Miscellaneous

Powershell

OpenSSL - create a new server and client cetificate with private keys

```
function New-ServerCertificate{
    openssl req -x509 -out SERVER-CERT.pem -subj "/CN=localhost" -nodes -newkey
rsa:2048 -keyout SERVER-KEY.pem
}

function New-ClientConfig{
    New-Item "CLIENT-CONF.conf"
    Add-Content "CLIENT-CONF.conf" "[req]"
    Add-Content "CLIENT-CONF.conf" "default_bits = 2048"
    Add-Content "CLIENT-CONF.conf" "prompt = no"
    Add-Content "CLIENT-CONF.conf" "default_md = sha256"
    Add-Content "CLIENT-CONF.conf" "req_extensions = req_ext"
    Add-Content "CLIENT-CONF.conf" "distinguished_name = dn"
    Add-Content "CLIENT-CONF.conf" "[dn]"
    Add-Content "CLIENT-CONF.conf" "C = UK"
    Add-Content "CLIENT-CONF.conf" "ST = Londinium"
    Add-Content "CLIENT-CONF.conf" "L = London"
    Add-Content "CLIENT-CONF.conf" "O = Client one"
    Add-Content "CLIENT-CONF.conf" "OU = OpSec"
    Add-Content "CLIENT-CONF.conf" "emailAddress = client1@test.com"
    Add-Content "CLIENT-CONF.conf" "CN = client1.test.com"
    Add-Content "CLIENT-CONF.conf" "[req_ext]"
    Add-Content "CLIENT-CONF.conf" "subjectAltName = @alt_names"
    Add-Content "CLIENT-CONF.conf" "extendedKeyUsage=clientAuth"
    Add-Content "CLIENT-CONF.conf" "[alt_names]"
    Add-Content "CLIENT-CONF.conf" "DNS = test.com"
}

function New-ClientCertificate{
    openssl genrsa -out CLIENT-KEY.pem 2048
    openssl req -new -key CLIENT-KEY.pem -out CLIENT-CSR.pem -subj "/CN=client1"
    openssl x509 -req -days 365 -in CLIENT-CSR.pem -CA SERVER-CERT.pem -CAkey
SERVER-KEY.pem -CAcreateserial -out CLIENT-CERT.pem -extfile CLIENT-CONF.conf -
extensions req_ext
    openssl verify -CAfile .\SERVER-CERT.pem .\SERVER-CERT.pem .\CLIENT-CERT.pem
}

function New-SslServerClientCertificates{
    New-ServerCertificate
    New-ClientConfig
    New-ClientCertificate
}
```

Encode/decode a string to Base64

```
function Convert-StringToBase64($s){
    $bytes = [System.Text.Encoding]::Unicode.GetBytes($s)
    $enc=[Convert]::ToBase64String($bytes)
    Write-Output $enc
}

function Convert-Base64ToString($s){
    $dec =
[System.Text.Encoding]::Unicode.GetString([System.Convert]::FromBase64String($s))
    Write-Output $dec
}
```

Other notes

SecureNetworking project notes.txt

1. ```
   1. Create certificate and key:
      openssl req -x509 -out SERVER-CERT.pem -subj "/CN=localhost" -nodes -
   newkey rsa:2048 -keyout SERVER-KEY.pem

   ----- Custom SSL Server -----
   2. Verify certificate and key (visual inspection of decoded PEM):
      openssl x509 -in SERVER-CERT.pem  -text -noout
      openssl rsa -in SERVER-KEY.pem -text -noout

   3. Validate certificate and key (compare modulus):
      openssl rsa -in SERVER-KEY.pem -noout -modulus | sha1sum
      openssl x509 -in SERVER-CERT.pem -noout -modulus | sha1sum

   4. Validate certificate and key (with web browser):
      N.B. The -www gives a response in HTML so the browser can display the
   details of the SSL exchange.
      openssl s_server -key SERVER-KEY.pem -cert SERVER-CERT.pem  -accept 443 -
   www
      [Open browser window]

   [Create server-side code]
   5. Validate certificate and key in your custom server:
      openssl s_client -connect [IP Address]:[port]

   ----- Custom SSL Client -----
   [Create client-side code]
   5. Validate certificate and key in your custom server:
      N.B. Notice the absence of the -www switch.
      openssl s_server -key SERVER-KEY.pem -cert SERVER-CERT.pem  -accept 443
   ```
   ```