Supplemental Guide

October 2019

Custom SSL Certificate Installation for the SnapServer Web Server (GuardianOS 8.0 or Later)

Summary

The SnapServer web server uses a default, self-signed SSL certificate generated using the server name. A custom SSL certificate signed by a certificate authority can be installed instead to properly validate the server's identity and prevent browser warnings or other potential security concerns when connecting over HTTPS.

Prerequisites

The following must be satisfied before installing a custom SSL certificate to a SnapServer.

- An SSL certificate signed by an official certificate authority for the server or server's DNS domain must be independently issued by a certificate authority (CA). These are typically available for purchase from a global CA or generated by an internally-controlled CA in the local network environment.
- The SSL certificate must be issued for either a fully-qualified domain name (FQDN), for example, **`snap1234567.example.com`**, or a wildcard domain, for example, **`*.example.com`**.
  - If the certificate is issued for an FQDN, the certificate's hostname must match the server name of the SnapServer.

    For example, if the certificate is for **`snap123456.example.com`**, the server name must be **`snap123456`**.
  - If the certificate is issued for either an FQDN or wildcard domain, the certificate's DNS domain must match the server's DNS domain.

    For example, if the certificate is for **`*.example.com`**, the server's DNS domain must be **`example.com`**.

  NOTE: The server's DNS Domain Name is set by DHCP or by the first DNS Domain Name defined in the TCP/IP Port Properties page. The server's FQDN can be viewed with the "**`hostname -f`**" command in a Linux shell in an SSH session as described in the Installation Procedure.

- The x509 SSL certificate, the private key used to submit the certificate request, and any required intermediate certificates must be in text files in the PEM format. These may be provided by the issuer in various formats and must be converted to PEM before starting this procedure.

  NOTE: Certificate PEM files contain one or more sections starting with
  "**`-----BEGIN CERTIFICATE-----`**"
  and ending with
  "**`-----END CERTIFICATE-----`**".
  Private key PEM files contain one section starting with
  "**`-----BEGIN <encryption_type> PRIVATE KEY-----`**"
  and ending with
  "**`-----END <encryption_type> PRIVATE KEY-----`**".

- An SSH client supporting SSH v2 must be installed on a remote workstation.

Important Considerations

The following must be taken into consideration when installing a custom SSL certificate to a SnapServer:

- If the SSL certificate is based on the server's FQDN, changing the server name causes the server to revert to a default self-signed certificate. Wildcard certificates are recommended.
- Changing the server's DNS Domain Name causes the server to revert to a default self-signed certificate. This can occur from either TCP/IP configuration, change of Windows Domain membership status, or a change in the DNS Domain Name setting assigned by the DHCP server.
- Changing the HTTPS minimum protocol version (via the CLI "web set https-min-protocol" or "sslv3 set" commands) causes the server to revert to a default self-signed certificate.

- If connecting to the server over HTTPS using either an IP address, a short hostname (without domain), or any name that doesn't match the certificate, browsers will still warn about an untrusted SSL certificate after a custom SSL certificate is installed.

Installation Procedure

Complete the following procedure to install a custom SSL certificate to the SnapServer web server.

1. Copy the certificate and private key PEM files into a directory in a share on the SnapServer, for example, `\\snap123456\SHARE1\ssl`.
2. Connect to the server over SSH and login as "**admin**" to load the Snap CLI.
3. Enter "**osshell**" to launch a standard Linux shell.
4. Change to the root user by entering "**su -**" and providing the password for the "**admin**" user.
5. Change to the directory containing the PEM files.

   For example, `cd /shares/SHARE1/ssl`.
6. Confirm that the SnapServer FQDN (**snap123456.example.com**) is consistent with either the FQDN of the certificate or the domain of the certificate.

   Run "**hostname -f**" to obtain the server's FQDN.
7. If multiple certificates were provided by the issuer (for example, the server's certificate and intermediate chain certificates), combine the PEM files in to a single **sslcerts.pem** file using the "**cat**" command. In the following example of the command, all certificate PEM files are listed before the ">".

   Do **not** include the private key PEM file:

   ```
   cat x509cert.pem intermediatecert.pem > sslcerts.pem
   ```
8. Generate a pkcs12 keystore (**keystore.p12**) from the single certificate PEM file and the private key. As shown in the following command, substitute the name of your private key PEM file for <*private_key.pem*>:

   ```
   openssl pkcs12 -export -in sslcerts.pem -inkey <private_key.pem> -name \
   tomcat -out keystore.p12 -passout pass:changeit
   ```

   NOTE: If your issuer provided only a single certificate file and you skipped Step 7, substitute its filename for **sslcerts.pem** above.
9. Back up the SnapServer's keystore:

   ```
   cp /etc/.keystore /etc/.keystore.bak
   ```
10. Replace the SnapServer's default Java KeyStore with the new Java KeyStore:

    ```
    cp keystore.p12 /etc/.keystore
    ```
11. Restart the web server:

    ```
    /etc/init.d/tomcat restart
    ```

    Wait until the message **Preparing to initialize Log4j** appears.
12. In a web browser, connect to the server's fully-qualified domain name over HTTPS and confirm proper display of the SnapServer web interface without errors or browser warnings.

    For example, `https://snap123456.example.com`.

Troubleshooting

If an error or security warning was returned by the browser when connecting over HTTPS, confirm the following:

- All required intermediate certificates are available and in PEM format.
- The certificate is still valid and has not expired.
- The private key file matches the key used when submitting the CRL to issue the SSL certificate.
- The HTTPS connection is made by FQDN and not IP address or hostname.
- The FQDN used to connect to the server matches the FQDN, domain, and hostname (if not wildcard) for which the certificate was issued.

To return to a default self-signed certificate:

1. Connect to the server over SSH, load a standard Linux shell, change to root, and change to the directory containing the certificates as in the procedure above.
2. Run the following to delete the server's new Java KeyStore:

   ```
   rm /etc/.keystore
   ```

3. Restart the web server:

   ```
   /etc/init.d/tomcat restart
   ```