End-user stuff

Poor man's benchmark

Quick way to compare processing power of CPUs.

Toggle line numbers

```
openssl speed sha1
```

To test whether the CPU and installed version of OpenSSL can work with crypto acceleration (i.e. AES-NI):

Toggle line numbers

```
openssl speed aes-256-cbc
openssl speed -evp aes-256-cbc
```

throughput should be faster (bigger numbers) with the second command.

Create certificate request/unsigned key

Toggle line numbers

```
# Create a key at the same time
openssl req -nodes -new -sha256 -keyout $DOMAIN.key.pem -out $DOMAIN.csr.pem
# Use an existing key
openssl req -nodes -new -sha256 -key $DOMAIN.key.pem -out $DOMAIN.csr.pem
```

$DOMAIN.key.pem will act as an `SSLCertificateKeyFile` for mod_ssl in Apache.

Create certificate request w/ SubjectAltName fields

SubjectAltName fields let a certificate apply to more than 1 domain. Unfortunately, OpenSSL does not allow to create these easily from the command line.

Create a configuration file, $DOMAIN.conf:

Toggle line numbers

```
cat > $DOMAIN.conf << EOF

[req]
distinguished_name = req_distinguished_name
req_extensions = req_ext

[req_distinguished_name]
```

```
countryName = Country Name (2 letter code)
countryName_default = US
stateOrProvinceName = State or Province Name (full name)
stateOrProvinceName_default = New York
localityName = Locality Name (eg, city)
localityName_default = New York City
organizationalUnitName = Organizational Unit Name (eg, section)
commonName = Common Name
commonName_default = $DOMAIN
commonName_max = 64

[req_ext]
subjectAltName = @alt_names

[alt_names]
DNS.1    = $DOMAIN
DNS.2    = www.$DOMAIN

EOF
```

Then use this configuration file to create a CSR:

Toggle line numbers

```
openssl req -nodes -new -sha256 -key $DOMAIN.key.pem -out $DOMAIN.csr.pem -config $DOMAIN.conf
```

Show key fingerprint

Toggle line numbers

```
openssl x509 -subject -dates -fingerprint -in $DOMAIN.key.pem
```

Generate key

Toggle line numbers

```
# RSA key
openssl genrsa -out $DOMAIN.key.pem 4096
# EC key (using prime256v1 curve)
openssl ecparam -out $DOMAIN.key.pem -name prime256v1 -genkey
```

Display certificate information

Toggle line numbers

```
# For a certificate signing request
openssl req -text -noout -in $DOMAIN.csr.pem
# For a generated certificate
openssl x509 -in $DOMAIN.crt.pem -noout -text
```

Creating a PEM file for servers

Toggle line numbers

```
cat $DOMAIN.key.pem $DOMAIN.crt.pem $DOMAIN.dhp.pem > $DOMAIN.pem
```

Used by courier-imap, etc.

If there are intermediate certificates, those must be concatenated AFTER the other certificates.

Creating a PKCS12-format file

Toggle line numbers

```
openssl pkcs12 -export -in $DOMAIN.crt.pem -inkey $DOMAIN.key.pem
-out blah.p12 -name "Bill Gates"
```

Used for creating certificates used in e-mail clients and web browsers

Signing e-mails

Toggle line numbers

```
openssl smine -sign -in msg.txt -text -out msg.encrypted -signer
$DOMAIN.crt.pem -inkey $DOMAIN.key.pem
```

Certificate Authority stuff

When setting up a new CA on a system, make sure index.txt and serial exist (empty and set to 01, respectively), and create directories private and newcert. Edit openssl.cnf - change default_days, certificate and private_key, possibly key size (1024, 1280, 1536, 2048) to whatever is desired.

Create CA certificate

Toggle line numbers

```
openssl req -new -x509 -keyout private/something-CA.key.pem -out
./something-CA.crt.pem -days 3650
```

Export CA certificate in DER format

Toggle line numbers

```
openssl x509 -in something-CA.crt.pem -outform der -out something-
CA.crt
```

Used by web browsers.

Revoke certificate

Toggle line numbers

```
openssl ca -revoke $DOMAIN.crt.pem
```

Generate Certificate Revocation List (CRL)

Toggle line numbers

```
openssl ca -gencrl -out crl/$DOMAIN-CA.crl
```

Sign Certificate Request

Toggle line numbers

```
openssl ca -out blah.crt.pem -in $DOMAIN.req.pem
```

blah.crt.pem acts as `SSLCertificateFile` for Apache

Create Diffie-Hoffman Parameters for Current CA

Toggle line numbers

```
openssl dhparam -out $DOMAIN-CA.dhp.pem 1536
```

Create self-signed certificate from generated key

Toggle line numbers

```
openssl req -new -x509 -sha256 -key $DOMAIN.key.pem -out
$DOMAIN.crt.pem
```

Use only when you've no CA and will only be generating one key/certificate (useless for anything that requires signed certificates on both ends)

Command-line tricks

Simple file encryption

Toggle line numbers

```
openssl enc -bf -A -in file_to_encrypt.txt
```

Simple file decryption

Toggle line numbers

```
openssl enc -bf -d -A -in file_to_encrypt.txt
```

Verify hosts

Toggle line numbers

```
# IMAP
openssl s_client -connect localhost:993 -quiet > /dev/null
# SMTP
openssl s_client -connect localhost:465 -quiet > /dev/null
# HTTP
echo HEAD / | openssl s_client -connect localhost:443 -quiet > /dev/null
```

Depth (first line) should be 2, with a return value of 0.

CategoryCheatSheet

SamatsWiki: CheatSheet/OpenSSL (last edited 2014-12-16 18:55:37 by SamatJain)