Splunk4Admins - Encryption of Data in Transit (DIT)

Lab Guide

Overview

This lab guide contains the hands-on exercises for the Splunk4Admins - Security & Encryption workshop. Before proceeding with these exercises, please ensure that you have a copy of the workshop slide deck, which will help to put into context the tasks you are carrying out.

Download the workshop slide deck: https://splk.it/S4A-DIT-Attendee

Note: Copy and Paste issues may occur when selecting single-line commands that are spread across multiple lines within this document.

Prerequisites

In order to complete these exercises, you will need your own Splunk instance. Splunk's hands-on workshops are delivered via the Splunk Show portal and you will need a splunk.com account in order to access this.

If you don't already have a Splunk.com account, please create one here before proceeding with the rest of the workshop.

⚠️ Troubleshooting Connectivity

If you experience connectivity issues with accessing either your workshop environment or the event page, please try the following troubleshooting steps. If you still experience issues please reach out to the team running your workshop.

- **Use Google Chrome** (if you're not already)
- If the event page (i.e. *https://show.splunk.com/event/<eventID>*) didn't load when you clicked on the link, try **refreshing the page**
- **Disconnect from VPN** (if you're using one)
- **Clear your browser cache and restart your browser** (if using Google Chrome, go to: Settings > Privacy and security > Clear browsing data)
- **Try using private browsing mode** (e.g. Incognito in Google Chrome) to rule out any cache issues
- **Try using another computer** such as your personal computer - all you need is a web browser! Cloud platforms like AWS can often be blocked on corporate laptops.

Table of Contents

Exercise 1 – Install Splunk Securely on Linux

Description

This lab guide provides step-by-step instructions on how to install Splunk Enterprise securely on your system. These guidelines will help ensure your Splunk installation is secure and adheres to best practices.

Prerequisites

- Access to a server or virtual machine where you will install Splunk.
- Root or administrative privileges on the server.
- Internet access to download Splunk installation files.
- Basic knowledge of Linux commands (if installing on a Linux system).

Steps

Download the Splunk Installation Package

1. Navigate to the Splunk download page.
2. Select the appropriate version and platform for your system.
3. Download the installation package to your server.

Verify the Integrity of the Downloaded Package

4. Obtain the checksum provided on the Splunk download page.
5. Use the following command to generate a checksum of the downloaded package and compare it with the provided checksum:

MD5	SHA512
`cat splunk-xxxx-release.tgz.md5`	`cat splunk-xxxx-release.tgz.sha512`

MD5	SHA512
`md5sum splunk-xxxx-release.tgz`	`sha512sum splunk-xxxx-release.tgz`

6. Compare the MD5 or SHA512 hash file output against the result from the md5sum or sha512sum utilities.
7. If the hashes match exactly, then the package you downloaded is authentic, and you can continue with the installation. If the hash does not match, try downloading the package again, as it is incomplete or has possibly been modified.

Install Splunk

8. After verifying that your installation package is authentic, you can proceed with installation. Please see the Installation instructions in the Installation Manual, for the purposes of this workshop, use the `/opt/showlab/s4a-s-aio` directory to install Splunk. **DO NOT START SPLUNK**.

```
tar -xf $your_download_location/splunk-xxxx-release.tgz -C /opt/showlab/s4a-s-aio/
```

NOTE: The Splunk Enterprise installer needs action from you to create administrator credentials. You must do one of the following:

a. Provide credentials as command-line arguments to the installer when you run the installer
b. Supply the password in a configuration file that the installer can read during the installation process
c. Answer the prompts during the installation process when they appear

If you do not create the password during the installation process using one of these methods, it's possible to end up with a temporarily unusable instance. This can happen, for example, if you use the --no-prompt Splunk CLI argument to start a Splunk Enterprise installation and do not provide an administrator password in the user-seed.conf configuration file inside the installation. In this case, the installer doesn't prompt you to create an administrator account. Since you did not specify a password, the installer succeeded in installing the software but did not create the administrator credentials.

In this case, you must create the administrator credentials manually for the instance to be accessible again.

If you installed Splunk Enterprise and did not create the administrator credentials, you can use one of the following methods to do so. These methods require physical access to the machine that runs the instance.

Create Secure Admin Credentials

9. Create the administrator credentials using the user-seed.conf configuration file.
   NOTE: This is currently the most secure method to create administrative credentials. Other methods can introduce security risks, mainly around access to command line history or process output.

10. Create the `/opt/showlab/s4a-s-aio/splunk/etc/system/local/user-seed.conf` file as follows:

```
[user_info]
USERNAME = admin
PASSWORD = <your password>
```

11. Start Splunk Enterprise.

```
/opt/showlab/s4a-s-aio/splunk/bin/splunk start --accept-license --answer-yes
```

> **Note**: When accessing the web URL after installing Splunk Enterprise, the web IP address listed in the terminal window after starting Splunk differs from the external AWS IP address exposed to the internet.

Exercise 2 – Obtain Certificates

Description

This exercise will teach you to secure your Splunk platform deployment. First, decide the level of security required for your environment, considering options for securing communications between Splunk Web, your browser, and individual Splunk platform instances. Obtain the necessary TLS certificates from third-party providers or by creating them yourself. Then, install and configure these certificates on each Splunk platform instance. Finally, test and troubleshoot the setup to ensure secure communications are properly established.

Steps

Decide how you want to secure your Splunk platform deployment.

For this lab, we have decided that you will secure the web port, splunk2splunk ports, and management ports.

Obtain the TLS certificates you need to secure the deployment.

Splunk's best practice is to get third-party certificates. However, for the purposes of this lab, we will be using self-signed certificates.

1. Create the root certificate authority certificate
    a. Open a command line interface, such as a shell prompt, Terminal, or PowerShell window.
    b. Connect to the Splunk platform instance where you want to generate the certificate signing request (CSR).
    c. Create a new directory within the Splunk platform instance installation for the certificates.

```
mkdir /opt/showlab/s4a-s-aio/splunk/etc/auth/mycerts
cd /opt/showlab/s4a-s-aio/splunk/etc/auth/mycerts
```

    d. Create a private key for your root certificate authority certificate.

```
/opt/showlab/s4a-s-aio/splunk/bin/splunk cmd openssl genrsa -aes256 -out
myCertAuthPrivateKey.key 2048
```

e. When the OpenSSL program prompts you, enter a password for the key. The OpenSSL program then creates a file `myCertAuthPrivateKey.key`.

f. Use the private key `myCertAuthPrivateKey.key` to generate a CSR for your certificate:

```
/opt/showlab/s4a-s-aio/splunk/bin/splunk cmd openssl req -new -key
myCertAuthPrivateKey.key -out myCertAuthCertificate.csr
```

g. When the OpenSSL program prompts you, enter the password you created for your private key `myCertAuthPrivateKey.key`.

h. The OpenSSL program asks for several different fields. At a minimum, provide values for the following:

   i. Country Name

   ii. State or Province Name (full name)

   iii. Locality Name (eg, city)

   iv. Organization Name (eg, company)

   v. Organizational Unit Name (eg, section)

   vi. Common Name (e.g. server FQDN or YOUR name)

   vii. Email Address

   After you enter the information, the OpenSSL program creates a new CSR file called `myCertAuthCertificate.csr`.

i. To create the root certificate authority certificate, use the CSR file you created and sign it with the private key you created previously.

```
/opt/showlab/s4a-s-aio/splunk/bin/splunk cmd openssl x509 -req -in
myCertAuthCertificate.csr -sha512 -signkey myCertAuthPrivateKey.key -CAcreateserial -out
myCertAuthCertificate.pem -days 1095
```

j. When the OpenSSL program prompts you, enter the password you created for your private key `myCertAuthPrivateKey.key` again. The OpenSSL program creates the root certificate authority certificate file called `myCertAuthCertificate.pem`.

k. Verify pem file

```
openssl x509 -in myCertAuthCertificate.pem -text -noout
```

```
Certificate:
    Data:
        Version: 1 (0x0)
        Serial Number:
            a1:5f:e7:4b:6c:e2:ef:5c
        Signature Algorithm: sha512WithRSAEncryption
        Issuer: C = US, ST = FL, L = Sarasota, O = Splunk, OU = TS&I, CN = S4A-FM, emailAddress = noreply@splunk.com
        Validity
            Not Before: Jul 31 19:03:31 2024 GMT
            Not After : Jul 31 19:03:31 2027 GMT
        Subject: C = US, ST = FL, L = Sarasota, O = Splunk, OU = TS&I, CN = S4A-FM, emailAddress = noreply@splunk.com
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
                Public-Key: (2048 bit)
                Modulus:
```

Create server certificates and sign them with the root certificate authority certificate

2. Create a private key for the server certificate.

```
/opt/showlab/s4a-s-aio/splunk/bin/splunk cmd openssl genrsa -aes256 -out
myServerPrivateKey.key 2048
```

3. When the OpenSSL program prompts you, enter a password for the key. The OpenSSL program then creates a file `myServerPrivateKey.key`.

   NOTE: Do not use the same password you used to sign the private key for the certificate authority certificate.

4. Use the private key myServerPrivateKey.key to generate a CSR for your certificate:

```
/opt/showlab/s4a-s-aio/splunk/bin/splunk cmd openssl req -new -key myServerPrivateKey.key
-out myServerCertificate.csr
```

5. When the OpenSSL program prompts you, enter the password you created for the private key `myServerPrivateKey.key`.
6. Provide the requested information for your certificate. The OpenSSL program creates a new CSR file called `myServerCertificate.csr`.
7. Use the CSR file you created and sign it with the private key you created previously, the certificate authority certificate, and its private key to create the server certificate.

```
/opt/showlab/s4a-s-aio/splunk/bin/splunk cmd openssl x509 -req -in
myServerCertificate.csr -SHA256 -CA myCertAuthCertificate.pem -CAkey
myCertAuthPrivateKey.key -CAcreateserial -out myServerCertificate.pem -days 1095
```

8. When the OpenSSL program prompts you, enter the password you created for the private key `myCertAuthPrivateKey.key` again. The OpenSSL program creates the server certificate file called `myServerCertificate.pem`.

Exercise 3 – Prepare TLS certificates for use with the Splunk platform

Description

In this exercise, you will learn to create a single combined certificate file to secure end-to-end communication between Splunk Enterprise components. After obtaining TLS certificates, you will prepare them for deployment by combining them with the respective private keys into a single certificate file. This combined certificate file is essential for the Splunk platform to function securely. Every certificate and key used in your deployment must be concatenated this way, as individual certificates or keys will not work even with the correct configuration. This lab will guide you through the process, ensuring that all your Splunk services and contact points utilize properly combined certificate files.

Steps

Create a single combined certificate file

```
cat myServerCertificate.pem myServerPrivateKey.key myCertAuthCertificate.pem > myCombinedCertificate.pem
```

After you create the combined certificate file, review it using a text editor. Its contents must contain, in the following order:

- The server certificate
- The private key
- The certificate authority certificate

Exercise 4 – Configuring Splunk indexing and forwarding using TLS certificates

Description

In this lab exercise, you will learn how to configure Splunk indexing and forwarding to use TLS certificates for secure communication. This process involves setting up TLS certificates to ensure data integrity and confidentiality between Splunk forwarders and indexers. You will leverage your previously created TLS certificates, configure Splunk instances to use these certificates, and verify secure data transmission across your Splunk deployment. By the end of this lab, you will be able to establish a secure indexing and forwarding infrastructure that protects your data from unauthorized access and tampering.

Steps

1. Within the apps directory of the all-in-one instance, create a new base app named `all_indexer_base`

```
cd /opt/showlab/s4a-s-aio/splunk/etc/apps
mkdir all_indexer_base
```

2. Within the `all_indexer_base` directory, create the metadata and local directories:

```
cd /opt/showlab/s4a-s-aio/splunk/etc/apps/all_indexer_base
mkdir metadata
mkdir local
```

3. Within the metadata directory, create a new file named `local.meta` with the following configuration:

```
cd metadata
vi local.meta
```

Press i to enter Insert mode.

```
[]
access = read : [ * ], write : [ admin ]
export = system
```

4. Save and close the file. Press Esc to exit Insert mode, then type :wq and press Enter to save and close the file.
5. Within the local directory, create a new file named `inputs.conf` with the following configuration:

```
cd ../local
vi inputs.conf
```

Press i to enter Insert mode.

```
# BASE SETTINGS
[splunktcp-ssl://9997]

# SSL SETTINGS
[SSL]
serverCert = /opt/showlab/s4a-s-aio/splunk/etc/auth/mycerts/myCombinedCertificate.pem
sslPassword = <myServerCertificatePassword>
# requireClientCert = true
sslVersions = *,-ssl2
# sslCommonNameToCheck = aio.mydomain.com
```

6. Make sure to update <myServerCertificatePassword> with your password.
    a. Save and close the file.
7. Within the local directory, create a new file named `server.conf` with the following configuration to enable SplunkdSSL in addition to splunktcp SSL:

```
vi server.conf
```

Press i to enter Insert mode.

```
[sslConfig]
sslRootCAPath = /opt/showlab/s4a-s-aio/splunk/etc/auth/mycerts/myCertAuthCertificate.pem
```

8. Save and close the file.
9. Restart Splunkd.

```
/opt/showlab/s4a-s-aio/splunk/bin/splunk restart splunkd
```

10. For this workshop, we are going to reuse the server certificate as the client certificate of the universal forwarder. First, we need to create the new certificate directory, though.

```
cd /opt/showlab/s4a-s-uf/splunkforwarder/etc/auth
mkdir mycerts
```

11. We need to copy the server and root certificates from the Splunk Enterprise installation to the universal forwarder installation.

```
cp /opt/showlab/s4a-s-aio/splunk/etc/auth/mycerts/myCertAuthCertificate.pem
/opt/showlab/s4a-s-uf/splunkforwarder/etc/auth/mycerts/
#--
cp /opt/showlab/s4a-s-aio/splunk/etc/auth/mycerts/myCombinedCertificate.pem
/opt/showlab/s4a-s-uf/splunkforwarder/etc/auth/mycerts/
#--
cd mycerts
mv myCombinedCertificate.pem myClientCertificate.pem
```

12. Navigate to the apps directory on the universal forwarder install under the showlab directory

```
cd /opt/showlab/s4a-s-uf/splunkforwarder/etc/apps
```

13. Create a new folder within the deployment-apps directory named `all_fowarder_outputs`:

```
mkdir all_forwarder_outputs
```

14. Within the `all_forwarder_outputs` directory, create the metadata and local directories:

```
cd all_forwarder_outputs
mkdir metadata
mkdir local
```

15. Within the metadata directory, create a new file named `local.meta` with the following configuration, replacing primary indexers with the IP address of the two indexers you have been assigned:

```
cd metadata
vi local.meta
```

Press i to enter Insert mode.

```
[]
access = read : [ * ], write : [ admin ]
export = system
```

16. Save and close the file. Press Esc to exit Insert mode, then type :wq and press Enter to save and close the file.
17. Within the local directory, create a new file named `outputs.conf` with the following configuration:

```
cd ../local
vi outputs.conf
```

Press i to enter Insert mode.

```
# BASE SETTINGS

[tcpout]
defaultGroup = primary_indexer
forceTimebasedAutoLB = true
forwardedindex.0.whitelist = (_audit|_introspection|_internal)

[tcpout:primary_indexer]
server = localhost:9997
disabled = 0
clientCert =
/opt/showlab/s4a-s-uf/splunkforwarder/etc/auth/mycerts/myClientCertificate.pem
useClientSSLCompression = true
sslPassword = <myServerCertificatePassword>
# sslCommonNameToCheck = aio.mycompany.com
# sslVerifyServerCert = true
```

18. Update the value for <myServerCertificatePassword>
    a. Save and close the file.
19. Within the local directory, create a new file named `server.conf` with the following configuration:

```
vi server.conf
```

Press i to enter Insert mode.

```
[sslConfig]
sslRootCAPath =
/opt/showlab/s4a-s-uf/splunkforwarder/etc/auth/mycerts/myCertAuthCertificate.pem
```

20. Save and close the file.
21. Restart the splunk forwarder.

```
/opt/showlab/s4a-s-uf/splunkforwarder/bin/splunk restart
```

Note: You may need to run a chown on the splunkfowarder directory before you can restart the splunk forwarder (chown -R splunk: /opt/showlab/s4a-s-uf/splunkforwarder).

22. Confirm the configurations are working properly. If you are having any issues with your deployment, please check out the Troubleshooting TLS Connections docs page.

Exercise 5 – Configuring Splunk Web to use TLS

Description

In this lab exercise, you will learn how to secure Splunk Web by configuring it to use TLS certificates. This involves generating a self-signed root certificate authority (CA) certificate, creating and signing a server certificate, and configuring Splunk Web to use these certificates for secure communication. By the end of this lab, you will ensure that all web traffic between Splunk Web and your browser is encrypted, enhancing the security of your Splunk deployment.

Steps

The great news is that we already have everything we need. In this lab, we will make a copy of the certificates we have already created and update their names.

1. Let's get started by going to the `mycerts` directory of the Splunk Enterprise installation:

```
cd /opt/showlab/s4a-s-aio/splunk/etc/auth/mycerts/
```

2. Now, we will copy the `myServerPrivateKey.key` and name it `mySplunkWebPrivateKey.key`, removing the passphrase.

```
cp myServerPrivateKey.key mySplunkWebPrivateKey.key

openssl rsa -in mySplunkWebPrivateKey.key -out mySplunkWebPrivateKey.key
```

3. Next, we will do the same without removing the passphrase for `myCombinedCertificate.pem`, renaming it to `mySplunkWebCertificate.pem`

```
cp myCombinedCertificate.pem mySplunkWebCertificate.pem
```

4. Now, we need to update the `web.conf` to enable Web SSL. Let's do this via a base app under the apps directory

```
cd /opt/showlab/s4a-s-aio/splunk/etc/apps
```

5. Create a new folder within the `apps` directory named `all_search_base`:

```
mkdir all_search_base
```

6. Within the `all_search_base` directory, create the `metadata` and `local` directories:

```
cd all_search_base
mkdir metadata
mkdir local
```

7. Within the `metadata` directory, create a new file named `local.meta` with the following configuration:

```
cd metadata
vi local.meta
```

Press i to enter Insert mode.

```
[]
access = read : [ * ], write : [ admin ]
export = system
```

8. Save and close the file. Press Esc to exit Insert mode, then type :wq and press Enter to save and close the file.
9. Within the local directory, create a new file named `web.conf` with the following configuration:

```
cd ../local
vi web.conf
```

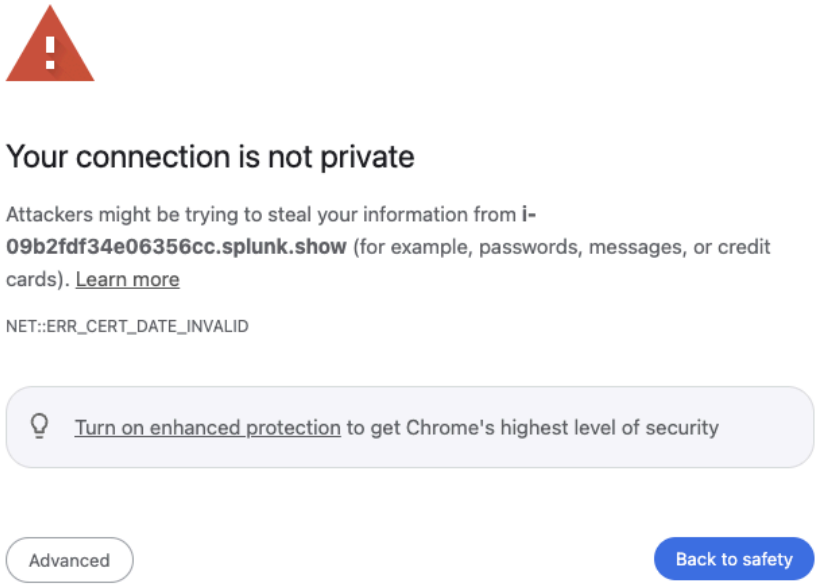
Press i to enter Insert mode.

```
[settings]
httpport = 8443
enableSplunkWebSSL = true
privKeyPath = /opt/showlab/s4a-s-aio/splunk/etc/auth/mycerts/mySplunkWebPrivateKey.key
serverCert = /opt/showlab/s4a-s-aio/splunk/etc/auth/mycerts/mySplunkWebCertificate.pem
```

10. Save and close the file.
11. Restart Splunk Enterprise.

```
/opt/showlab/s4a-s-aio/splunk/bin/splunk restart
```

12. Confirm the configurations are working properly. If you are having any issues with your deployment, please check out the Troubleshooting TLS Connections docs page.



Note: You will have to update your URL to be HTTPS and change the port from 8000 to 8443.

Exercise 6 – Configuring Splunkd to use TLS

Description

In this lab exercise, you will learn how to secure Splunkd by configuring it to use SSL certificates. This involves generating a self-signed root CA certificate, creating and signing a server certificate, and configuring Splunkd to use these certificates for secure communication. By the end of this lab, you will ensure that all communication between Splunk instances is encrypted, enhancing the security of your Splunk deployment.

Steps

1. We are going to continue using what we already created, in the real world you would either recreate the certificates or request them via your authority.
2. Navigate to the local directory within the `all_indexer_base` directory

```
cd /opt/showlab/s4a-s-aio/splunk/etc/apps/all_indexer_base/local
```

3. Update the server.conf to look like this, ensuring you update <myServerCertificatePassword> with the correct value

```
[sslConfig]
enableSplunkdSSL = true
sslRootCAPath = /opt/showlab/s4a-s-aio/splunk/etc/auth/mycerts/myCertAuthCertificate.pem
serverCert = /opt/showlab/s4a-s-aio/splunk/etc/auth/mycerts/myCombinedCertificate.pem
sslPassword = <myServerCertificatePassword>
# requireClientCert = true
sslVersions = *,-ssl2
# sslCommonNameToCheck = aio.mydomain.com
```

4. Navigate to the system/local directory of the full instance and comment out the sslConfig stana and sslPassword in the server.conf

```
vi /opt/showlab/s4a-s-aio/splunk/etc/system/local/server.conf
```

```
#[sslConfig]
#sslPassword = $7$oE9ikOJjA9cnEaOfw+NaLbzufl3TqLyigBHfwWYpYgm8hhziWqx9fg==
```

5. Restart Splunk

```
/opt/showlab/s4a-s-aio/splunk/bin/splunk restart
```

6. Confirm the configurations are working properly. If you are having any issues with your deployment, please check out the Troubleshooting TLS Connections docs page.
7. Now we are going to enable this host as a deployment server. Start by moving the all_deploymentclient app on the universal forwarder into the deployment apps directory on the full Splunk installation. This will enable the forwarder management console on the UI.

```
cp -r /opt/showlab/s4a-s-uf/splunkforwarder/etc/apps/all_deploymentclient/
/opt/showlab/s4a-s-aio/splunk/etc/deployment-apps/
```

8. Ensure the mycerts directory on the universal forwarder exists and ensure you have copied over myCertAuthCertificate.pem and myCombineCertificate.pem
9. Navigate to the all_deploymentclient app on the universal forwarder.

```
cd /opt/showlab/s4a-s-uf/splunkforwarder/etc/apps/all_deploymentclient/local
```

10. Create a server.conf file within this directory with the following details

```
[sslConfig]
enableSplunkdSSL = true
sslRootCAPath =
/opt/showlab/s4a-s-uf/splunkforwarder/etc/auth/mycerts/myCertAuthCertificate.pem
serverCert =
/opt/showlab/s4a-s-uf/splunkforwarder/etc/auth/mycerts/myClientCertificate.pem
sslPassword = myServerCertificatePassword
# requireClientCert = true
sslVersions = *,-ssl2
# sslCommonNameToCheck = aio.mydomain.com
```

11. Navigate to the system/local directory of the universal forwarder and comment out the sslConfig stanza and sslPassword in the server.conf

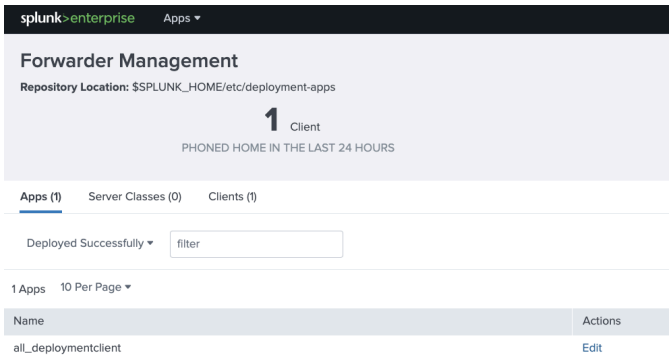
```
vi /opt/showlab/s4a-s-uf/splunkforwarder/etc/system/local/server.conf
```

```
#[sslConfig]
#sslPassword = $7$oE9ikOJjA9cnEaOfw+NaLbzufl3TqLyigBHfwWYpYgm8hhziWqx9fg==
```

12. Restart the Splunk UF

```
/opt/showlab/s4a-s-uf/splunkforwarder/bin/splunk restart
```

13. Navigate to the web browser of the newly formed Deployment Server and validate the universal forwarder has established a connection to the deployment server.



14. Confirm the configurations are working properly. If you are having any issues with your deployment, please check out the Troubleshooting TLS Connections docs page.

Certificates and Configs Table

Service	Default Port(s)	Certificate(s) Needed	Configuration File	Configuration Stanza
Splunk Web	8000	web_privkey.key web_cert.pem cacert.pem	web.conf	[settings] httpport = enableSplunkWebSSL = true privKeyPath = serverCert =
Splunkd	8089	cacert.pem + server.pem + privkey.key ->
CombinedCert.pem	server.conf	[sslConfig] enableSplunkdSSL = true sslRootCAPath = serverCert = sslPassword = sslVersions =
Replication	8191 (SHC) 9887 (IDXC)	cacert.pem server.pem	server.conf	[replication_port-ssl://9887] rootCA = serverCert = sslPassword =
KV Store	8191	cacert.pem server.pem	server.conf	[kvstore] caCertFile = serverCert = sslPassword =
HTTP Event	8088	server.pem	inputs.conf	[http]

Collector (HEC)		privkey.key cacert.pem		enableSSL = true serverCert = privKeyPath = caCertFile =
Forwarding (Forwarder)	9997	server.pem	outputs.conf	[tcpout:<output_group>] server = disabled = 0 clientCert = useClientSSLCompression = true sslPassword =
Forwarding (Forwarder & Data Receivers)	9997	cacert.pem	server.conf	[sslConfig] sslRootCAPath =
Forwarding (Data Receivers)	9997	cacert.pem + server.pem + privkey.key -> CombinedCert.pem	inputs.conf	[splunktcp-ssl://9997] [SSL] serverCert = sslPassword =