OpenVPN のサーバを立てる

KusaReMKN

要約

OpenVPN のサーバを立て, 接続できるところまでを説明します．大まかには,

(1) OpenVPN と easy-rsa をインストールし,

(2) 認証局や各種証明書, 鍵を作成し,

(3) OpenVPN の設定ファイルを作成・編集し,

(4) クライアントに配るための設定ファイルを作成します．

0. 表記上の注意

本文書では一貫して `sudo` コマンドを用いません．その代わりに, プロンプト文字によってそのコマンド操作が特権を必要とするかを区別します．特権を必要とするコマンド操作である場合にはプロンプト文字を `#` で表示し, さもなくば `$` で表示します．これは, 一般的な B シェルのプロンプト表示と一致します．また, ワーキングディレクトリはユーザのホームディレクトリであることを想定しています．

コマンド操作のうち, ユーザ (あなた) が入力するべき部分は **`Courier Bold`** の書体で表示されます．出力結果などの部分は `Courier` の書体で表示されます．また, 環境によって入力や出力が異なる部分は ***`Bold Italic`*** や *`Italic`* で表示されます．例えば次のようになります．

```
$ echo "hello, world"
hello, world
$ whoami
user
```

1. 基本システムのインストール

ここでは OpenVPN をインストールするまでの準備を行います．

1.1. CentOS 9 Stream をインストールする

CentOS 9 Stream を適当にインストールします．構成は Server を選択します．

1.2. パッケージを更新する

インストール済みのパッケージを最新の状態に更新しておきます．この際, しばしばカーネルの更新が入るので再起動も行います．

```
# dnf upgrade
# reboot
```

2. OpenVPN と easy-rsa のインストール

ここでは OpenVPN と easy-rsa のインストールを行います．

2.1. EPEL を有効にする

EPEL は Extra Packages for Enterprise Linux の頭字語です．OpenVPN と easy-rsa は EPEL の一部として提供されるため, 初期状態ではインストールできません．試してみると良いでしょう．

```
$ dnf search openvpn
No matches found.
$ dnf search openvpn
No matches found.
```

EPEL に含まれるパッケージを利用するためには次のように実行します.

```
# dnf install epel-release
```

2.2. OpenVPN と easy-rsa をインストールする

OpenVPN と easy-rsa をインストールします. easy-rsa パッケージには認証局や証明書の発行を簡単に行うためのシェルスクリプトが含まれます.

```
# dnf install openvpn easy-rsa
```

正常にインストールされたかを確かめるためには次のように実行します.

```
$ hash -r         # ハッシュテーブルのクリア
$ openvpn --version
$ /usr/share/easy-rsa/3/easyrsa
```

2.3. easy-rsa のスクリプトへのエイリアスをつくる

現状, easy-rsa のスクリプトを利用するにためにはスクリプトへのフルパスを指定する必要があります. これではとても手間になるのでエイリアスを作成して実行を楽にします. 次のように実行すると, そのシェルが実行されている間は easyrsa コマンドを利用可能になります.

```
$ alias easyrsa="/usr/share/easy-rsa/3/easyrsa"
$ easyrsa         # 動作確認用
```

3. 証明書たちの作成

ここでは easy-rsa のスクリプトを利用して各種証明書や鍵を発行します. ここで発行される証明書は所謂 “オレオレ証明書” であることに留意してください (個人利用では問題にならないかもしれません).

3.1. 作業用ディレクトリを初期化する

作業用ディレクトリを初期化します. この操作でディレクトリ pki が出現します.

```
$ easyrsa init-pki
```

3.2. 認証局 (CA) をつくる

認証局 (CA) をつくります. この際, CA 用のパスフレーズ (New CA Key Passphrase) と PEM 用のパスフレーズ (PEM pass phrase) を求められるのでそれぞれ設定します. さらに, 認証局の一般名 (Common Name) も求められますが, そのまま [Enter] を押下することでデフォルトの名前 (Easy-RSA CA) を利用できます. この操作でファイル ca.crt が所定のディレクトリに出現します.

```
$ easyrsa build-ca
```

3.3. サーバ証明書をつくる

サーバ証明書と秘密鍵をつくります. この際, 認証局の PEM 用のパスフレーズを求められるので入力します. サーバ名 (*server*) は任意の名前を利用できます (ファイル名なども同様に変化します). この操作でファイル *server*.crt とファイル *server*.key が所定のディレクトリに出現します.

```
$ easyrsa build-server-full server nopass
```

3.4. クライアント証明書をつくる

クライアント証明書と秘密鍵をつくります. この際, 認証局の PEM 用のパスフレーズを求められるので入力します. クライアント名 (*client*) は任意の名前を利用できます (ファイル名なども同様に

変化します). この操作でファイル *client*.crt とファイル *client*.key が所定のディレクトリに出現します.

```
$ easyrsa build-client-full client nopass
```

3.5. Diffie–Hellman パラメータをつくる

Diffie–Hellman 鍵交換に用いられるパラメータとなる素数をつくります. とても大きな素数を生成しているので少し時間が掛かります. この操作でファイル dh.pem が所定のディレクトリに出現します.

```
$ easyrsa gen-dh
```

3.6. TLS-Auth 鍵をつくる

TLS-Auth 鍵をつくります. HMAC 認証の追加レイヤの実現のために必要なようです. この操作でファイル ta.key がディレクトリ pki 配下に出現します. **このファイルは使われないかもしれません.**

```
$ openvpn --genkey secret ./pki/ta.key
```

3.7. ディレクトリ pki の中身

ここで, ディレクトリ pki の中身を確認しておきましょう (一部のファイルやディレクトリを省略しています).

```
pki/                                3.1. で作成した作業用ディレクトリ
|-- ca.crt                          3.2. で作成した認証局証明書
|-- certs_by_serial/...
|-- dh.pem                          3.5. で作成した DH パラメータ
|-- index.txt
|-- index.txt.attr
|-- issued/                         証明書のディレクトリ
|   |-- client.crt                  3.4. で作成したクライアント証明書
|   `-- server.crt                  3.3. で作成したサーバ証明書
|-- openssl-easyrsa.cnf
|-- private/                        秘密鍵のディレクトリ
|   |-- ca.key                      3.2. で作成した認証局秘密鍵
|   |-- client.key                  3.4. で作成したクライアント秘密鍵
|   `-- server.key                  3.3. で作成したサーバ秘密鍵
|-- renewed/...
|-- reqs/...
|-- revoked/...
|-- safessl-easyrsa.cnf
|-- serial
`-- ta.key                          3.6. で作成した TLS-Auth 鍵
```

3.8. 証明書たちを配置する

上の作業で作成した各種証明書や鍵を所定のディレクトリ (/etc/openvpn/server 配下) に配置します.

```
# cp -R ./pki/* /etc/openvpn/server/.
# ls /etc/openvpn/server/.        # 確認用
```

4. IP フォワーディングの設定

OpenVPN の機能のなかで IP フォワーディングを利用するので有効にしておきます. この操作でファイル /etc/sysctl.d/01-ipv4_forward.conf が作成されます.

```
# echo "net.ipv4.ip_forward = 1" | tee /etc/sysctl.d/01-ipv4_forward.conf
net.ipv4.ip_forward = 1
# sysctl --system          # 設定を適用する
$ sysctl -a 2>/dev/null | grep "ip_forward"        # 設定確認
net.ipv4.ip_forward = 1
net.ipv4.ip_forward_update_priority = 1
net.ipv4.ip_forward_use_pmtu = 0
```

5. 設定ファイルの記述

ここでは OpenVPN の設定ファイルを記述します.

5.1. 設定ファイルのサンプルをコピーする

OpenVPN に同梱されているサンプルをもとに設定ファイルを記述します. まずは設定ファイルのサンプルをコピーしてきます.

```
# cp /usr/share/doc/openvpn/sample/sample-config-files/server.conf \
    /etc/openvpn/server/.
```

5.2. 設定ファイルを編集する

上の操作でコピーしてきた設定ファイル /etc/openvpn/server/server.conf を編集します. 編集点の要約は次の通りです.

```
$ gzip -c9 server.conf.diff | base64  # これはただのメモです
H4sICDvaw2ICA2RhbS5kaWZmAJVW227jNhB9rr5i4DwkgSxZUpyrHxp0Fy0WBbYBstuiQF9oibaI
lUmBpOxVv75nKCl2dlOgNRJTIud65szQSZLQonN24Wph5aIy5cK0Uu9bvXBi1zZyXJLS6I3aJhvV
SLdw0u6lTXnvhyIriiS7Tool5bcPy7uH/C7Npg/F2X2WRXEc0/c6t1Cj/PqhuHlY5mmRX93l17cv
Oo+PlBT38zyneFgeHyM6I6PJ15IQlaSdKGul5Zw6J0lQpTYbaaX21BrrWVh3u7W0tDGWJGShLFOi
P01HB9U0pKWsyJtgFklT18K2ckGdHfWmg7Ky8iCaJo2SsJ/n98soDo/L5VVErP3p3RPBx+f3T2Oa
P0bJqrXGG/JlC8Xw2FUtFF9249VxO1iZVXJPvtOzIbrSSuE5L2s6j0A/PPGhls38KCzab4Q1ScBj
tfSjcBqAvL2Z51cUD8sA5G/I+Penj1RCRzTOjCA+/frumc7ygkHbCc+Ov8ieuO6sdeGkpFn7pXR5
MQPiVpZe7SUpjWpoasVWXqYRjOIvLVGFpJRAaio+b7C58Z0fGT4GnT2Qq03XVLSWcNqyGvLyURxs
KOc6WS1OTMWs31q1R+qL/2cyAP4ehFGSatk0Q/AWrPLSupRPf5FaWgY10MAcNJD29QMfUeCLcw1V
ddCiBDXCS5Et79JW7ogfoqSqT/aiOLyGx+D+o/QHY7+Aga1pzLbnveeXaF235ipeiKqy0jmlt7RX
Aiy4DBXNr+7ntxQPy1DRURL0NQ1d5FmKRkyzRXF9nU7/2SULrkXJXkMjTTQY4Yto1XauplkgHeX3
RZrf3KUwltErQzNm+BuSxRuS8VuC18vvBYdmMiSQ71aTa2WpNqpk6o/JSceBTwcsXjYKLe+4/9QG
BMaA0UxK4DUcUS0c9keeDHNlmTN4wzKCB8W2J7MZRgBoFEjwiWeNhM3qSI8TSp1n56dDKWCIJqzo
PD+fdscAMT984xLR+Rp9G3iavSIq1onvq/8qGQB7lg0S5gxs33qztaKtAVqp2jpUdFQdJjgpD3ru
Ouc5fuSshhkYYLnJAixhGWD5GbCWZtcGCiIhfhZerREFdwOZpkLGUw14hLBE0vwdpuqHDTcPSS1Y
QaEUmM/zsBci4LnDckcBdYrZFDInDfhWL6bjo5OpSDvxVe263TTyUUcol53l+6DpJ1bI6pQyByQh
4AaMwoA3h2FWFnfLgEJYxlnZ+bZjhFF5nmZe+O44Xg7gWjA7eHthoNFuTt52uhSjY2YGbhOrMFcx
qUGWnnZKoy2Q3mh1vH7H1xSDIYrHo8Ve2AU2Fm/IBCB+6qmSG9E1fk7YpB2qhonshjtia6amd73j
4wtjR/b+oXRlDggXLYSANXdP6BqmtCpRM6j1J2bG63lA6v6I1KFmUGawnogWUVbj/TS8sH/lcQV/
BlFwGQfvNoRk+NqiizVg1gbcxDtfJCuOc/qMaYd8k9XRx+uT+PTgW8iOWD3zFQnHkLQGw4EnfYOS
NMydIHU2VBhVWhunfD/Q4yov5kuKh4WTDqPceLXpT6nra+GBhnw1GdBFXlgwzzEOgTN8+3be4KpV
JUjYQ2bkDyghv7aNKpXHA7704CSP/oI//Hg5NPjxQ3DDmSLocEfHq39R+geD+OYP6gkAAA==
```

```
--- a/server.conf    2022-05-24 17:48:18.000000000 +0900
+++ b/server.conf    2022-07-05 15:26:41.213815700 +0900
@@ -29,11 +29,11 @@
 # on the same machine, use a different port
 # number for each one.  You will need to
 # open up this port on your firewall.
-port 1194
+port 443

 # TCP or UDP server?
-;proto tcp
-proto udp
+proto tcp
+;proto udp

 # "dev tun" will create a routed IP tunnel,
 # "dev tap" will create an ethernet tunnel.
@@ -76,13 +76,13 @@
 # OpenVPN can also use a PKCS #12 formatted key file
 # (see "pkcs12" directive in man page).
 ca ca.crt
-cert server.crt
-key server.key  # This file should be kept secret
+cert issued/server.crt
+key private/server.key  # This file should be kept secret

 # Diffie hellman parameters.
 # Generate your own with:
 #   openssl dhparam -out dh2048.pem 2048
-dh dh2048.pem
+dh dh.pem

 # Network topology
 # Should be subnet (addressing via IP)
@@ -139,7 +139,7 @@
 # address pool (10.8.0.0/255.255.255.0)
 # back to the OpenVPN server.
 ;push "route 192.168.10.0 255.255.255.0"
-;push "route 192.168.20.0 255.255.255.0"
+push "route 192.168.54.0 255.255.255.0"

 # To assign specific IP addresses to specific
 # clients or if a connecting client has a private
@@ -241,7 +241,7 @@
 # a copy of this key.
 # The second parameter should be '0'
 # on the server and '1' on the clients.
-tls-auth ta.key 0 # This file is secret
+;tls-auth ta.key 0 # This file is secret

 # Select a cryptographic cipher.
 # This config item must be copied to
@@ -260,7 +260,7 @@
 # For compression compatible with older clients use comp-lzo
 # If you enable it here, you must also
 # enable it in the client config file.
-;comp-lzo
+comp-lzo
```

```
 # The maximum number of concurrently connected
 # clients we want to allow.
@@ -284,7 +284,7 @@
 # Output a short status file showing
 # current connections, truncated
 # and rewritten every minute.
-status openvpn-status.log
+status /var/log/openvpn-status.log

 # By default, log messages will go to the syslog (or
 # on Windows, if running as a service, they will go to
@@ -294,7 +294,7 @@
 # or the other (but not both).
 ;log         openvpn.log
-;log-append  openvpn.log
+log-append  /var/log/openvpn.log

 # Set the appropriate level of log
 # file verbosity.
@@ -312,4 +312,4 @@

 # Notify the client that when the server restarts so it
 # can automatically reconnect.
-explicit-exit-notify 1
+;explicit-exit-notify 1
```

6. Service ファイルの書き換え

Service ファイルを編集しないと OpenVPN が起動しないようです．これよりも良い解決方法があると信じていますが, 現時点では見出せないのでこれで妥協します．

デフォルトの service ファイルのコピーを取り, その内容を編集します．この操作でファイル /etc/systemd/system/my-openvpn-server@.service を作成し, 編集します．

```
# cp /usr/lib/systemd/system/openvpn-server@.service \
    /etc/systemd/system/my-openvpn-server@.service
# sed -i"" -e"13s/:BF-CBC / /" \
    /etc/systemd/system/my-openvpn-server@.service
```

7. OpenVPN サーバのサービス起動

OpenVPN サーバのサービスを起動します．次のように実行します．ここで, *server* には 3.3. で指定したサーバ名を指定します．**きっと起動します**．

```
# systemctl start my-openvpn-server@server
# systemctl enable my-openvpn-server@server
```

8. Firewall の設定

Firewall の設定をします．今回は (とても行儀の悪いことですが) HTTPS の格好をしてポートを開けてもらいます．HTTPS は TCP の 443 番ポートを利用し, firewall はそれを知っています．さらに, SELinux もそれを知っているので好都合であるからです．**うーん, 行儀が悪過ぎます**．ゾーンを切っている場合は適宜対応してください．

```
# firewall-cmd --permanent --add-service=https
# firewall-cmd --reload
```

9. クライアントに配る設定ファイルの準備

ここでは, クライアントに配るための設定ファイルを準備します.

9.1. 設定ファイルのサンプルをコピーする

まずは OpenVPN に同梱されているサンプルをもとに設定ファイルを記述します. まずは設定ファイルのサンプルをコピーしてきます.

```
$ cp /usr/share/doc/openvpn/sample/sample-config-files/client.conf ./.
```

9.2. 設定ファイルを編集する

上の操作でコピーしてきた設定ファイル clienc.conf を編集します. 編集点の要約は次の通りです.

```
$ gzip -c9 client.conf.diff | base64  # これもただのメモです
H4sICP/vw2ICA2FiYS5kaWZmAG1Sy27jMAw8r7+CwB4dO5ZjN6/Dpuhecguw7QJ7ZBQ6FupYhqS0
yN8vJTlNXz7YFD1DkcPJsgxwKjtFvcul7psfZVGWWVFnZQVivqoWK7HIi+sDabEsiiRNU9h/w5oz
EcTdqq5X1Swvymoh6vkba7OBbDabiBmk8bPZJPAT7g3BKwGX6Uk61R/BaUB4fNiBNh7w9HsHlswL
mV8AT5bAtQQWT/wiFwhoPU738U+A5km2HozmUk4OSRbD82FI0ls2Xd/S4Cs8Mr3V1vVcfLrdAfYH
GLRxoJsPpRn6T59BYg8tvhCczp1TQ0dg6KQdActiFIWmuHyn8QB77LCXBHtyr0TvO7Xc6sg7XbKY
ywQIsaySdPwhlmUu7hZ5XeWiXkJVzRJYfyGVkRRGeWi1ZqkQDA+hT2EsaAxH/uZIDRtZ1JMlpOEd
94Hc14AGuXQujZvmz3SBRvF0A6qwkEYbIJQtjA4AuAfLe2CIxABQIex5WjhbOgQGdt1I8BNLZICv
zyGxwlcz+YS/cDxzyGu6gTn+hE7Xn+Bh/L9kVHMZFQbPUY2SfqY9g1uSz8FoLTqvh2e8x7Rog0xS
G8OeBH/D2eIxOC4PsominswhjR8vnK+xbVg919kMz64NLGWjAB+8GXzRsgmID9fe2US8Ieysjp7y
cK7AUr0VdBh2IXjmr7nQwB/qfL8sl7kMTh8NDq2SINXQjr7dRifHDOjBKW7t+y7/A2+hXjQcBAAA
```

```
--- a/client.conf    2022-05-24 17:48:18.000000000 +0900
+++ b/client.conf    2022-07-05 16:55:43.024815700 +0900
@@ -33,13 +33,13 @@
 # Are we connecting to a TCP or
 # UDP server?  Use the same setting as
 # on the server.
-;proto tcp
-proto udp
+proto tcp
+;proto udp

 # The hostname/IP and port of the server.
 # You can have multiple remote entries
 # to load balance between the servers.
-remote my-server-1 1194
+remote 192.168.54.159 443
 ;remote my-server-2 1194

 # Choose a random host from the remote
@@ -85,9 +85,9 @@
 # a separate .crt/.key file pair
 # for each client.  A single ca
 # file can be used for all clients.
-ca ca.crt
-cert client.crt
-key client.key
+;ca ca.crt
+;cert client.crt
+;key client.key

 # Verify server certificate by checking that the
 # certificate has the correct key usage set.
@@ -105,7 +105,7 @@

 # If a tls-auth key is used on the server
 # then every client must also have the key.
-tls-auth ta.key 1
+;tls-auth ta.key 1

 # Select a cryptographic cipher.
 # If the cipher option is used on the server
```

9.3. 証明書と鍵の連結

上で編集したファイルに各種証明書と鍵を連結します.

```
$ echo '<ca>' >>./client.conf
# cat /etc/openvpn/server/ca.crt >>./client.conf
$ echo '</ca>' >>./client.conf
$ echo '<cert>' >>./client.conf
# awk -e '
    BEGIN { PEM = 0 }
    /^--/ { if (!(PEM = !PEM)) print }
          { if (PEM) print }' /etc/openvpn/server/issued/client.crt \
    >>./client.conf
$ echo '</cert>' >>./client.conf
$ echo '<key>' >>./client.conf
# cat /etc/openvpn/server/private/client.key >>./client.conf
$ echo '</key>' >>./client.conf
```

9.4. クライアントに転送

上で編集したファイルを client.ovpn と名付けてクライアントに転送します．転送方法は適当にやります．**なんとかしてください**．

10. 完成

完成しています．完成しているので完成しています．**本当です**．

全部自動でやってくれるシェルスクリプトがどこかで配布されるかもしれません．期待しないでお待ちください．