jre1.6.0_18/bin/keytool

keytool -importcert -keystore /home/swivel/.keystore -alias swivel -file /backups/upload/response.txt -trustcacerts

Enter keystore password:

keytool error: java.lang.Exception: Public keys in reply and keystore don't match

The imported certificate does not match against the keystore from which it was generated.

keytool error: java.lang.Exception: Alias <alias name> does not exist

The alias does not exist or has been incorrectly specified

Alias: swivel

Filename: swivel.crt

Please delete the existing certificate, before generating a new one.

The imported certificate does not match against the keystore from which it was generated. Try with a different keystore file. Ensure all keystore files are backed up and none are overwritten.

keytool -genkey -keysize 2048 -alias swivel -keyalg RSA -keystore /home/swivel/.keystore

Enter keystore password:

keytool error: java.lang.Exception: key pair not generated, alias <swivel> already exists

Certificate alias already exists, use a different alias name.

Certificate already exists in keystore under the alias <xyz>

Do you still want to add it?

If the intermediate certificate already exists, such as upgrading a certificate, then the intermediates do not need to be reimported.

SEVERE: Error starting endpoint java.io.FileNotFoundException: /home/swivel/.keystore (Permission denied)

SEVERE: Catalina.start: LifecycleException: service.getName(): "Catalina"; Protocol handler start failed: java.io.FileNotFoundException: /home/swivel/.keystore (Permission denied)

This can occur if the wrong permissions are set on the .keystore file, and it may stop Tomcat from starting. Ensure the correct permissions are set on the file.

keytool error: java.security.cert.CertificateParsingException: invalid DER-encoded certificate data

The certificate may be invalid, check to see that it has been correctly copied.

keytool error: java.lang.Exception: Input not an X.509 certificate

This error can be seen if an alias has not been created for the key entry or intermediate certificate.