Table of Contents

1 SSL Certificate PINsafe Appliance How to Guide 1

2 Overview 2

3 Prerequisites 3
3.1 Considerations 3

4 Certificate Management CMI menu option 4

5 Procedure Summary 5
5.1 1. Create a local certificate 5
5.2 2. Generate a CSR 7
5.3 3. Apply for the Certificate 9
5.4 4. Import the Certificate 9
5.5 5. Check that the certificate is valid 10
5.6 6. Delete the old selfsigned or old certificate alias if it exists 10
5.7 7. Restart Tomcat 10

6 Exporting certificate from Primary to Standby Server 11

7 Moving from one virtual or hardware appliance to another 12

8 Importing certificates generated on other virtual or hardware appliances and servers 13

9 Certificates using a CSR from a different appliance or servers 14

10 Known Issues 15
10.1 * in alias name 15
10.2 Keysize 15

11 Troubleshooting 16
11.1 Error Messages 17

12 SSL Disabling On Appliance 19
12.1 Overview 19
12.2 Prerequisites 19
12.3 How to Guide 19
12.4 Known Issues 19

13 SSL Solutions 20

14 SSL Certificate Solutions 21

15 Prerequisites 22

16 Exporting the Certificate from the Appliance 23

17 Importing Server Certificates from Another Machine 24
17.1 The Problem 24
17.2 Prerequisites 24
17.3 Solution - The Simpler Way 24
17.4 Solution - The Old Way 24

18 Managing Certificate Errors 26
18.1 Symptoms 26
18.2 Expired Certificates 26
18.3 Incorrect Hostname 26
18.4 Untrusted Signing Certificate 26

19 Troubleshooting 27
19.1 Wildcard certificate import problem checks 27

20 Tomcat 6 SSL 28
20.1 Overview 28
20.2 Prerequisites 28
20.3 Symptoms 28
20.4 Solution 28

1 SSL Certificate PINsafe Appliance How to Guide

2 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.

3 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

3.1 Considerations

3.1.1 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.

3.1.2 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.

3.1.3 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.

3.1.4 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.

3.1.5 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.

3.1.6 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

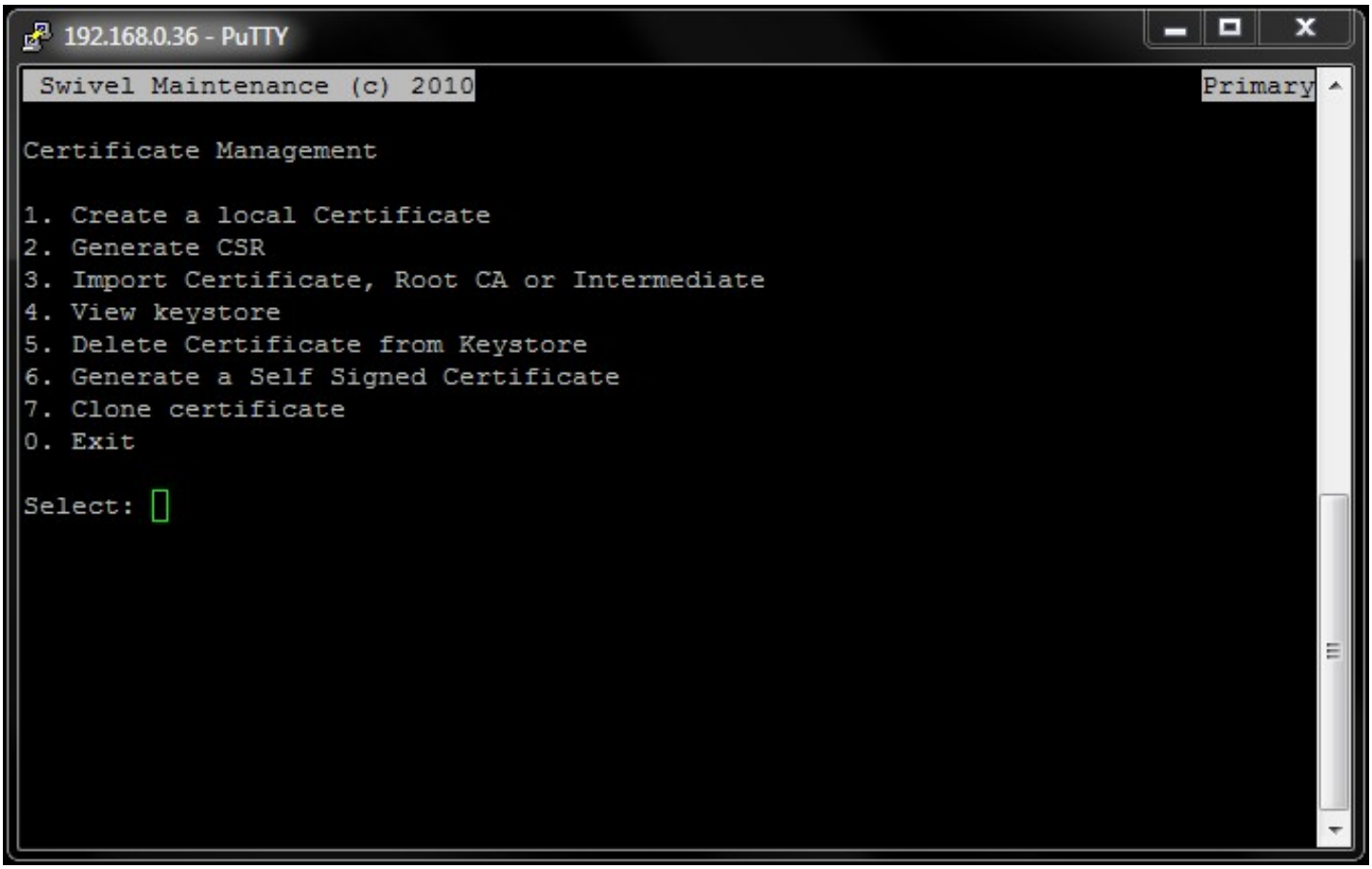
3.1.7 Internal Certificate Authority

See SSL Internal Certificate Authority

4 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):



5 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.

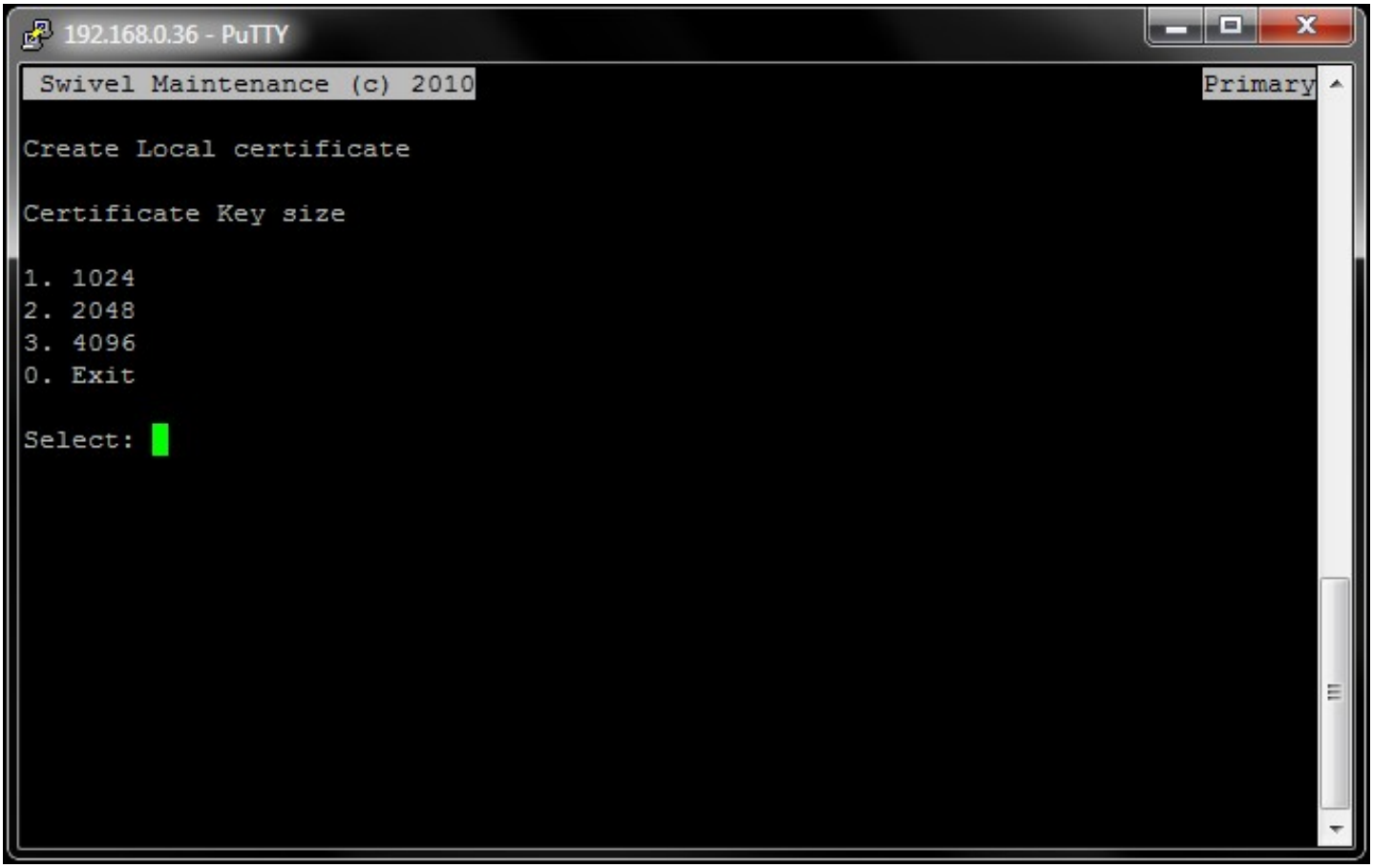
The following article sections detail how this is done via the Certificate Management CMI menu option.

5.1 1. Create a local certificate

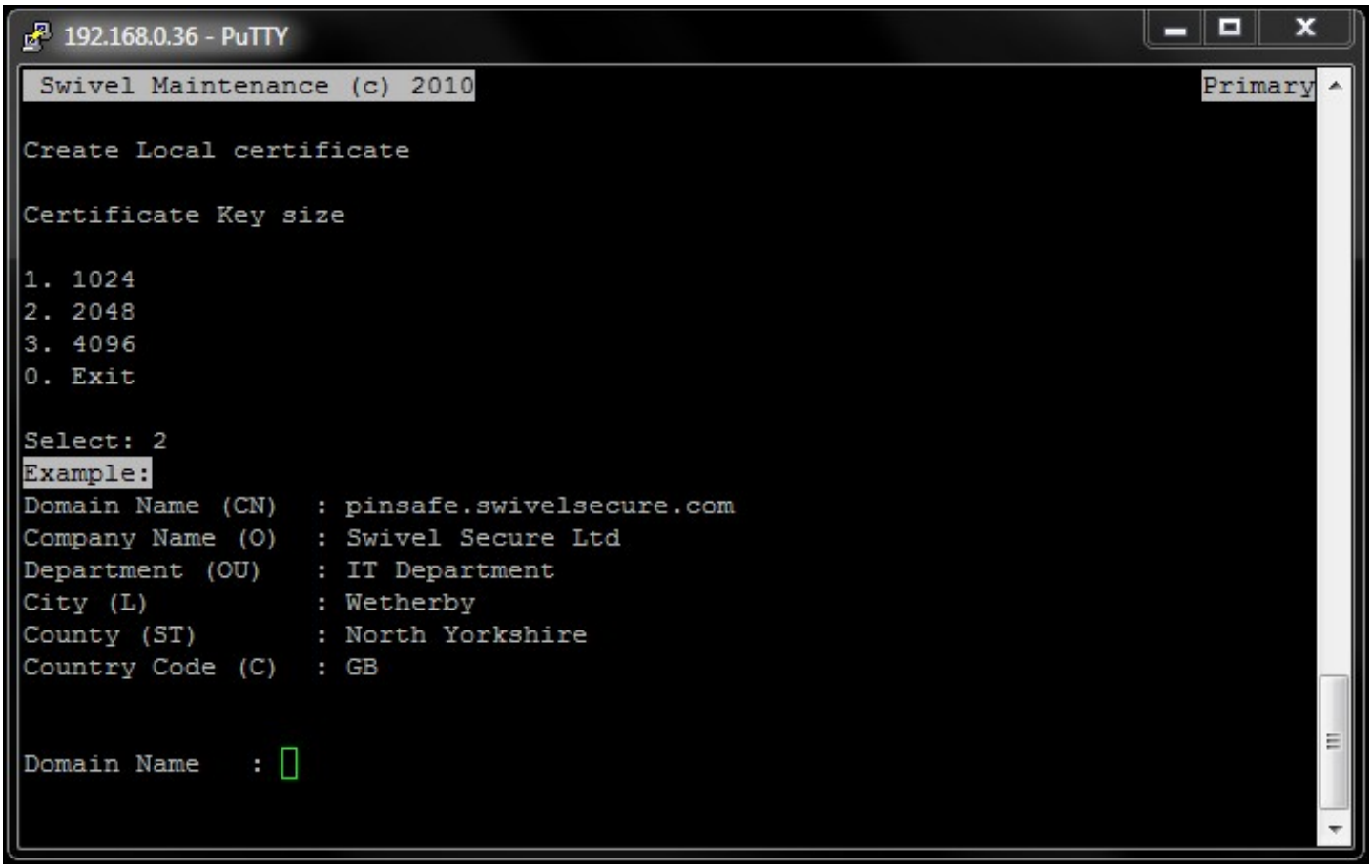
From the Certificate Management menu, select the Create a local Certificate option.

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"**.

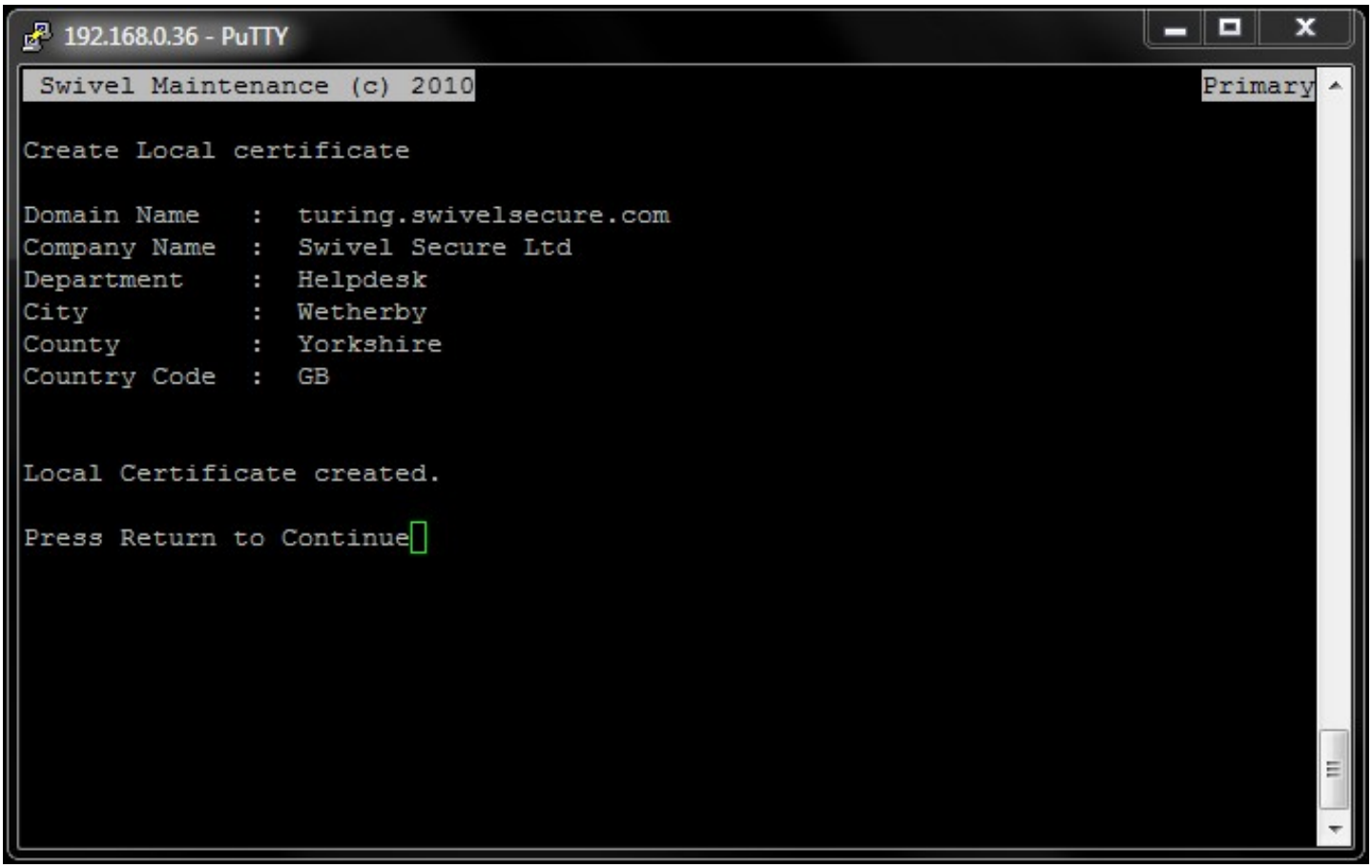
Here is a screen shot of the first screen you encounter when selecting the Create a local Certificate option. At the time of writing this article, most CAs (Certificate Authorities) require at least 2048 bit key size.



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.



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.

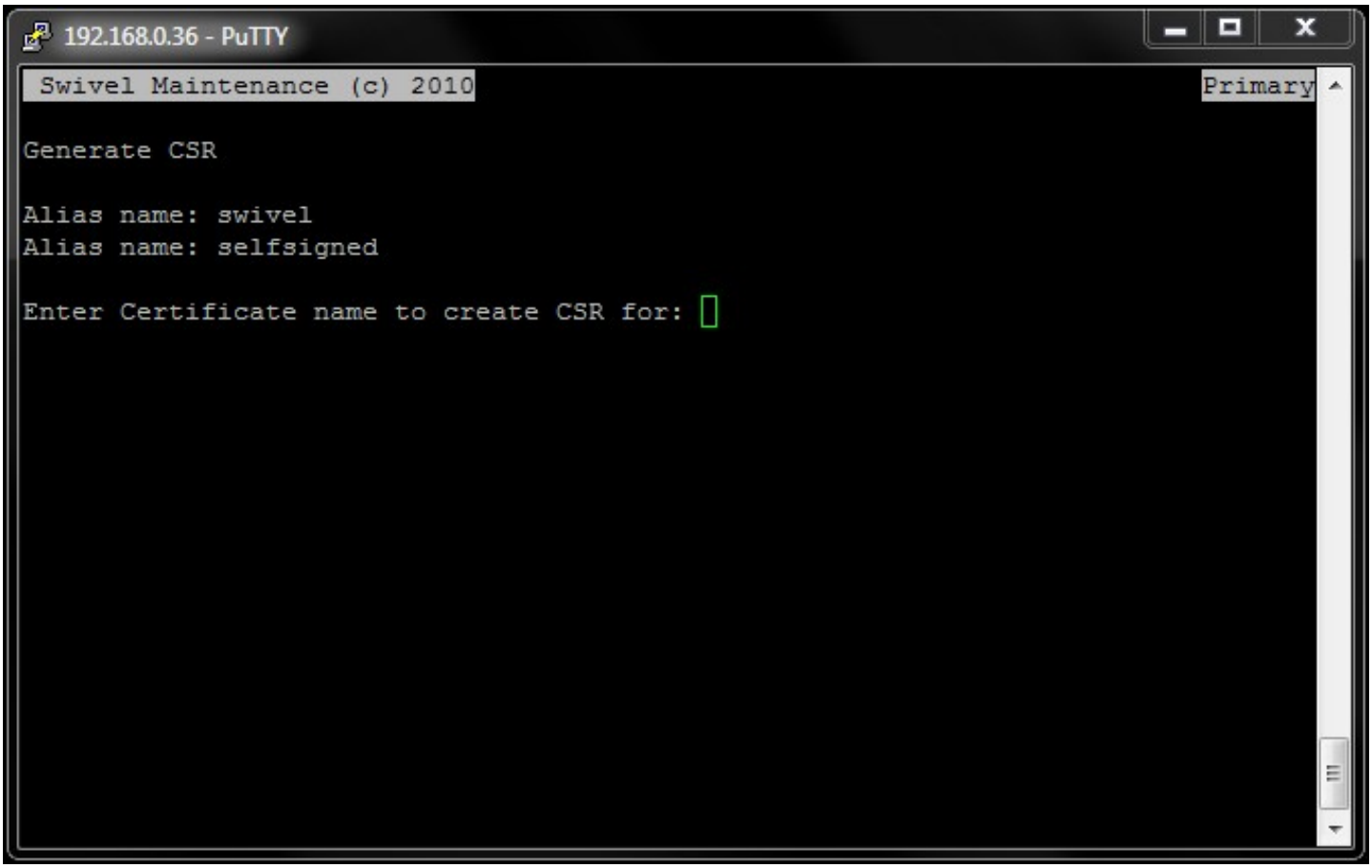


Now you have created a local Certificate, you can generate a CSR (Certificate Signing Request) from it.

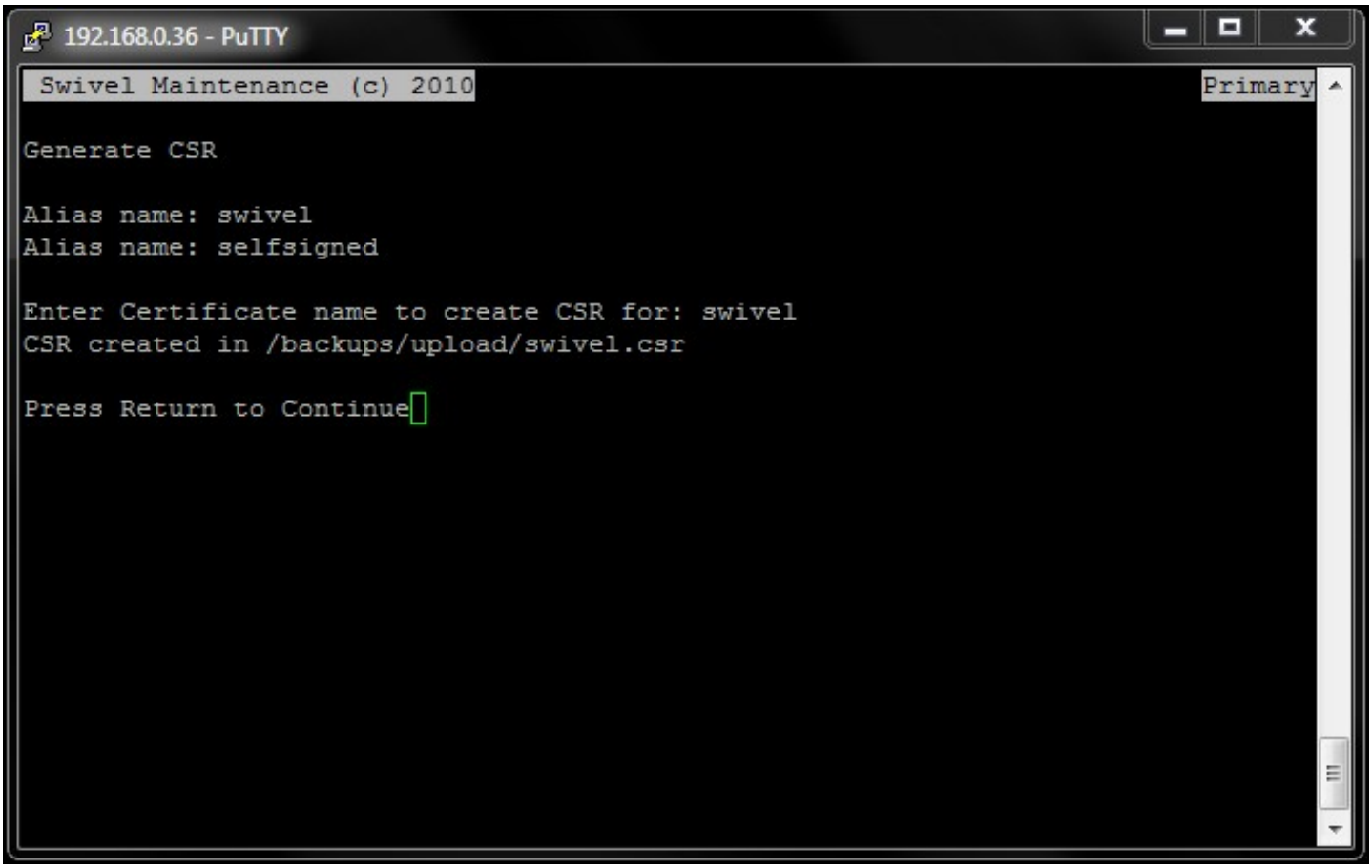
5.2 2. Generate a CSR

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.



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.



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.

5.3 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.

5.4 4. Import the Certificate

5.4.1 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.

5.4.2 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.

5.4.3 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.

5.5 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

5.6 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.

5.7 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

6 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

7 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.

8 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.

9 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.

10 Known Issues

10.1 * 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 *.

10.2 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.

11 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.

11.1 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.

12 SSL Disabling On Appliance

12.1 Overview

This article explains how to disable SSL on a PINsafe appliance, so that pages can be accessed using http, rather than https.

NOTE: carrying this out is a security risk. Allowing users to access PINsafe without SSL encryption is inherently insecure. In particular, it is not recommended to allow http access to a production environment over the internet. This solution is only advised for pre-production testing, or if the PINsafe appliance is only accessible on the internal network. If you are implementing this change simply to avoid problems with certificate errors, the correct solution is to get a commercial SSL certificate for production use.

12.2 Prerequisites

These instructions assume you have a PINsafe appliance with Webmin. Otherwise, the instructions apply to all versions of PINsafe.

12.3 How to Guide

NOTE: this process involves restarting Tomcat, so PINsafe services will be unavailable for a short time.

Open webmin in a web browser (https://<pinsafe_server>:10000). Replace <pinsafe_server> with the name or IP address of your PINsafe server. You will need to log in, so make sure you know the administrator password for your PINsafe appliance.

Select **Servers** from the top menu, then **PINsafe**.

Select **Edit Tomcat Config File**.

Assuming you want to disable SSL only for the applications on port 8443 (proxy, changepin, reset), locate the line that starts as follows:

```
<Connector address="0.0.0.0" port="8443" scheme="https"
```

Delete everything from **scheme** up until the end marker: **/>**. The line should now look like this:

```
<Connector address="0.0.0.0" port="8443" />
```

If you also want to disable SSL for the pinsafe application (not recommended), then locate the line starting

```
<Connector address="0.0.0.0" port="8080" scheme="https"
```

and carry out the same procedure.

Click the Save button to return to the menu.

Finally, restart Tomcat to implement the changes. Most versions of PINsafe have the option to restart Tomcat on the webmin page you just returned to. However, this is not available on all versions, in which case you will have to restart Tomcat from the CMI menu on the appliance console.

12.4 Known Issues

Note that if you enable http in order to display a TURing image without certificate errors, and the image is embedded into a page that is using https, you may get warnings about mixing secure and non-secure elements on a web page. Read the warning carefully before choosing which button to click, as the response for Internet Explorer (IE) in particular has changed. In older versions of IE (6 or earlier), you selected "Yes" to allow mixed content. In newer versions, you select "No" to allow mixed content: "Yes" means show only secure content.

13 SSL Solutions

14 SSL Certificate Solutions

This document discusses problems people have when using HTTPS and SSL certificates within Swivel:

- Handling invalid certificates
- Importing certificates from another machine

Further information on SSL certificates with appliances can be found in the SSL Certificate PINsafe Appliance How to Guide, and for non appliance installations see Tomcat 6 SSL.

15 Prerequisites

- Some of the solutions below require a tool to manipulate Java keystores. The one we recommend is Keystore Explorer.
- DNS name for the Swivel instance, usually the public IP address

16 Exporting the Certificate from the Appliance

This may be required where the certificate is site certificate and is to be used elsewhere. It is generally simpler to take a copy of the Appliance certificate store to your local machine and work on it there. For this, you can either use Webmin, or an application such as WinSCP. If you use WinSCP, use the same credentials as you normally do to connect to the Appliance console.

The file you need to retrieve is /home/swivel/.keystore. Make a copy of it on your local PC.

To extract the certificate, you will need Keystore Explorer, mentioned above. Install it, if you have not already done so, and run it. Click on "Open an existing KeyStore", and locate the keystore you have just downloaded. When asked for a password, it is "lockbox".

Locate the appliance certificate, which should be called "swivel" if you have created it using the CMI. Right-click on in and select Export, then "Export Certificate Chain". Accept the default settings on the next page, and Save the file with an extension of .cer.

17 Importing Server Certificates from Another Machine

This section refers to installing private/public key pair server certificates within Swivel. For details on trusting public key certificates, see above.

17.1 The Problem

Many customers already have a commercial certificate (e.g. a wildcard certificate) installed on a Windows server, and want to use the same certificate on a Swivel appliance.

Note: we recommend that you generate the Appliance SSL certificate using the CMI menus. You should only use this technique if you already have a suitable certificate, and do not want to spend more money on another one.

17.2 Prerequisites

It is essential for this solution that you have both the public and private keys for the certificate. The public key certificate returned by the certificate authority in response to a certificate signing request is not enough. Also, the certificate needs to be marked as exportable.

17.3 Solution - The Simpler Way

17.3.1 Export the certificate as a .pfx file

The certificate must be exported and the step below explains how to do this. If this step is carried out by another party then the certificate will need to be supplied, usually as a pfx file with a password.

Open the Certificate management plug-in as described earlier: Select Start, Run and enter "MMC". Click File -> "Add/Remove Snap-in". In the pop-up dialog, click Add, then select Certificates from the list. Click Add, and make sure you select Computer Account, then Local computer.

Navigate to Personal, then Certificates, and locate the certificate you wish to export. Right-click on it, then select "All Tasks", "Export". Make sure you select Yes to importing the private key. It is also recommended that you choose to include the certificates in the certification path.

You will be required to enter a password for this file. You will need to enter this when you import the certificate to the appliance later on. Note that older appliances had to use "lockbox" as a password, and entering anything else would cause problems. This is no longer an issue.

17.3.2 Transfer the PFX file to the appliance

You need to copy the PFX to your appliance. You can do this using WinSCP or any SCP/SFTP file transfer tool.

The credentials to connect to the appliance are the same as your login credentials for the CMI: username admin and your password: you should have been given an initial password, but it is recommended you change that.

The file must be copied to the appliance in the folder /backups/upload in order for it to be visible to the CMI.

17.3.3 Install the PFX file as the new certificate

- Log into the appliance CMI using your credentials.
- Select the Tomcat menu.
- Select the Certificates menu.
- Select the option "Import / Roll Back to Previous Keystore"
- Select the option "Import Keystore"
- Choose your certificate from the list shown
- Enter the password when prompted

You will be prompted to restart Tomcat before the new certificate is active. Select Yes if you are confident the certificate is correct. If you want to check it has been successfully installed first, select No and then go back to the Certificate menu and select View Keystore. There should be an entry corresponding to your certificate. Select that and review it, checking the owner, issuer, validity and entry type: this last should be PrivateKeyEntry.

Once you are happy the certificate is correct, you must restart Tomcat to make it active, assuming you didn't do this in the step above. Do this from the Tomcat menu.

17.4 Solution - The Old Way

This documentation is provided for backward compatibility. It won't normally be required for new appliances, unless the method above fails.

17.4.1 Export the certificate as a .pfx file

This is the same first step as above.

17.4.2 Import the PFX into KeyStore Explorer

Open Keystore Explorer and select "Open an existing Keystore". Select the PFX file you just exported.

17.4.3 Convert the keystore into a JKS keystore

NOTE: this is not necessary on a version 4.x appliance. It can import PFX keystores directly.

From the Tools menu, select "Change Type", then "JKS". Accept any warnings displayed.

Warning: despite exporting the certificate with a password of "lockbox" in the above steps, this password will be lost when you do the conversion. So you need to change the password again within Keystore Explorer, in two places - the keypair and the keystore. First, right click the keypair entry in the Keystore Explorer window and set the keypair password to be lockbox. Then go to Tools and Set the keystore password to be lockbox.

Warning: if you are converting a wildcard certificate, the certificate alias (the name displayed) will probably contain a "*" character. This doesn't affect Tomcat's ability to use the certificate, but it does cause problems if you want to view the certificate using the appliance CMI later. Since the alias is just a name used to identify the certificate, it doesn't have to be the same as the certificate subject. You can rename the certificate by right-clicking on it and selecting "Rename". You will be asked for the password again before entering the new name.

After you have made any necessary changes, save the file with a different name. It is recommended you use an extension of either .jks or .keystore.

17.4.4 An alternative method of importing PFX files

If the above method fails to import the private key, which has been observed on occasions, try the following, slightly modified method instead:

- Create a new keystore of type "JKS".
- From the Tools menu, select "Import Key Pair".
- Select type "PKCS #12".
- Enter the password and PFX file name.
- Enter a name for the certificate (see the notes above).
- Enter a new password

Continue as above.

17.4.5 Copy the keystore on to the Swivel Appliance

You can do this with Webmin, but we recommend using WinSCP or similar, as it has more file management options.

NOTE: any changes you make will not take effect until you restart Tomcat.

Firstly, make a backup copy of the current Tomcat keystore, which is /home/swivel/.keystore. We recommend that you keep a copy of this file off the Appliance anyway, for recovery purposes, but in this particular instance, we also recommend that you rename .keystore to something else so that you can restore it quickly if anything goes wrong.

Now copy the new keystore you have just created into the /home/swivel folder, and rename it to .keystore.

Check the owner of the file, (using right-click, Properties in WinSCP). If the owner is not swivel, then change both owner and group to swivel.

Restart Tomcat and check that you can access the Swivel admin console. If you can't, check the file catalina.out for errors.

18 Managing Certificate Errors

18.1 Symptoms

At best, certificate errors can cause warning messages to be displayed in your browser. At worst, they will stop https requests from working at all. This is particularly a problem with embedded TURing images where the hostname is not the same as the main page, or where a server needs to make requests to Swivel behind the scenes, and it is not possible to configure the server to ignore these errors.

There are essentially three ways in which SSL certificates can be invalid:

- The certificate has expired
- The certificate hostname does not match the request hostname
- The certificate is not issued by a trusted authority

18.2 Expired Certificates

There is nothing that can be done about expired certificates, except to renew the certificate. See the Swivel Appliance guide for details on renewing a Swivel certificate.

18.3 Incorrect Hostname

There is little that can be done about this: the hostname within the request **must** match the hostname in the certificate. However, there are some techniques you should be aware of that can help to ensure this.

First of all, always use hostnames when specifying HTTPS requests, never IP addresses. You can get away with IP addresses when using certain Swivel tools that allow you to ignore certificate requests, but when accessing direct from the browser, or when making requests with built-in tools, you should always use the hostname specified by the SSL certificate.

One problem that can arise is that the certificate hostname cannot be resolved, or resolves to an external IP address when an internal one is required. This can be overcome by editing the hosts file on the Swivel Agent. On Windows machines, this file is located in C:\Windows\System32\drivers\etc. It is a simple text file, and all you need to do add a new line consisting of the target IP address, followed by one or more spaces or tabs, followed by the hostname. Note that this solution will affect **all** references to this hostname from this machine, so make sure it is not required to access the external IP address by the same hostname, if relevant.

18.4 Untrusted Signing Certificate

This is a problem that can be resolved relatively easily in a number of ways.

This simplest but most expensive solution is to buy a commercial certificate from a recognised authority. This authority should be included in your list of trusted root certificates already.

You can also sign your own certificates. The Swivel appliance has an option to generate a self-signed certificate, which is fine for a one-off solution. You can also set up your own certificate signing authority. Windows Server operating systems include a Certificate Services option, which allows you to accept and sign certificates.

The problem with signing your own certificates is that they are not trusted by default. However, you can add any certificate to the trusted root certificates store on your computer. All you need is a copy of the signing authority public key certificate. For a Swivel appliance, getting the certificate is described below. For Windows Certificate Services, you can download a copy of the root CA certificate.

To install certificates on a Windows machine, select Start, Run and enter "MMC". Click File -> "Add/Remove Snap-in". In the pop-up dialog, click Add, then select Certificates from the list. Click Add, and make sure you select Computer Account, then Local computer. Close all dialogs, and expand the Certificates tree to "Trusted Root Certification Authorities", then "Certificates". Right-click and select All Tasks -> Import. Select the root certificate you have already retrieved, and install it.

19 Troubleshooting

19.1 Wildcard certificate import problem checks

Does the certificate have both a private and public key?

Are the keystore and certificate passwords the same? If the password is not the Swivel default then you need to change that in the Tomcat server.xml file.

Are the intermediate and root certificates included in the keystore file.

Is the file named ".keystore" when you copied it to the appliance?

Check the permissions and that the owner is "swivel"?

20 Tomcat 6 SSL

20.1 Overview

Additional configuration to Tomcat to allow SSL.

20.2 Prerequisites

Tomcat 6 PINsafe 3.x

20.3 Symptoms

org.apache.catalina.startup.SetAllPropertiesRule begin WARNING: [SetAllPropertiesRule]{Server/Service/Connector} Setting property 'keystoreFile' to 'conf\certificate\.keystore' did not find a matching property. org.apache.catalina.startup.SetAllPropertiesRule begin WARNING: [SetAllPropertiesRule]{Server/Service/Connector} Setting property 'keystorePass' to 'password' did not find a matching property.

and

SEVERE: Error initializing endpoint java.lang.Exception: Invalid Server SSL Protocol SEVERE: Catalina.start LifecycleException: Protocol handler initialization failed: java.lang.Exception: Invalid Server SSL Protocol

20.4 Solution

Add protocol=?org.apache.coyote.http11.Http11NioProtocol? to the server.xml file in <path to Apache Tomcat>\Tomcat 6.0\conf

Example:

C:\Program Files\Apache Software Foundation\Tomcat 6.0\conf\server.xml

Example:

From:

```
<Connector port="8443" protocol="HTTP/1.1" SSLEnabled="true"
           maxThreads="150" scheme="https" secure="true"
           clientAuth="false" sslProtocol="TLS" />
```

To:

```
<Connector port="8443" protocol=?org.apache.coyote.Http11NioProtocol? SSLEnabled="true?
           maxThreads="150" scheme="https" secure="true?
           clientAuth="false" sslProtocol="TLS" />
```