Self Signed Certificate Creation

Author: Chris Morales

Summary: This guide should only be used for **testing purposes only!** It's made to showcase the process of getting the required materials to secure your services with a certificate that you obtain.

Resources used: Self Signed Cert Creation Guide

Create a Root CA

Note: Because this shouldn't be used too often, I'll give you just the commands for this. They should run with no problems under a Linux machine. In this case, I tested it under an Ubuntu-based machine.

1. Create a directory named *openssl* to store everything in.

```
    mkdir openssl
    cd openssl
```

2. Create a **rootCA.key** and **rootCA.crt** for the root CA authority.

```
    openssl req -x509 -sha256 -days 356 -nodes -newkey rsa:2048 -subj "/CN=ccdc-jenkins-ca.com/C=US" -keyout rootCA.key -out rootCA.crt
```

This will be the CA that will sign our SSL cert.

Create Self-signed Certs

Information:

- *Server name: 192.168.0.94*

1. Create the Server Private Key

```
    openssl genrsa -out server.key 2048
```

You now have *server.key*.

2. Create a CSR *configuration* file named *csr.conf*.

```
[ req ]
default_bits = 2048
prompt = no
default_md = sha256
```

```
req_extensions = req_ext
distinguished_name = dn

[ dn ]
C = US
CN = 192.168.0.94
```

You now have *csr.conf*.

3. Create the *actual CSR* using the private key.

```
    openssl req -new -key server.key -out server.csr -config csr.conf
```

Now, you have *server.csr*.

In your directory, you should have *csr.conf, server.csr* and *server.key*.

4. Create an *external* file to be used for the SSL certificate and name it *cert.conf*.

```
authorityKeyIdentifier=keyid,issuer
basicConstraints=CA:FALSE
keyUsage = digitalSignature, nonRepudiation, keyEncipherment, dataEncipherment
subjectAltName = @alt_names

[alt_names]
DNS.1 = 192.168.0.94
```

You now have *cert.conf*.

5. Create the SSL cert with the self signed CA.

```
    openssl x509 -req -in server.csr -CA rootCA.crt -CAkey rootCA.key -
CAcreateserial -out server.crt -days 365 -sha256 -extfile cert.conf
```

You now have *server.crt*. This will be used in tandem with *server.key* to **enable SSL in applications.**

Reminder: This should only be used for testing how to actually secure something. Self-Signed certs are problematic as they provide little security. If it's internal, we typically want to get a certificate from some type of PKI infrastructure from within the company. So, use this as a final resort or during testing.