Using Easy-RSA to generate certificates and keys X.509

Yuriy Andamasov - 2021-09-21 - 2 Comments - in OpenVPN

Introduction

Several methods of setting up a Virtual Private Network require (or prefer) the use of X.509 certificates and keys for authentication, over the less secure shared secret method. Ideally, your organization should have an existing Public Key Infrastructure (PKI) which should be used with your specific settings. However, you don't have to have a complicated setup to utilize X.509 for authentication on VyOS. In fact, all of the tools you need to set up basic X.509 authentication come with VyOS already!

This article will cover a very basic setup that is referenced in other articles and can be tailored to your specific use case. We will go over using OpenVPN's built-in Easy-RSA scripts that come installed on VyOS by default. This article will not cover how X.509 PKIs work, or how to set up a more secure/better managed PKI, only how to set up a very basic PKI for use with OpenVPN and VyOS specifically.

Using OpenVPN's Easy-RSA:

OpenVPN ships with a set of scripts called Easy-RSA that can generate the appropriate files needed for an OpenVPN setup using X.509 certificates. The scripts can be a little obtuse at times to configure and use, however, Easy-RSA comes installed by default on VyOS routers (as it comes with OpenVPN itself), making it fairly standard across all installations.

Setting up Easy-RSA

Firstly, we need to copy the Easy-RSA scripts to a new directory so we can modify the values. We'll be copying it to `/config/my-easy-rsa-config`, so from the terminal in operational mode, run the following shell command (VyOS 1.2.x only):

```
cp -r /usr/share/easy-rsa/ /config/my-easy-rsa-config
cd /config/my-easy-rsa-config
```

This should copy the scripts from the example directory into one where we can change the values, and then changes the current directory to our new location.

VyOS 1.1.x Note: On VyOS version 1.1.x, the Easy-RSA scripts are found in `/usr/share/doc/openvpn/examples/easy-rsa/`. Inside 'easy-rsa' folder, use 2.0

directory for the latest script files. The configuration and use of Easy-RSA should be the same between 1.1.8 and 1.2.x.

```
cp -r /usr/share/doc/openvpn/examples/easy-rsa/ /config/my-easy-rsa-config
cd /config/my-easy-rsa-config/2.0
```

Next we can edit the `vars` file, which contains the default values for the keys (such as the organizational settings). If unchanged from the default, these will be prompted at runtime, so editing the file isn't explicitly needed, but it saves on having to specify things like the organization name, city name, etc. repeatedly, and cuts down on potential mistakes that may come from re-entering the same data repeatedly.

In the `vars` file, the values we specifically want to change are at the bottom, so open `vars` in your preferred editor (VyOS comes with the `nano` and `vi` editors by default; if you're unsure of which to use, you probably want `nano`.) and navigate to the bottom of the file. There, you should find the following text:

```
# These are the default values for fields
# which will be placed in the certificate.
# Don't leave any of these fields blank.
export KEY_ALTNAMES="something"
export KEY_COUNTRY="US"
export KEY_PROVINCE="CA"
export KEY_CITY="SanFrancisco"
export KEY_ORG="Fort-Funston"
export KEY_EMAIL="me@myhost.mydomain"
export KEY_EMAIL=mail@host.domain
export KEY_CN=changeme
export KEY_NAME=changeme
export KEY_OU=changeme
export PKCS11_MODULE_PATH=changeme
export PKCS11_PIN=1234
```

The values prefixed by `KEY_` are the most relevant for us in our case. Many of them should be fairly obvious, though of particular note are the `KEY_CN`, `KEY_NAME`, `KEY_OU`, and the `PKCS11_` values.

`KEY_CN` and `KEY_NAME` refer to the common name field and the name of the certificate, respectively. Because we will be generating client *and* server certificates (and possibly multiple client certificates), we probably want to leave these fields blank and specify them when we run the generation scripts. The name value is less important than the CN value, and the CN value can be set from the command line (more on that below)

KEY_OU is referring to an Organizational Unit and can be set to whatever if there's no need for it.

The PKCS11_ values refer to values used for Hardware Security Modules and smart cards, and can be safely ignored if that is not in your use case.

You may also change other values in the file at your discretion/need, though for most cases the defaults should be just fine.

Generating the Files

After you've edited the variables file, you'll need to load the variables into your current shell's environment. This is done with the following command:

```
vyos@vyos:/config/my-easy-rsa-config$ source ./vars
```

Next we'll want to clean things up to start completely fresh:

```
vyos@vyos:/config/my-easy-rsa-config$ ./clean-all
```

And now to generate the actual certificates, keys, and related files. In our case, we'll be generating two client keys (`branch1` and `branch2`) and a server key (`central`). These are the names and variables used in the Basic OpenVPN Client-Server example article, and shows how to generate two different client keys.

```
vyos@vyos:/config/my-easy-rsa-config$ ./build-ca
vyos@vyos:/config/my-easy-rsa-config$ ./build-dh
vyos@vyos:/config/my-easy-rsa-config$ ./build-key-server central
vyos@vyos:/config/my-easy-rsa-config$ ./build-key branch1
vyos@vyos:/config/my-easy-rsa-config$ ./build-key branch2
```

This sets up our Certificate Authority (CA), the key exchange files, our server key, and two client keys. The `build-key` commands may prompt for additional input, and will ask for confirmation to sign and commit the requests. You may also specify a password for each key, but unless you know you need that and know what to do with it, most people should probably leave it blank.

Installing the Files

Every host that needs these keys will need to have some particular files on it. In the other articles that rely on X.509 certificates, we use the directory `/config/auth/ovpn/`, so this is where we will place the files. The files that Easy-RSA generates are found in the `keys` subdirectory of where we copied it to in the first place (so, `/config/my-easy-rsa-config/keys` in our case here.)

On the server, we need to copy the following files:

```
ca.crt
dh1024.pem
```

```
<server-key-file>.key
<cerver-crt-file>.crt
```

so for instance:

```
vyos@vyos:/config/my-easy-rsa-config$ sudo mkdir /config/auth/ovpn
vyos@vyos:/config/my-easy-rsa-config$ sudo cp keys/ca.crt
/config/auth/ovpn
vyos@vyos:/config/my-easy-rsa-config$ sudo cp keys/dh2048.pem
/config/auth/ovpn
vyos@vyos:/config/my-easy-rsa-config$ sudo cp keys/central.key
/config/auth/ovpn
vyos@vyos:/config/my-easy-rsa-config$ sudo cp keys/central.crt
/config/auth/ovpn
```

Additionally, each client needs a copy of `ca.crt` and its own client key and cert files. The files are plaintext so they may be copied either manually, or through a remote file transfer tool like `scp`. Whichever method you use, the files need to end up in the proper location on each router. For example, Branch 1's router might have the following files:

```
vyos@branch1-rtr:$ ls /config/auth/ovpn
ca.crt  branch1.crt  branch1.key
```

You cannot dodge the issue of having to copy client keys to the clients from wherever you generated the keys by generating new keys on the client, as the keys generated must be signed against the same CA in order for authentication to function properly.

Comments (2)

Kirill Tue, 5th Feb 2019 07:36

Since, at least, vyos-1.2.0-epa2 version, easy-RSA scripts location has changed to /usr/share/easy-rsa/

Yuriy Andamasov Tue, 5th Feb 2019 12:28

will update the article, thanks for heads up