**root@computer:/Library/ssl-admin-> ssl-admin**

Program ini akan memandu Anda melalui permintaan, penandatanganan, pengorganisasian, dan pencabutan sertifikat SSL.

**Looks like this is a new install, installing... You will first need to edit the /Library/ssl-admin/ssl-admin.conf default variables. Have you done this? (y/n): y I need the CA credentials. Would you like to create a new CA key and certificate now? (y/n): y Please enter certificate owner's name or ID.**

**Usual format is first initial-last name (jdoe) or hostname of server which will use this certificate. All lower case, numbers OK.**

**Owner []: Mastering OpenVPN File names will use Mastering\_OpenVPN.**

**===> Creating private key with 2048 bits and generating request.**

**Do you want to password protect your CA private key? (y/n): y**

**Generating RSA private key, 2048 bit long modulus**

**.....................................+++ ............+++ e is 65537 (0x10001) Enter pass phrase for Mastering\_OpenVPN.key:**

**Verifying - Enter pass phrase for Mastering\_OpenVPN.key:**

**===> Self-Signing request.**

**Enter pass phrase for /Library/ssl-admin/Mastering\_OpenVPN.key:**

**===> Moving certficate and key to appropriate directory.**

**===> Creating initial CRL.Using configuration from /Library/ssl-admin/ openssl.conf Enter pass phrase for /Library/ssl-admin/active/ca.key:**

**ssl-admin installed Wed Oct 1 12:28:23 CDT 2014**

**OPTIONAL: I can't find your OpenVPN client config. Please copy your config to /Library/ssl-admin/packages/client.ovpn**

**=====================================================**

**# SSL-ADMIN v1.2.1 #**

**=====================================================**

**Please enter the menu option from the following list:**

1. **Update run-time options:**

**Key Duration (days): 3650**

**Current Serial #: 01**

**Key Size (bits): 2048**

**Intermediate CA Signing: NO**

1. **Create new Certificate Request**
2. **Sign a Certificate Request**
3. **Perform a one-step request/sign**
4. **Revoke a Certificate**
5. **Renew/Re-sign a past Certificate Request**
6. **View current Certificate Revokation List8) View index information for certificate.**

**i) Generate a user config with in-line certifcates and keys.**

**z) Zip files for end user.**

**dh) Generate Diffie Hellman parameters. CA) Create new Self-Signed CA certificate. S) Create new Signed Server certificate.**

**q) Quit ssl-admin**

**Menu Item:**

Seperti yang Anda lihat, ssl-admin jauh lebih verbose dan interaktif daripada Easy-RSA. Juga, ssl-admin secara otomatis menghasilkan CRL awal untuk Anda.

Sebelum menu ditampilkan, ada peringatan OPSIONAL tentang konfigurasi OpenVPN. Jika Anda memberikan konfigurasi client.ovpn Anda, ssl-admin dapat secara otomatis mengemas file konfigurasi dengan sertifikat tertanam atau file ZIP multifile. Baris sertifikat file konfigurasi harus generik:

**ca ca.crt cert client.crt key client.key**

Nilai-nilai ini akan secara otomatis diganti dengan kunci sebaris, atau file akan diganti namanya sesuai dengan bagaimana sertifikat, kunci, dan konfigurasi OpenVPN didistribusikan. Sekarang kami menginisialisasi PKI dengan membuat pasangan kunci otoritas sertifikat root, kami dapat mulai membuat server dan sertifikat klien kami.

## OpenVPN server certificates

Pertama, kita akan membuat sertifikat untuk digunakan di server OpenVPN. Opsi menu S akan menghasilkan CSR, kunci, dan akan meminta untuk menandatangani sertifikat oleh CA:

**Menu Item: S Please enter certificate owner's name or ID. Usual format is first initial-last name (jdoe) or hostname of server which will use this certificate. All lower case, numbers OK.**

**Owner []: Mastering OpenVPN Server**

**File names will use Mastering\_OpenVPN\_Server.**

**Please enter certificate owner's name or ID. Usual format is first initial-last name (jdoe) or hostname of server which will use this certificate. All lower case, numbers OK.**

**Owner [Mastering\_OpenVPN\_Server]:**

**Would you like to password protect the private key (y/n): y**

**Generating a 2048 bit RSA private key**

**..................+++ .........+++**

**writing new private key to 'Mastering\_OpenVPN\_Server.key' Enter PEM pass phrase:**

**Verifying - Enter PEM pass phrase:**

**-----**

**===> Serial Number = 01**

**Using configuration from /Library/ssl-admin/openssl.conf Enter pass phrase for /Library/ssl-admin/active/ca.key:**

**Check that the request matches the signature**

**Signature ok The Subject's Distinguished Name is as follows countryName :PRINTABLE:'ZA' stateOrProvinceName :PRINTABLE:'Enlightenment' localityName :PRINTABLE:'Overall' organizationName :PRINTABLE:'Mastering OpenVPN' commonName :PRINTABLE:'Mastering OpenVPN Server' emailAddress :IA5STRING:'root@example.org' Certificate is to be certified until Sep 28 17:48:20 2024 GMT (3650 days)**

**Write out database with 1 new entries**

**Data Base Updated**

**=========> Moving certificates and keys to /Library/ssl-admin/active for production.**

**Can I move signing request (Mastering\_OpenVPN\_Server.csr) to the csr directory for archiving? (y/n): y ===> Mastering\_OpenVPN\_Server.csr moved.**

**MENU**

instead it is replaced with the word **MENU**.

Dalam kode sebelumnya, kami menggunakan sertifikat CN dengan spasi untuk menunjukkan perilaku ssl-admin. Di sini, ia memperingatkan bahwa spasi akan diganti dengan karakter garis bawah, dan memberi pengguna kesempatan untuk mengubah CN, jika diperlukan. Lebih jauh, kami memilih untuk mengamankan kunci pribadi dengan frasa sandi. Akhirnya, pengguna ditanya apakah CSR dapat diarsipkan.

Untuk menampilkan token server tambahan, kami menjalankan lagi perintah openssl untuk menampilkan detail sertifikat. Output berikut menghilangkan beberapa detail utama untuk singkatnya:

**root@computer:/Library/ssl-admin-> openssl x509 -noout -text -in active/ Mastering\_OpenVPN\_Server.crt Certificate:**

**Data:**

**Version: 3 (0x2)**

**Serial Number: 1 (0x1)**

**Signature Algorithm: sha1WithRSAEncryption**

**Issuer: C=ZA, ST=Enlightenment, L=Overall, O=Mastering OpenVPN,**

**CN=Mastering OpenVPN/emailAddress=root@example.org**

**Validity**

**Not Before: Oct 1 17:48:20 2014 GMT**

**Not After : Sep 28 17:48:20 2024 GMT**

**Subject: C=ZA, ST=Enlightenment, O=Mastering OpenVPN,**

**CN=Mastering OpenVPN Server/emailAddress=root@example.org Subject Public Key Info:**

**Public Key Algorithm: rsaEncryption**

**RSA Public Key: (2048 bit) Modulus (2048 bit):**

**...**

**Exponent: 65537 (0x10001) X509v3 extensions:**

**X509v3 Basic Constraints:**

**CA:FALSE Netscape Cert Type:**

**SSL Server Netscape Comment: ssl-admin (OpenSSL) Generated Server Certificate X509v3 Subject Key Identifier:**

**FB:A8:91:01:E3:51:5D:A7:29:8C:54:63:9F:22:7F:F8:DE:AB:**

**5A:39 X509v3 Authority Key Identifier:**

**keyid:1F:85:DF:90:5C:3F:73:A9:03:B9:F4:E6:C2:2C:A3:27:CF: 5B:44:95**

**DirName:/C=ZA/ST=Enlightenment/L=Overall/O=Mastering**

**OpenVPN/CN=Mastering OpenVPN/emailAddress=root@example.org serial:D2:93:32:F0:8E:BC:58:EE**

**X509v3 Extended Key Usage:**

**TLS Web Server Authentication X509v3 Key Usage:**

**Digital Signature, Key Encipherment**

**Signature Algorithm: sha1WithRSAEncryption**

## Perhatikan bahwa Penggunaan Kunci Diperpanjang x509v3 mencakup Server Web TLS

## Autentikasi. Standar yang lebih lama, nsCertType sebagai Netscape Cert Type, juga disertakan untuk kompatibilitas ke belakang. Tidak hanya ini terkait untuk OpenVPN, tetapi ssl-admin ditulis sebagai utilitas manajemen x509 CA umum.

## OpenVPN client certificates

Sertifikat klien dihasilkan dengan cara yang hampir sama dengan sertifikat server. Opsi 4 pada menu akan membuat Permintaan Penandatanganan Sertifikat (CSR) dan selanjutnya menandatangani CSR:

**Menu Item: 4 Please enter certificate owner's name or ID. Usual format is first initial-last name (jdoe) or hostname of server which will use this certificate. All lower case, numbers OK.**

**Owner []: client1**

**File names will use client1.**

**Please enter certificate owner's name or ID. Usual format is first initial-last name (jdoe) or hostname of server which will use this certificate. All lower case, numbers OK.**

**Owner [client1]:**

**Would you like to password protect the private key (y/n): n**

**Generating a 2048 bit RSA private key**

**.........................................................................**

**............................................................+++ .........+++ writing new private key to 'client1.key' -----**

**===> Serial Number = 02**

**=========> Signing request for client1**

**Using configuration from /Library/ssl-admin/openssl.conf Enter pass phrase for /Library/ssl-admin/active/ca.key:**

**Check that the request matches the signature**

**Signature ok The Subject's Distinguished Name is as follows countryName :PRINTABLE:'ZA' stateOrProvinceName :PRINTABLE:'Enlightenment' localityName :PRINTABLE:'Overall' organizationName :PRINTABLE:'Mastering OpenVPN' commonName :PRINTABLE:'client1' emailAddress :IA5STRING:'root@example.org' Certificate is to be certified until Sep 28 18:05:14 2024 GMT (3650 days)**

**Write out database with 1 new entries**

**Data Base Updated**

**=========> Moving certificates and keys to /Library/ssl-admin/active for production.**

**Can I move signing request (client1.csr) to the csr directory for archiving? (y/n): ===> client1.csr moved. MENU**

Menggunakan biner openssl untuk memeriksa sertifikat, kita dapat melihat bahwa sertifikat client1 tidak memiliki ekstensi penggunaan kunci server yang ada dalam sertifikat server yang kita buat sebelumnya.

**root@computer:/Library/ssl-admin-> openssl x509 -noout -text -in active/ client1.crt Certificate:**

**Data:**

**Version: 1 (0x0)**

**Serial Number: 2 (0x2)**

**Signature Algorithm: sha1WithRSAEncryption**

**Issuer: C=ZA, ST=Enlightenment, L=Overall, O=Mastering OpenVPN,**

**CN=Mastering OpenVPN/emailAddress=root@example.org**

**Validity**

**Not Before: Oct 1 18:05:14 2014 GMT**

**Not After : Sep 28 18:05:14 2024 GMT**

**Subject: C=ZA, ST=Enlightenment, O=Mastering OpenVPN, CN=client1/ emailAddress=root@example.org Subject Public Key Info:**

**Public Key Algorithm: rsaEncryption**

**RSA Public Key: (2048 bit) Modulus (2048 bit):**

**...**

**Exponent: 65537 (0x10001)**

**Signature Algorithm: sha1WithRSAEncryption ...**

Sertifikat ini jelas memiliki struktur yang lebih sederhana daripada sertifikat server, dan parameter penggunaan kunci server tidak ada.

Setelah membuat CA, server, dan tiga sertifikat klien kami, kami memiliki struktur direktori berikut:

**root@computer:/Library/ssl-admin-> ls -lrth total 16 -rw-rw-r-- 1 root wheel 541B Oct 1 12:22 ssl-admin.conf drwxr-x--- 2 root wheel 68B Oct 1 12:24 revoked -rw-rw---- 1 root wheel 2.5K Oct 1 12:27 openssl.conf drwxr-x--- 2 root wheel 102B Oct 1 12:28 packages -r--r--r-- 1 root wheel 531B Oct 1 12:43 ssl-admin.conf.sample -r--r--r-- 1 root wheel 2.5K Oct 1 12:43 openssl.conf.sample drwxr-x--- 2 root wheel 340B Oct 1 13:05 prog drwxr-x--- 2 root wheel 340B Oct 1 13:05 csr drwxr-x--- 2 root wheel 544B Oct 1 13:05 active**

## Direktori aktif berisi semua sertifikat dan kunci yang belum dicabut, termasuk sertifikat dan kunci CA. Ketika sertifikat dicabut, sertifikat tersebut dipindahkan dari aktif ke dicabut. Untuk menggunakan utilitas OpenSSL untuk mencabut sertifikat, sertifikat harus ada. Tanpa sertifikat, pengeditan yang berpotensi bermasalah perlu dilakukan secara manual ke file index.txt. Seperti namanya, direktori csr berisi semua CSR. Ini umumnya aman untuk dihapus, dan hanya disimpan untuk pemecahan masalah, atau jika sertifikat perlu dibuat ulang.

## Disarankan agar administrator meninggalkan konten toko sertifikat ke manajemen utilitas. Ini berlaku untuk ssl-admin dan Easy-RSA.

## Direktori prog berisi file operasional openssl dan CRL terbaru. Tidak disarankan file-file ini diganggu, karena ada potensi untuk membuat PKI Anda tidak dapat digunakan jika terjadi kesalahan.

## Terakhir, direktori paket akan berisi semua file yang dapat Anda distribusikan kepada pengguna akhir Anda: tidak hanya klien OpenVPN, tetapi administrator server web dan sebagainya. Pengemasan sertifikat dan kunci memastikan pengguna akhir menerima semua file yang diperlukan dan bahwa mereka berada dalam format yang tepat.

## Other features

## Utilitas ssl-admin memiliki beberapa fitur lain yang mungkin menarik bagi seseorang yang mengelola PKI. Indeks dapat dicari (opsi 8), yang menunjukkan status sertifikat yang diberikan. Tampilan CRL saat ini juga dimungkinkan (opsi 7). ssl-admin mampu mengemas file konfigurasi OpenVPN dengan sertifikat untuk pengguna, baik dalam format inline (opsi i), maupun file terpisah — semua terkandung dalam file zip (opsi z). Dua opsi terakhir ini akan dibahas lebih lanjut dalam buku ini, karena kami menghasilkan konfigurasi server dan klien.

## Multiple CAs and CRLs

## Easy-RSA 3.0 cukup mudah mendukung beberapa CA root. Dengan membuat direktori CA terpisah di bawah EASYRSA root, dan memiliki file vars yang berbeda untuk masing-masing, masing-masing CA individu dapat dikelola dengan Easy-RSA.

## Saat ini, ssl-admin tidak mendukung beberapa CA root, tetapi pembuatan CA menengah didukung.

## Dengan OpenVPN, sebuah instance server tunggal dapat mendukung beberapa CA root, dengan koneksi klien yang telah ditandatangani oleh CA yang diterima. Untuk mengaktifkan dukungan seperti itu, sertifikat CA untuk setiap CA yang berwenang perlu digabungkan bersama menjadi satu file yang dapat dipanggil dengan opsi --ca OpenVPN. Hal yang sama dapat dilakukan dengan daftar pencabutan sertifikat.

## Secara umum, tidak direkomendasikan untuk menggunakan beberapa sertifikat CA untuk satu contoh OpenVPN; pengecualian dapat berupa server, atau migrasi otoritas sertifikat, akuisisi perusahaan atau organisasi, dan sebagainya.

## Dalam kondisi apa pun, sebaiknya menggunakan otoritas sertifikat root browser web untuk rantai sertifikat OpenVPN. Tidak ada cara untuk memastikan siapa yang memiliki sertifikat, dan siapa pun yang termasuk dalam hierarki CA akan dapat terhubung ke instance VPN Anda.

## Ke depan, ada rencana untuk menggabungkan ssl-admin dan proyek Easy-RSA menjadi satu paket administratif PKI yang sepenuhnya mampu. Keinginannya adalah agar Easy-RSA 4.0 dapat mewujudkan migrasi ini menggunakan fitur terbaik dari kedua utilitas.

## Extra security – hardware tokens, smart cards, and PKCS#11

### Di bagian ini, kami akan memberikan beberapa informasi latar belakang tentang perangkat keras kriptografi. Anda akan belajar cara membuat kunci pribadi pada token perangkat keras, dan cara menyalin sertifikat X.509 yang terkait ke token juga. Setelah itu, kita akan membahas bagaimana OpenVPN dapat menemukan dan menggunakan pasangan kunci sertifikat / privat ini untuk membuat koneksi VPN.

### Initializing a hardware token

Diasumsikan bahwa paddicli Aladdin atau SafeNet AuthenticationClient diinstal dan bahwa token perangkat keras dikenali oleh perangkat lunak driver. Jika eToken telah diinisialisasi, lewati langkah ini.

Pertama, buka jendela properti klien eToken, dan klik Initialize eToken. Ini akan memunculkan kotak dialog berikut:



### Isi kata sandi token dan kata sandi administrator, hapus centang Kata Sandi Token harus diubah pada tanda centang masuk pertama, dan klik Mulai.

### Semua konten pada token sekarang akan dihancurkan dan eToken akan diinisialisasi dengan token baru dan kata sandi administrator.

### Generating a certificate/private key pair

### Saat menggunakan token perangkat keras, proses menghasilkan sertifikat dan pasangan kunci pribadi sedikit berbeda dibandingkan dengan menggunakan ssl-admin atau alat Easy-RSA. Dengan ssl-admin atau Easy-tools, kunci pribadi, permintaan sertifikat, dan sertifikat X.509 dihasilkan dalam satu langkah. Dengan token perangkat keras, pertama-tama kita perlu membuat kunci pribadi pada token.

### Dengan menggunakan kunci pribadi ini, kita perlu membuat CSR. CSR ini kemudian ditandatangani oleh CA, yang menghasilkan sertifikat X.509. Sertifikat ini kemudian ditulis kembali ke token. Alat ssl-admin dan Easy-RSA sebenarnya mengikuti proses yang sama, tetapi mereka menyembunyikan file CSR dari pengguna.

### Generating a private key on a token

Untuk menghasilkan kunci privat pada eToken, kita memerlukan perintah pkcs11-tool, yang merupakan bagian dari paket OpenSC. Paket OpenSC tersedia untuk Microsoft Windows, Mac OS X, dan Linux. Perintah pkcs11-tool menyediakan antarmuka ke token perangkat keras menggunakan driver PKCS # 11. Driver PKCS # 11 yang disertakan dengan perangkat lunak driver Aladdin / SafeNet adalah libeTPkcs11.so (Linux dan Mac OS X) atau eTPkcs11.dll (Windows).

Perintah berikut dikeluarkan pada mesin Linux 64-bit dan menghasilkan kunci RSA 2048-bit yang diidentifikasi menggunakan label movpn dan ID 20141001. Saat kita membuat kunci pribadi, kita wajib masuk ke token:

**# pkcs11-tool --module libeTPkcs11.so \**

**--keypairgen --key-type rsa:2048 \**

**--label "movpn" --id 20141001 --login**

**Using slot 0 with a present token (0x0) Logging in to "Mastering OpenVPN".**

**Please enter User PIN: [enter Token password] Key pair generated:**

**Private Key Object; RSA label: movpn ID: 20141001**

**Usage: decrypt, sign, unwrap**

**Public Key Object; RSA 2048 bits label: movpn ID: 20141001**

**Usage: encrypt, verify, wrap**

### Diperlukan waktu untuk membuat kunci 2048-bit pada token perangkat keras, di mana lampu merah pada token perangkat keras mati. Setelah itu lampu akan menyala kembali.

### Output dari perintah di atas memberitahu kita bahwa kunci pribadi RSA dihasilkan, bersama dengan kunci publik RSA publik 2048-bit. Ini tidak sama dengan sertifikat SSL. Untuk menghasilkan sertifikat SSL atau X.509, pertama-tama kita perlu membuat permintaan sertifikat.

### Generating a certificate request

Kita perlu menggunakan engine OpenSSL engine\_pkcs11 untuk menghasilkan permintaan sertifikat menggunakan kunci pribadi dari token perangkat keras. Engine engine\_pkcs11 paling baik digunakan dengan file openssl.cnf khusus. Kami pertama kali membuat file ini:

**openssl\_conf = openssl\_def**

**[ openssl\_def ] engines = engine\_section**

**[ engine\_section ] pkcs11 = pkcs11\_section**

**[ pkcs11\_section ] engine\_id = pkcs11**

**dynamic\_path = /usr/lib64/openssl/engines/engine\_pkcs11.so**

**MODULE\_PATH = /usr/lib64/libeTPkcs11.so init = 0**

**[ req ]**

**distinguished\_name = req\_distinguished\_name**

**[ req\_distinguished\_name ]**

File ini dihasilkan untuk sistem CentOS Linux 64-bit. Pada sistem lain, jalur dan nama driver akan berbeda. Perhatikan bahwa pernyataan dalam file openssl.cnf peka huruf besar kecil!

Kami sekarang membuat permintaan sertifikat dengan subjek / CN = movpn, menggunakan perintah openssl berikut:

**$ openssl req -engine pkcs11 -keyform engine -key 20141001 \**

**-new -text -out movpn.csr -config openssl.cnf \ -subj "/CN=movpn" engine "pkcs11" set. PKCS#11 token PIN: [enter Token password]**

### Perintah tidak menghasilkan output lebih lanjut ketika berhasil. Seharusnya sekarang ada file movpn.csr, yang harus ditandatangani oleh CA yang diatur sebelumnya dalam bab ini. Diasumsikan bahwa sertifikat yang ditandatangani akan dinamai movpn.crt.

### Writing an X.509 certificate to the token

OpenVPN mengharapkan sertifikat X.509 hadir pada token perangkat keras. Oleh karena itu, kita harus terlebih dahulu menulis sertifikat X.509 dari langkah sebelumnya ke eToken.

Pertama, konversikan sertifikat yang ditandatangani ke format DER:

**$ openssl x509 -in movpn.crt -outform der -out movpn.der**

Next we write the **DER** file to the token:

**$ pkcs11-tool --module libeTPkcs11.so \**

**--write-object movpn.der --type cert \**

**--label movpn --id 20141001 --login**

**Using slot 0 with a present token (0x0) Logging in to "Mastering OpenVPN".**

**Please enter User PIN: [enter Token password] Created certificate: Certificate Object, type = X.509 cert label: movpn ID: 20141001**

We need to verify that the IDs of the private key and certificate match:

**$ pkcs11-tool --module libeTPkcs11.so --login -O**

**Using slot 0 with a present token (0x0) Logging in to "Mastering OpenVPN".**

**Please enter User PIN: [enter Token password] Private Key Object; RSA label: movpn ID: 20141001**

**Usage: decrypt, sign, unwrap Public Key Object; RSA 2048 bits label: movpn ID: 20141001**

**Usage: encrypt, verify, wrap Certificate Object, type = X.509 cert label: movpn ID: 20141001**

The token is now ready to be used with OpenVPN.

### Getting a hardware token ID

Untuk menggunakan sertifikat dan kunci pribadi dari token perangkat keras, Anda harus terlebih dahulu mengetahui ID token perangkat keras yang diharapkan OpenVPN. Ini dilakukan dengan menggunakan opsi --show-pkcs11-id:

**$ openvpn --show-pkcs11-ids /usr/lib64/libeTPkcs11.so The following objects are available for use.**

**Each object shown below may be used as parameter to --pkcs11-id option please remember to use single quote mark.**

**Certificate**

**DN: CN=movpn**

**Serial: 01**

**Serialized id: SafeNet\x20Inc\x2E/eToken/00a3659e/Mastering\ x20OpenVPN/20141001**

The serialized ID consists of the following:

* The name of the PKCS#11 driver (**SafeNet Inc**)
* The name of the product (**eToken**)
* The serial number of the token (**00a3659e**)
* The name of the token (**Mastering OpenVPN**)
* The ID of the certificate and private key on the token (**20141001**)

### Using a hardware token with OpenVPN

Kami sekarang akhirnya siap menggunakan token perangkat keras di OpenVPN. Untuk menggunakannya, kami mengganti baris dalam file konfigurasi OpenVPN:

**cert myclient.crt key myclient.key**

With the options **pkcs11-providers** and **pkcs11-id**:

**pkcs11-providers /usr/lib64/libeTPkcs11.so pkcs11-id 'SafeNet\x20Inc\x2E/eToken/00a3659e/Mastering\**

**x20OpenVPN/20141001'**