Client Authentication using Certificates

Contents

Client Authentication using Certificates during SSO

Overview

This article describes how to setup and configure Certificate authentication using AuthControl Sentry SSO. Some Linux knowledge and experience with certificates is recommended as this involves command line work and preparation of certificates from the CA within your enterprise. To establish Client Authentication with certificates in AuthControl Sentry, a Java keystore should be created with a certificate that is signed by your Enterprise CA. Changes then need to be made to the Apache Tomcat application server configuration, on the webapps2 connector entry in the server.xml configuration file.

Pre-requisites

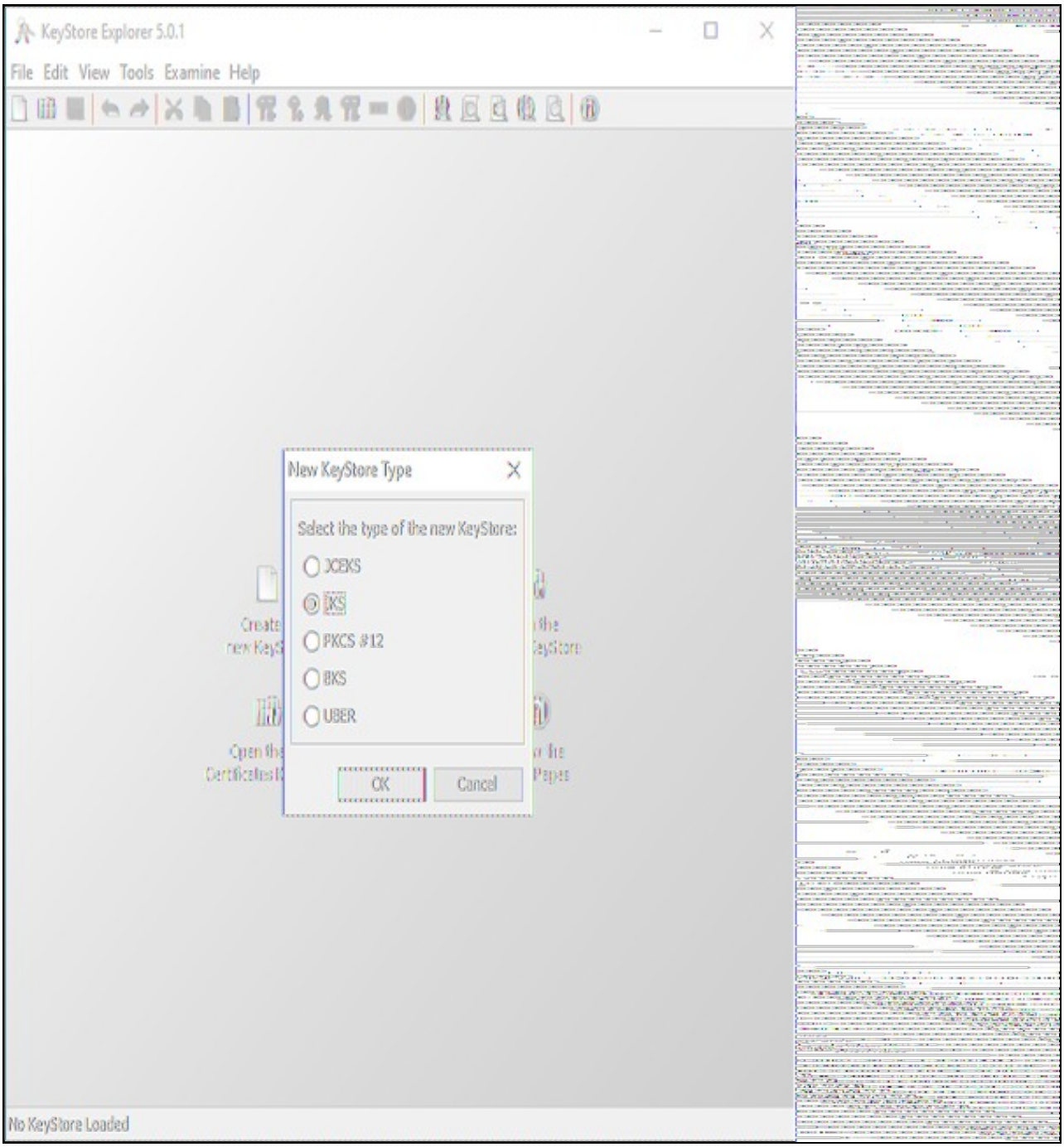
- Certificate Authority within your organisation for signing certificates
- Creation of user certificate for ?Client Authentication? purposes with Private Key
- Keystore Explorer (freeware) installed on your workstation, to create and view Java Keystore files
- Command line access to the Swivel Secure appliance
- Some experience with the *vi* Linux command for file editing purposes

Setup a Server-Side Client Authentication keystore (truststore)

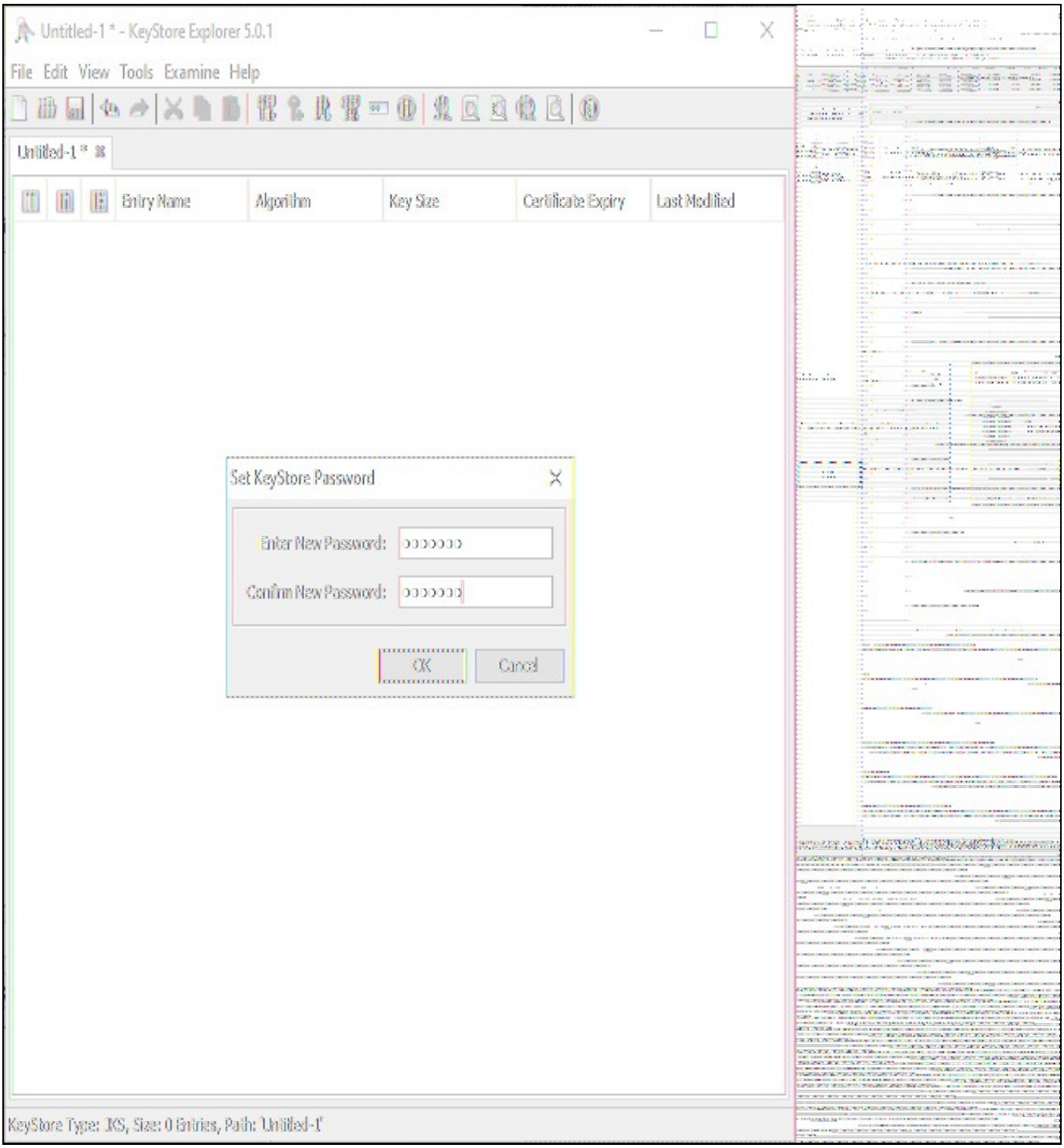
An easy method to create a new Java Keystore is by using a free third-party application called Keystore Explorer. The alternative method would be to take a copy of the keystore associated with the Apache Tomcat connectors (providing https to users connecting to ports 8080 and 8443) and modify it.

Create a new Java keystore

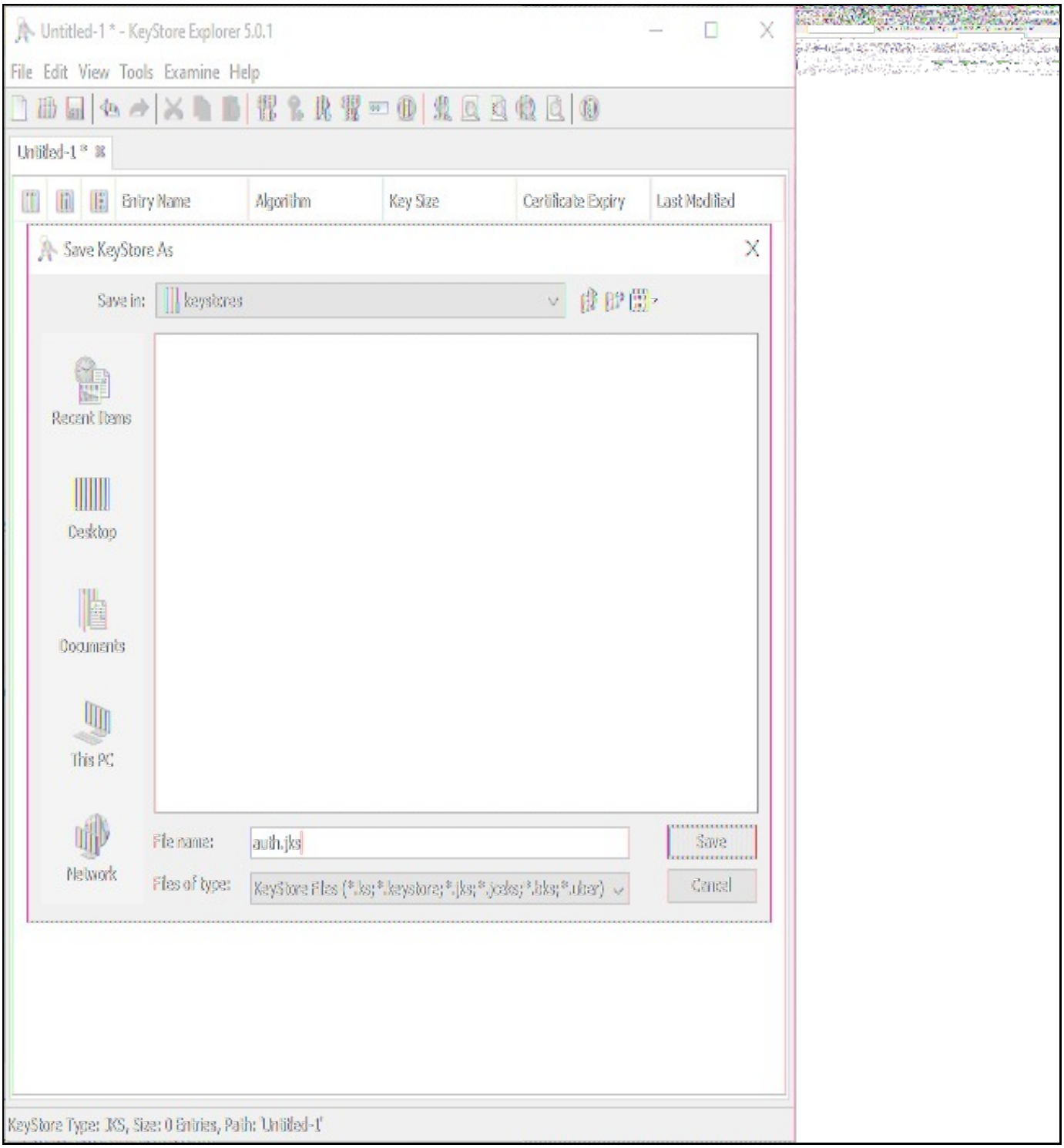
Create a new keystore e.g. auth.jks Import your Root CA certificate Import any intermediary CA certificates (especially if the certificate on the device was signed by those Intermediaries)



Click File -> Save As. You will first be prompted for a Keystore password. Enter the password as ???lockbox???



Save the file as auth.jks

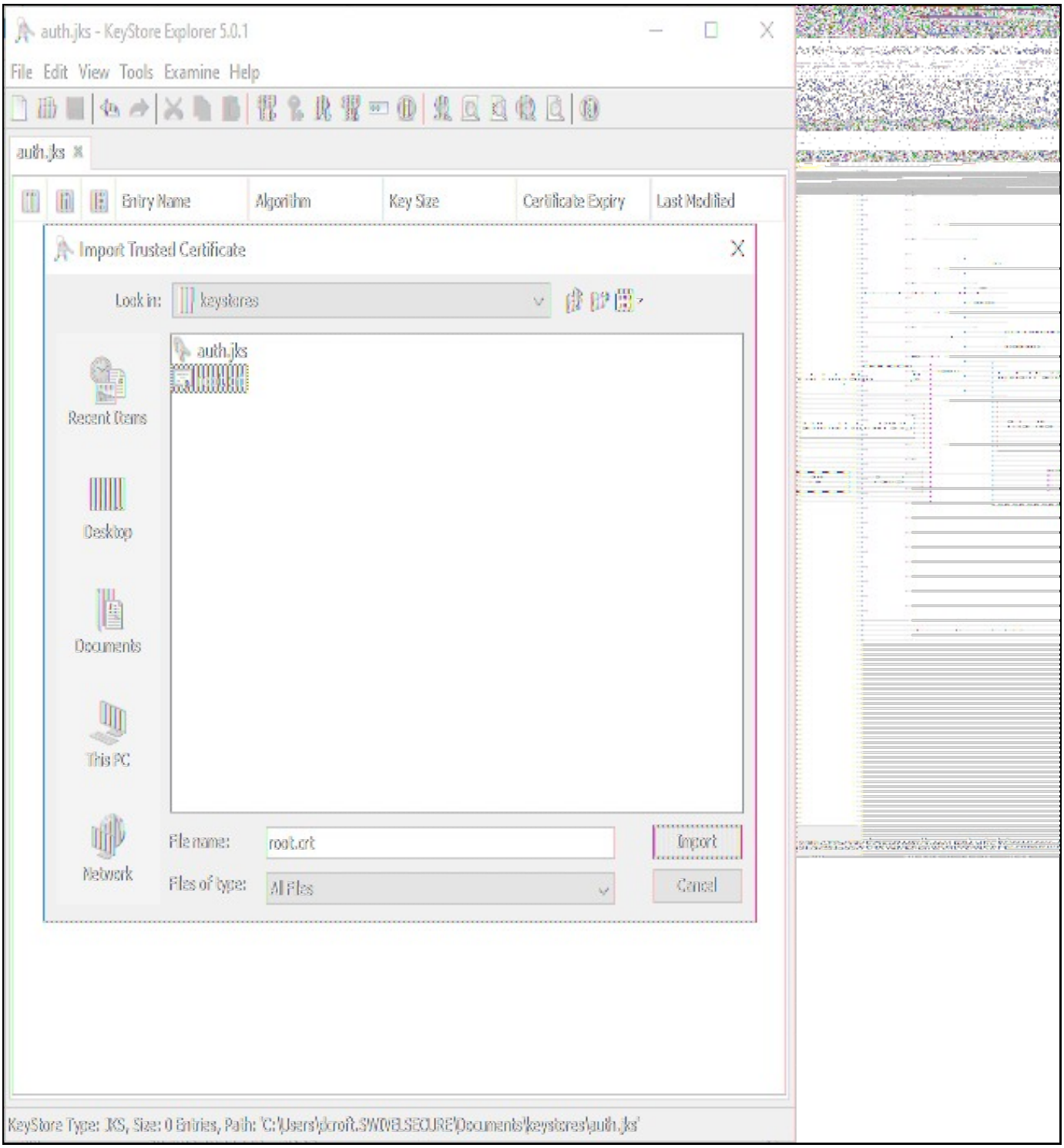


Import your CA certificate(s)

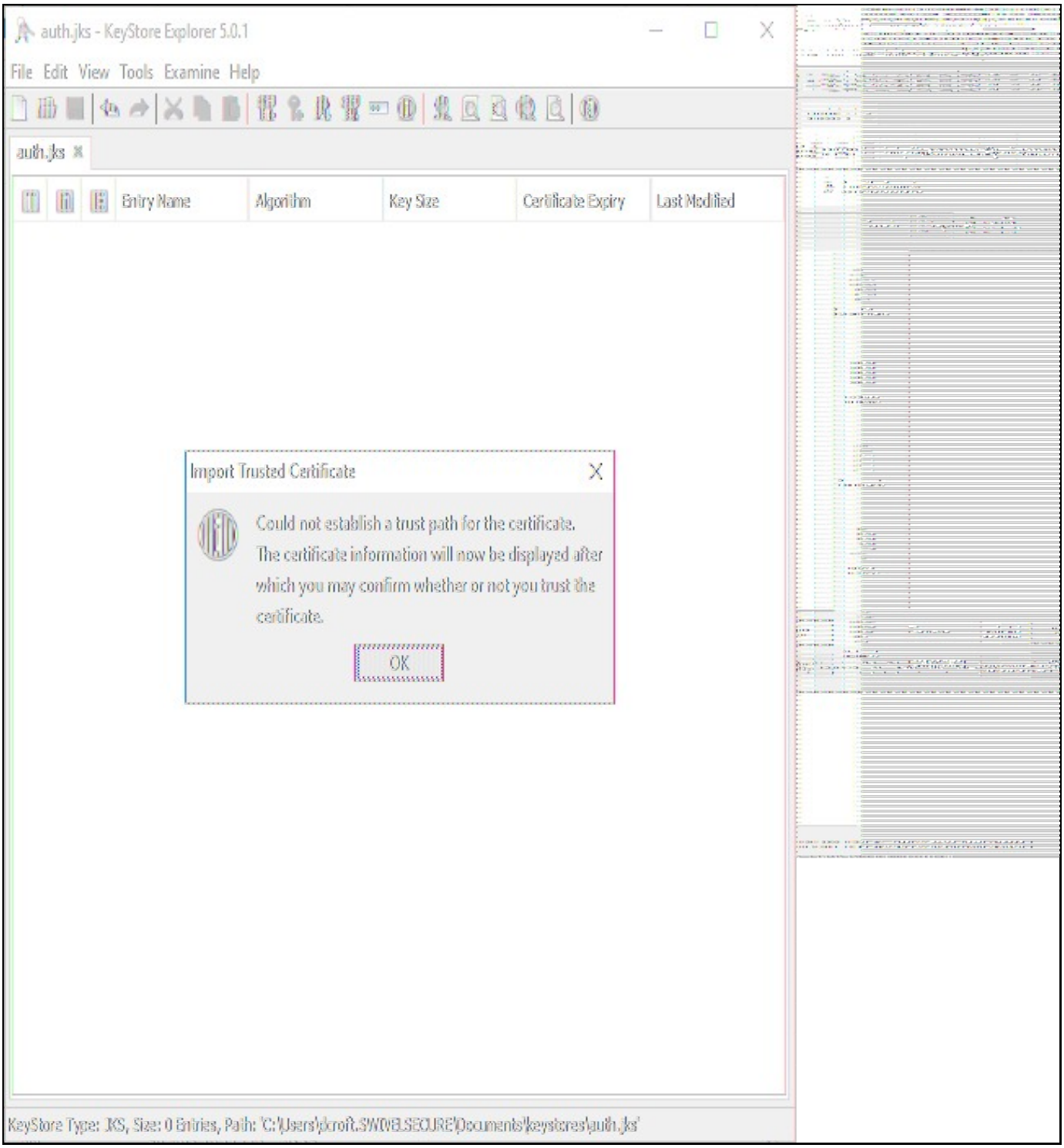
Next, import your Intermediate AND/OR your Root CA certificates into the keystore, so that any client certificates generated by your PKI can be validated during RBA using this keystore.

To import your CA trusted certificates, goto Tools -> Import Trusted Certificate.

Locate and select the certificate, then click Import.



You may receive this warning, where you will be prompted to trust the certificate you have imported:



Click OK and follow the onscreen prompts to review and accept the certificate as trusted.

auth.jks - KeyStore Explorer 5.0.1

File Edit View Tools Examine Help

auth.jks

Certificate Details for File 'root.crt'

Certificate Hierarchy:

Version: 3

Subject: CN=AddTrust External CA Root,OU=AddTrust External TTP Netwo

Issuer: CN=AddTrust External CA Root,OU=AddTrust External TTP Netwo

Serial Number: 0x1

Valid From: 30/May/2000 11:48:38 BST

Valid Until: 30/May/2020 11:48:38 BST

Public Key: RSA 2048 bits

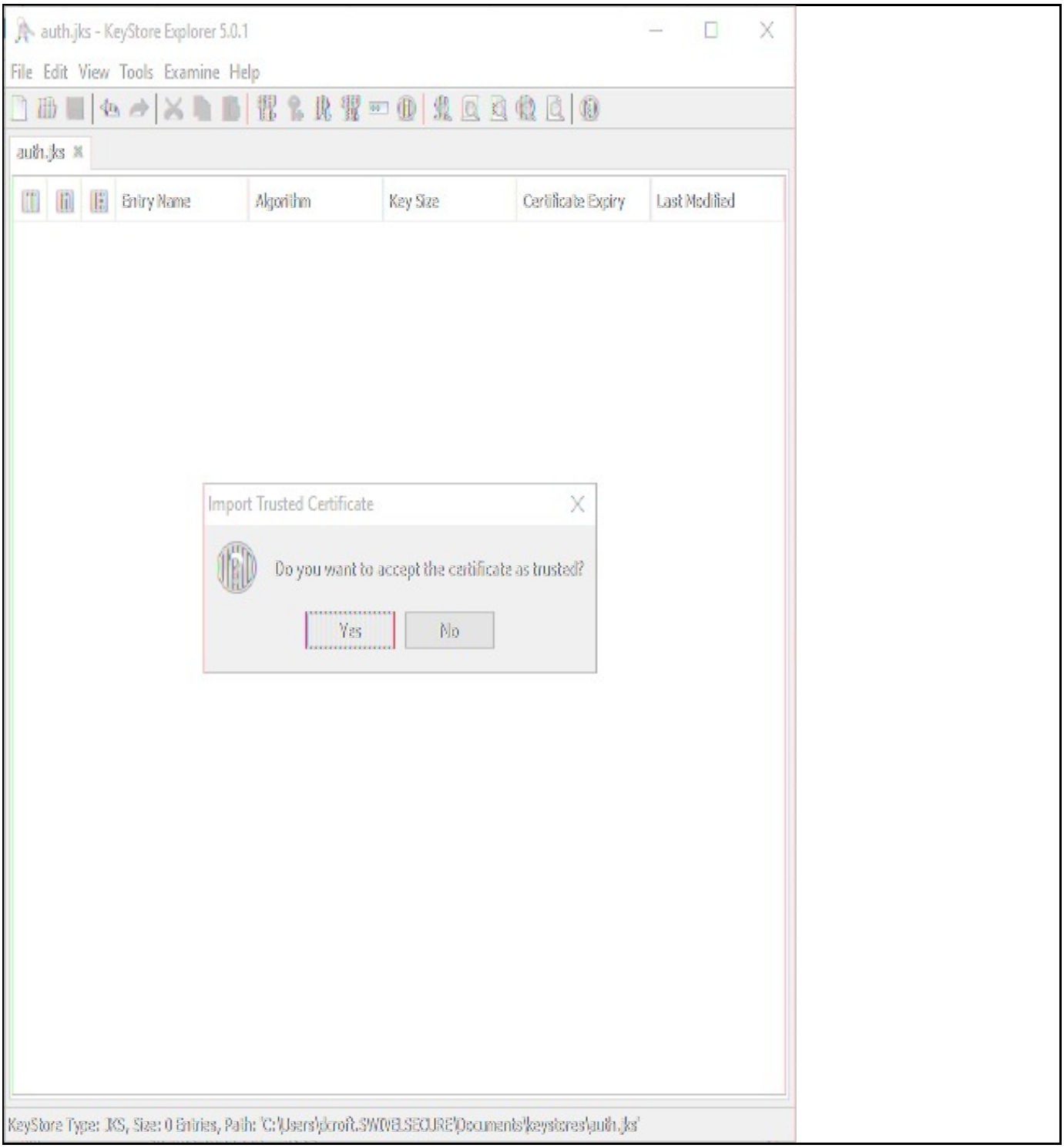
Signature Algorithm: SHA-1 with RSA

Fingerprint: SHA-1 02:FA:F3:E2:91:43:54:68:60:78:57:69:4D:F5:E4

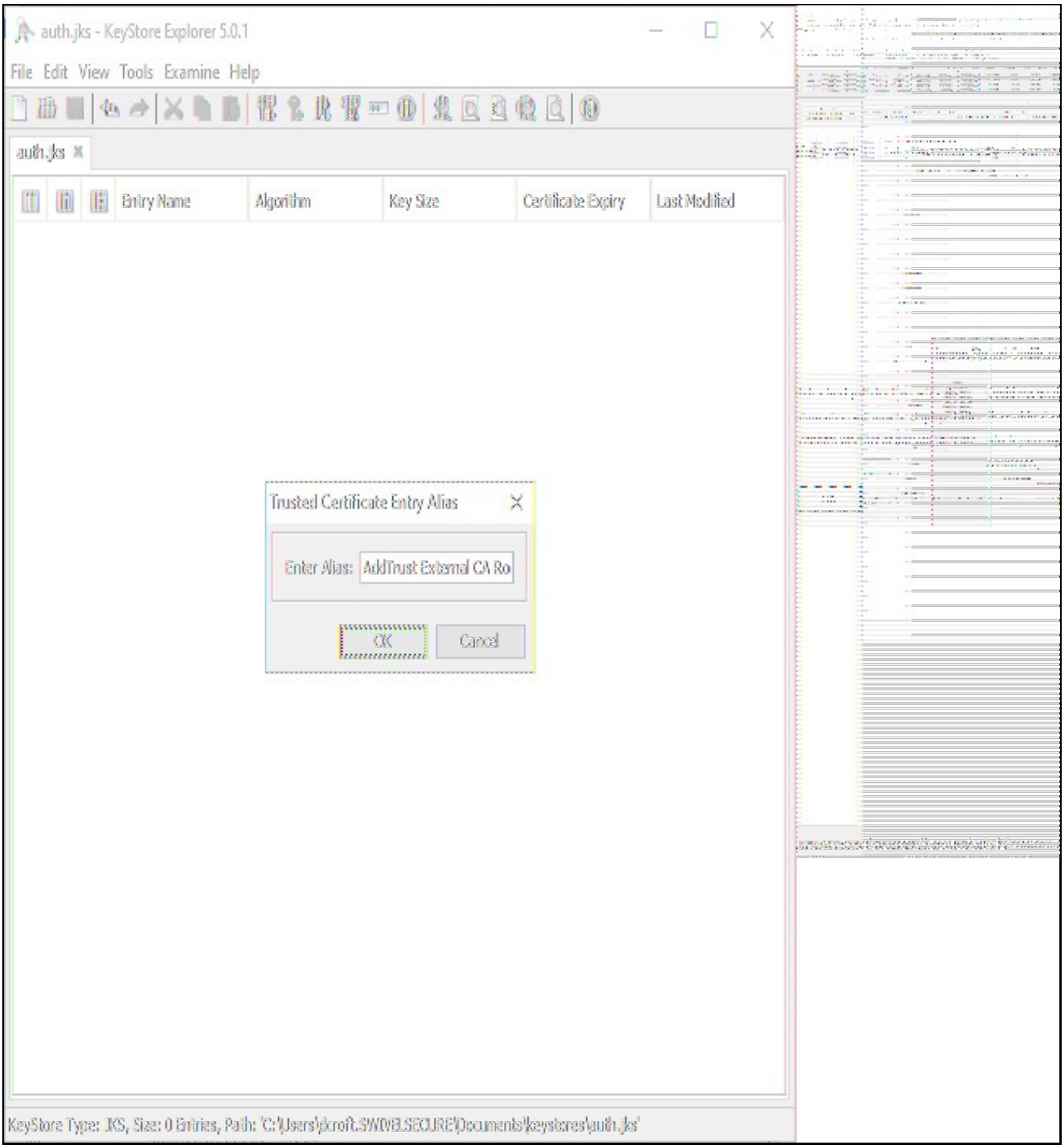
Extensions PEM ASN.1

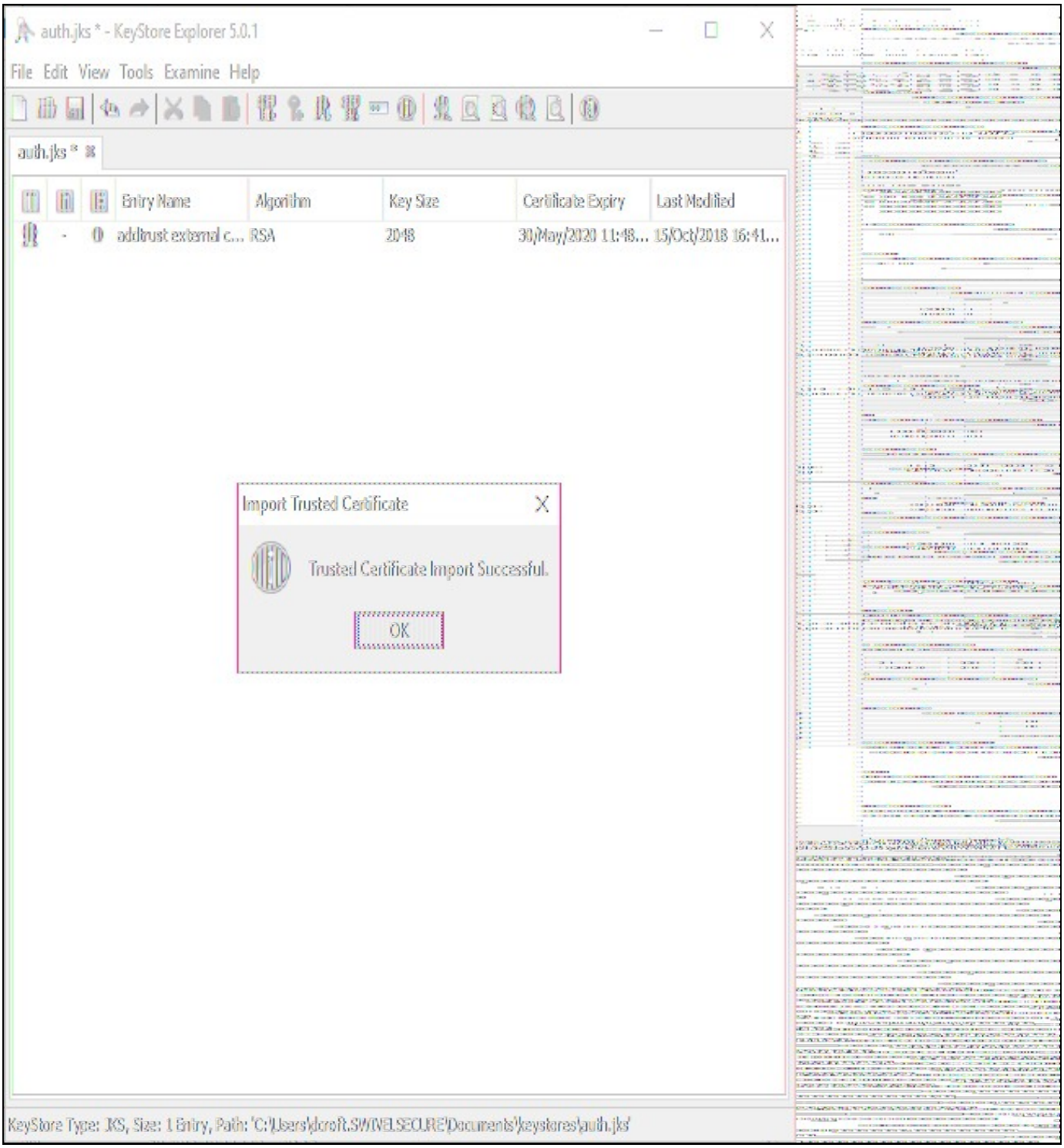
OK

KeyStore Type: JKS, Size: 0 Entries, Path: 'C:\Users\jcroft.SWOBLSECURE\Documents\keystores\auth.jks'



Enter a meaningful alias for the new CA certificate as it will appear in the keystore:


auth.jks - KeyStore Explorer 5.0.1
File Edit View Tools Examine Help
auth.jks
Entry Name
Algorithm
Key Size
Certificate Expiry
Last Modified
Trusted Certificate Entry Alias
Enter Alias:
AddTrust External CA Ro
OK
Cancel
KeyStore Type: JKS, Size: 0 Entries, Path: 'C:\Users\jcroft.SWOBLSECURE\Documents\keystores\auth.jks'



Save your changes, using the File -> Save menu option.

Modify the Apache Tomcat server.xml

On the Swivel Secure appliance, take a backup of the /usr/local/tomcat/conf/server.xml file, prior to making the necessary changes.

For the 8443 connector entry, add the following parameters:

```
clientAuth="want"
truststoreFile="/home/swivel/.swivel/auth.jks"
truststoreType="JKS"
truststorePass="lockbox"
```

So that it looks like this...

Before:

```
<Service name="webapps2">
<Connector SSLEnabled="true" acceptCount="100" address="0.0.0.0" ciphers="TLS_ECDHE_DSS_WITH_AES_128_GCM_SHA256,TLS_DHE_ECDSA_WITH_AES_128
    <Engine defaultHost="localhost" name="webapps2">
      <Host appBase="webapps2" autoDeploy="true" name="localhost" unpackWARs="true">
```

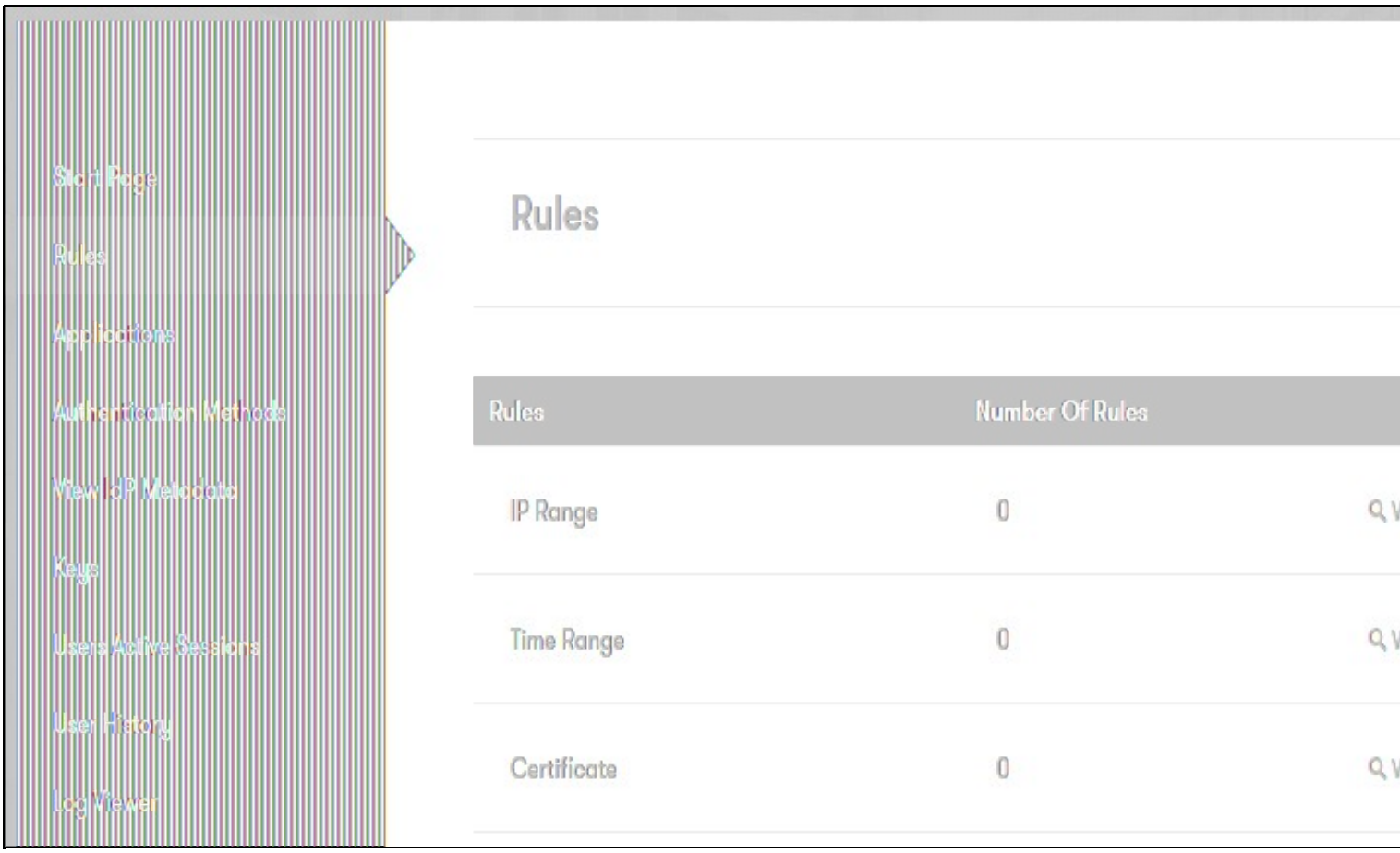
```
        <Valve className="org.apache.catalina.valves.AccessLogValve" directory="logs" pattern="common" prefix="webapps2" rotatable="false"
      </Host>
    </Engine>
</Service>
```

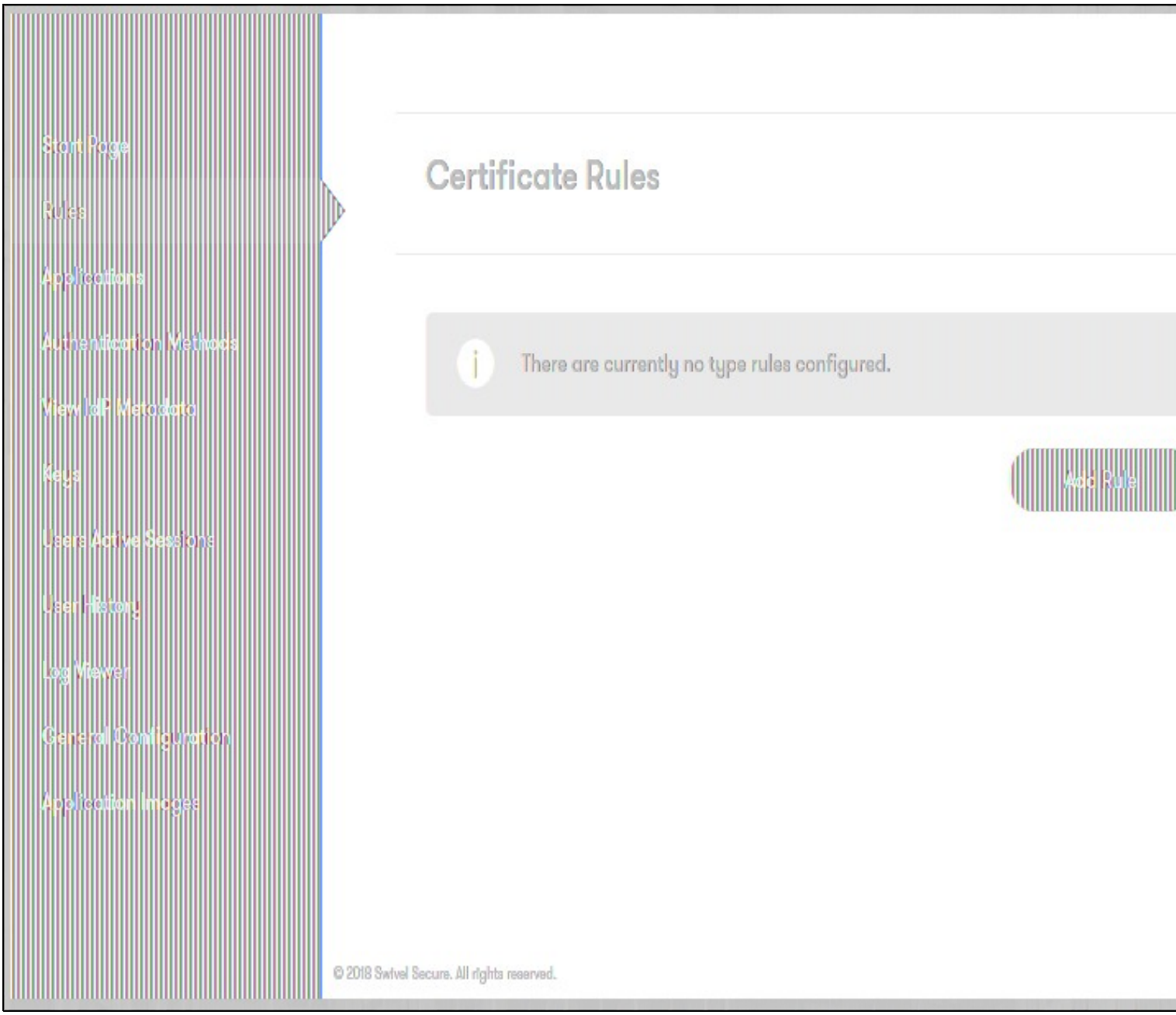
After:

```
<Service name="webapps2">
<Connector SSLEnabled="true" acceptCount="100" address="0.0.0.0" ciphers="TLS_ECDHE_DSS_WITH_AES_128_GCM_SHA256,TLS_DHE_ECDSA_WITH_AES_128
    <Engine defaultHost="localhost" name="webapps2">
      <Host appBase="webapps2" autoDeploy="true" name="localhost" unpackWARs="true">
        <Valve className="org.apache.catalina.valves.AccessLogValve" directory="logs" pattern="common" prefix="webapps2" rotatable="false"
      </Host>
    </Engine>
</Service>
```

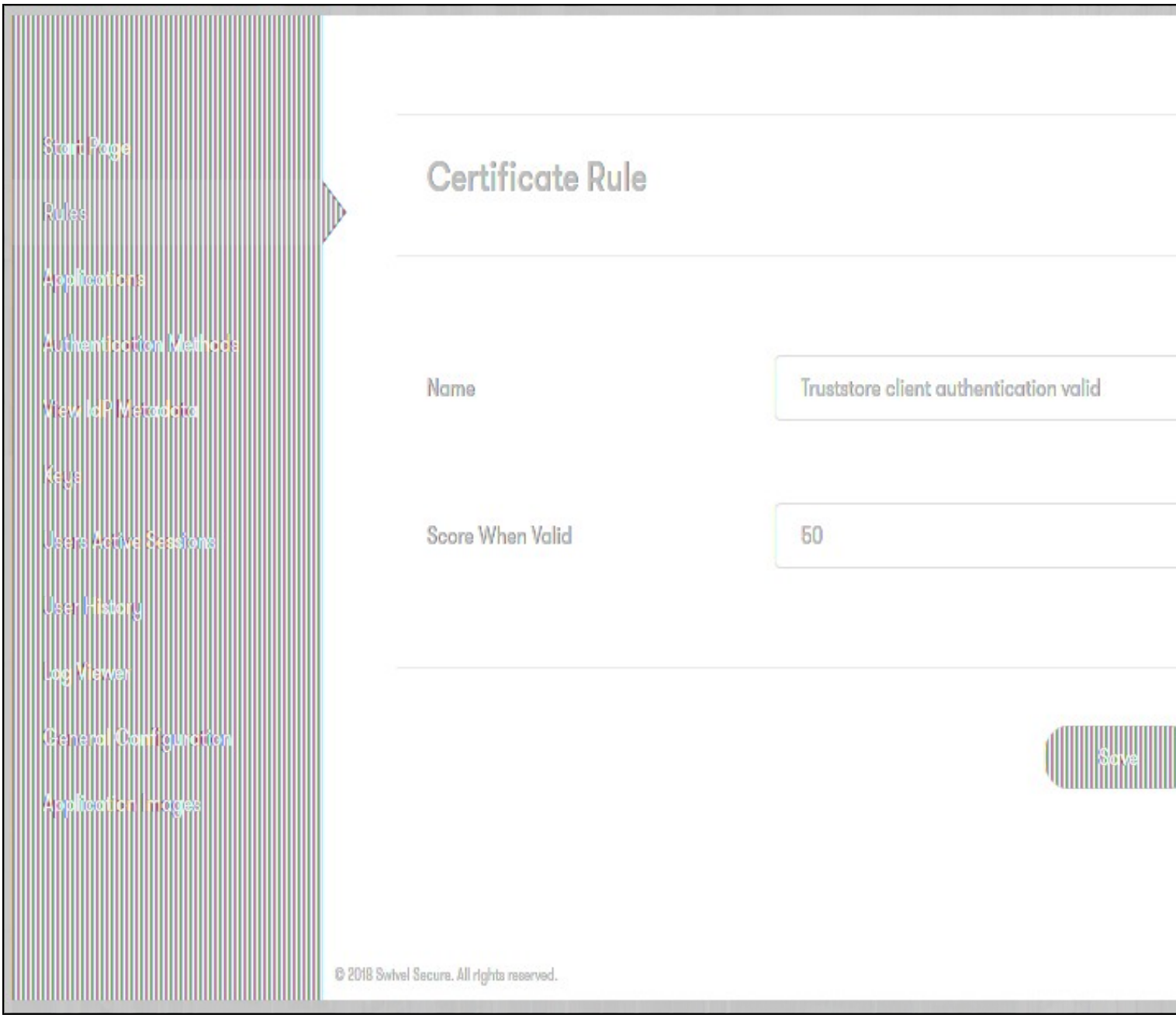
Define the points within AuthControl Sentry SSO

Login to the AuthControl Sentry SSO Administration Portal. Goto Rules. Against Certificate, click ?View Rules?:





Add a new Certificate Rule. You can give any arbitrary name and assign the points you wish to award a user if they present a valid client authentication certificate during authentication:



Establishing a Client-Side Certificate

- Ensure that your Client-Side certificate is PKCS#12 with a private key exported;
- The Client-Side certificate should be signed by your CA or Intermediate CA;
- Import this signed PKCS#12 certificate into the personal or computer trust store so that it is visible in the Settings -> Certificates panel of your Web Browser;

Enabling Certificate Revocation List checking

It's possible to enable CRL checks to establish a stronger chain of trust. Enabling this feature will enhance security by checking to see if an issued certificate has been revoked by the Certificate Authority.

Note: This feature relies upon an Outbound Internet connection and DNS to be configured. If not configured properly it may cause performance issues

Note: This feature relies upon restarting tomcat every time the CRL is updated

Create a job to download your CRL file(s)

```
#!/bin/bash
wget http://url1/crlfile1.crl -O /home/swivel/crl.crl
if [ $? -ne 0 ]; then
  exit
fi
openssl crl -inform DER -in /home/swivel/crl.crl -outform PEM -out /home/swivel/.swivel/crl.pem
rm /home/swivel/crl.crl
service tomcat restart
```

If you have more than one crl, you can concatenate them in one file:

```
#!/bin/bash
wget http://url1/crlfile1.crl -O /home/swivel/crla.crl
if [ $? -ne 0 ]; then
  exit
fi
wget http://url2/crlfile2.crl -O /home/swivel/crlb.crl
if [ $? -ne 0 ]; then
  exit
fi
openssl crl -inform DER -in /home/swivel/crla.crl -outform PEM -out /home/swivel/crla.pem
openssl crl -inform DER -in /home/swivel/crlb.crl -outform PEM -out /home/swivel/crlb.pem
cat /home/swivel/crla.pem /home/swivel/crlb.pem > /home/swivel/.swivel/crl.pem
rm /home/swivel/crla.crl
rm /home/swivel/crlb.crl
rm /home/swivel/crla.pem
rm /home/swivel/crlb.pem
service tomcat restart
```

save the script as /home/swivel/.swivel/getCRLs.sh and create a cron job to run the script in "/etc/crontab". This example runs every day at midnight:

```
0 0 * * * swivel /home/swivel/.swivel/getCRLs.sh
```

Modify the Apache Tomcat server.xml

On the Swivel Secure appliance, take a backup of the /usr/local/tomcat/conf/server.xml file, prior to making the necessary changes.

For the 8443 connector entry, add the following parameter:

```
crlFile="/home/swivel/.swivel/crl.pem"
```

So that it looks like this...

Before:

```
<Service name="webapps2">
<Connector SSLEnabled="true" acceptCount="100" address="0.0.0.0" ciphers="TLS_ECDHE_DSS_WITH_AES_128_GCM_SHA256,TLS_DHE_ECDSA_WITH_AES_128
    <Engine defaultHost="localhost" name="webapps2">
      <Host appBase="webapps2" autoDeploy="true" name="localhost" unpackWARs="true">
        <Valve className="org.apache.catalina.valves.AccessLogValve" directory="logs" pattern="common" prefix="webapps2" rotatable="false"
      </Host>
    </Engine>
</Service>
```

After:

```
<Service name="webapps2">
<Connector SSLEnabled="true" acceptCount="100" address="0.0.0.0" ciphers="TLS_ECDHE_DSS_WITH_AES_128_GCM_SHA256,TLS_DHE_ECDSA_WITH_AES_128
    <Engine defaultHost="localhost" name="webapps2">
      <Host appBase="webapps2" autoDeploy="true" name="localhost" unpackWARs="true">
        <Valve className="org.apache.catalina.valves.AccessLogValve" directory="logs" pattern="common" prefix="webapps2" rotatable="false"
      </Host>
    </Engine>
</Service>
```

Troubleshooting

- If the Client-Side certificate is signed by the Root CA, then ensure that the Root CA trusted cert is imported into the server-side trust store. Likewise, if it is signed by an Intermediate CA, ensure that both the Root CA trusted cert AND the Intermediate CA trusted cert are BOTH imported into the server-side trust store;
- An Apache Tomcat restart is required, for the Server-Side changes to the server.xml to take effect;
- If you've just imported the Client Authentication PKCS#12 certificate into your device then you may need to completely close and re-open your Web Browser for it to become available;
- Ensure that the truststore password in the server.xml matches what was set in the KeyStore Explorer application. Re-apply the password under the Tools -> Set Password option on the KeyStore Explorer application if necessary.
- Check file permissions on the truststore file, so that they match those permissions of the existing keystore file being used for secure HTTP.
- Further clues about successfully loading the truststore keystore during startup can be found in /var/log/tomcat/catalina.out;
- View the AuthControl Sentry SSO logs for client authentication troubleshooting once your Server-side setup is established;
- If performance issues are encountered with CRL checking enabled, check to ensure that your Outbound Internet connection or DNS is still in place. Try to telnet the CRL URL hostname using telnet in the CMI Tools menu, or on the command line.
- ERR_BAD_SSL_CLIENT_AUTH_CERT (chrome), SSL_ERROR_CERTIFICATE_UNKNOWN_ALERT (firefox) or "Cannot securely connect to this page" (IE) means that the Client certificate is not valid or that the CRL file is not valid