Setting up signed and encrypted email with openssl, part 1.

by Philippe Bogaerts, alias xxradar.
http://www.radarhack.com
mailto:xxradar@radarhack.com.

Version 1.0 20-09-2003

1. Introduction

Spending months in studying PKI and certificate related stuff, I started using openssl to do it 'by hand' instead of with a fancy GUI. I came across so many interesting articles and other things that did not work, that I decided to write down what actually worked. This is an introduction paper, explaining the install and basic configuration of an openssl based CA and generating certificates for use in Outlook.

2. Installing openssl

You can obtain openssl for win32 by doing a search on google. There is a simple release, a zip file containing the binary and some DLL's. I also found a version with a setup program. If you install it manually, make sure that the DLL files are copied into the %system%\system32 directory. Also take a look at http://www.openssl.org of course.

- Install openssl in a directory, for example c:\openssl

- Create a subdirectory, for example my_ca
  c:\openssl\my_ca

- Create within c:\openssl\my_ca\ the following subdirectories
  certs
  crl
  csr
  newcerts
  private

- Create in c:\openssl\my_ca\ a file "index.txt" with nothing in (0 bytes length). Type "notepad index.txt", create the file and save. Verify that the file totally empty. You can also try "copy con index.txt" followed by ctrl+z.

- Create in c:\openssl\my_ca\ a file "serial" starting with 01.
  Type echo 01 >serial

- Then create or edit the c:\openssl\openssl.cnf and edit the file, to reflect the directory layout as above stated. Be careful, there are so many examples on the internet that do not seem to work on win32.

Here is a sample openssl.cnf file (tested on W2k sp4)

```
#
# SSLeay example configuration file.
# This is mostly being used for generation of certificate requests.
#

RANDFILE          = .rnd

####################################################################
[ ca ]
default_ca  = CA_default                    # The default ca section

####################################################################
[ CA_default ]

dir               = ./my_ca           # Where everything is kept
certs             = $dir/certs        # Where the issued certs are kept
crl_dir           = $dir/crl          # Where the issued crl are kept
database          = $dir/index.txt    # database index file.
new_certs_dir     = $dir/newcerts     # default place for new certs.
```

```
certificate       = $dir/certs/ca.crt     # The CA certificate
serial            = $dir/serial           # The current serial number
crl               = $dir/crl.pem          # The current CRL
private_key       = $dir/private/cakey.key      # The private key
RANDFILE          = $dir/private/private.rnd    # private random number
file

x509_extensions   = x509v3_extensions     # The extentions to add to
the cert
default_days      = 365                   # how long to certify for
default_crl_days  = 30                    # how long before next CRL
default_md        = md5                   # which md to use.
preserve          = no                    # keep passed DN ordering

# A few difference way of specifying how similar the request should
look
# For type CA, the listed attributes must be the same, and the optional
# and supplied fields are just that :-)
policy            = policy_match

# For the CA policy
[ policy_match ]
countryName             = optional
stateOrProvinceName     = optional
organizationName        = optional
organizationalUnitName  = optional
commonName              = supplied
emailAddress            = optional

# For the 'anything' policy
# At this point in time, you must list all acceptable 'object'
# types.
[ policy_anything ]
countryName             = optional
stateOrProvinceName     = optional
localityName            = optional
organizationName        = optional
organizationalUnitName  = optional
commonName              = supplied
emailAddress            = optional

####################################################################
[ req ]
default_bits            = 1024
default_keyfile         = privkey.pem
distinguished_name      = req_distinguished_name
attributes              = req_attributes

[ req_distinguished_name ]
countryName             = Country Name (2 letter code)
countryName_min         = 2
countryName_max         = 2
stateOrProvinceName     = State or Province Name (full name)
localityName            = Locality Name (eg, city)
organizationName        = Organization Name (eg, company)
organizationalUnitName  = Organizational Unit Name (eg, section)
commonName              = Common Name (eg, your website's domain name)
commonName_max          = 64
emailAddress            = Email Address
emailAddress_max        = 40
```

```
[ req_attributes ]
challengePassword       = A challenge password
challengePassword_min   = 4
challengePassword_max   = 20

[ x509v3_extensions ]

# under ASN.1, the 0 bit would be encoded as 80
#nsCertType
#nsBaseUrl
#nsRevocationUrl
#nsRenewalUrl
#nsCaPolicyUrl
#nsSslServerName
#nsCertSequence
#nsCertExt
#nsDataType
```

The installation process is done.
We are ready to start building our CA.

3. Create a CA certificate and keys.

Once this done, we must create our CA keys and a CA root certificate. First of all, generate RSA keys for the CA.

```
C:\openssl>openssl genrsa -des3 -out ./my_ca/private/cakey.key 2048
Loading 'screen' into random state - done
warning, not much extra random data, consider using the -rand option
Generating RSA private key, 2048 bit long modulus
....+++
.....................................................................+++
e is 65537 (0x10001)
Enter PEM pass phrase:
Verifying password - Enter PEM pass phrase:
C:\openssl>
```

Once we have a key pair, we can generate the CA certificate.

```
C:\openssl>openssl req -new -x509 -days 365 -key ./my_ca/private/cakey.key -out
./my_ca/certs/ca.crt
Using configuration from ./openssl.cnf
Enter PEM pass phrase:
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:be
State or Province Name (full name) [Some-State]:br
Locality Name (eg, city) []:my_town
Organization Name (eg, company) [Internet Widgits Pty Ltd]:radarhack.com
Organizational Unit Name (eg, section) []:r&d
Common Name (eg, YOUR name) []:my_ca
Email Address []:
C:\openssl>
```

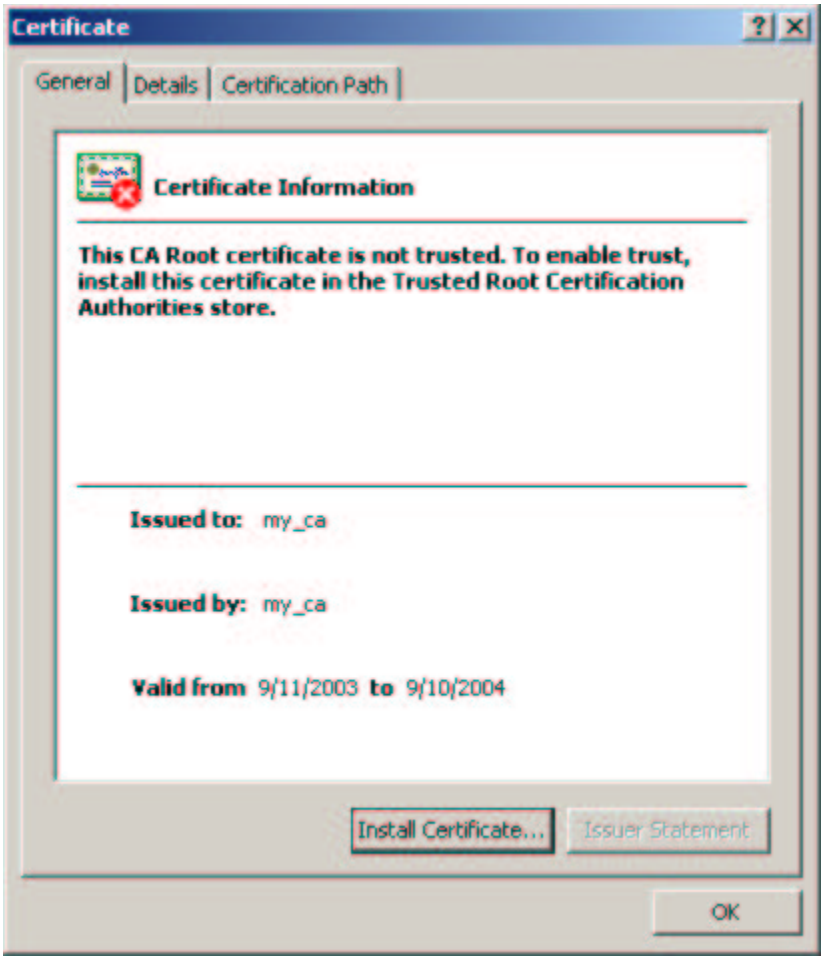
Make sure this .crt file as well as the key is reflected in the openssl.cnf file.

```
..
[ CA_default ]

dir               = ./my_ca          # Where everything is kept
certs             = $dir/certs       # Where the issued certs are kept
crl_dir           = $dir/crl         # Where the issued crl are kept
database          = $dir/index.txt   # database index file.
new_certs_dir     = $dir/newcerts    # default place for new certs.
certificate       = $dir/certs/ca.crt      # The CA certificate
serial            = $dir/serial            # The current serial number
crl               = $dir/crl.pem           # The current CRL
private_key       = $dir/private/cakey.key      # The private key
RANDFILE          = $dir/private/private.rnd    # private random number
..
```

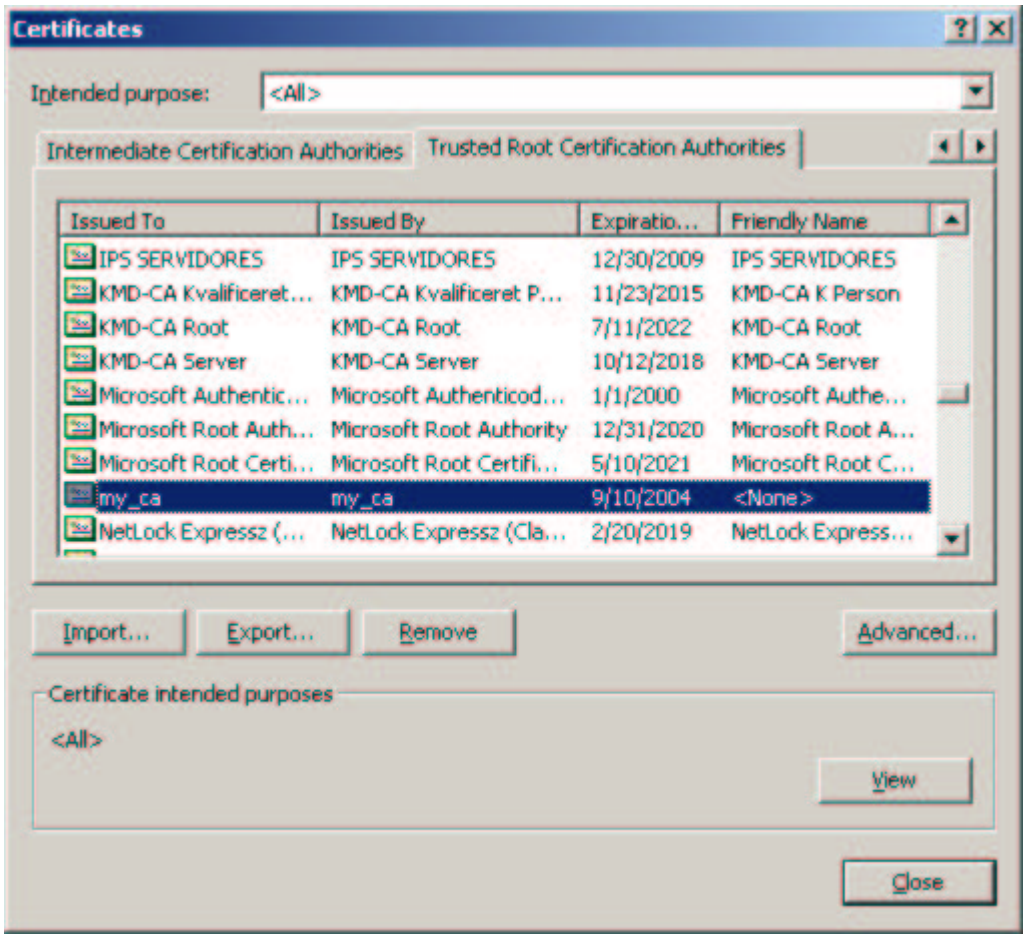
This is the CA's root certificate. This certificate must be distributed to everyone that should trust this CA in a secure manner. Copy (or distribute via http (not secure) the file to the client computers and install the certificate by double clicking or

```
C:\openssl\my_ca\certs>start ca.crt
```



After clicking Install Certificate (and trusting the CA certificate) it will show up in IE.

Go to **Tools->Content->Certificates->Trusted Root Certification Authorities**



4. Create certificates for the users.

Generate a key for a user or server. There is no technical difference in server certificates compared to user certificates. The CN will contain the F.Q.D.N. for servers, and the user name for users.

```
C:\openssl>openssl genrsa -des3 -out .\my_ca\private\user1.key 2048
Loading 'screen' into random state - done
warning, not much extra random data, consider using the -rand option
Generating RSA private key, 2048 bit long modulus
..................+++
..................+++
e is 65537 (0x10001)
Enter PEM pass phrase:
Verifying password - Enter PEM pass phrase:
```

Generate a CSR for user1 (certificate signing request)

```
C:\openssl>openssl req -new -key .\my_ca\private\user1.key -out .\my_ca\csr\user1.csr
Using configuration from .\openssl.cnf
Enter PEM pass phrase:   (to unlock user1.key)
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:be
State or Province Name (full name) [Some-State]:br
Locality Name (eg, city) []:my_town
Organization Name (eg, company) [Internet Widgits Pty Ltd]:radarhack.com
Organizational Unit Name (eg, section) []:r&d
Common Name (eg, YOUR name) []:xxradar
Email Address []:xxradar@radarhack.com

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
An optional company name []:

C:\openssl>
```

The next step is to sign the certificate

```
C:\openssl>openssl ca -config .\openssl.cnf -policy policy_anything -out
.\my_ca\newcerts\user1.pem -infiles .\my_ca\csr\user1.csr
Using configuration from .\openssl.cnf
Loading 'screen' into random state - done
Enter PEM pass phrase: (to unlock cakey.key)
Check that the request matches the signature
Signature ok
The Subjects Distinguished Name is as follows
countryName           :PRINTABLE:'be'
stateOrProvinceName   :PRINTABLE:'br'
localityName          :T61STRING:'my_town'
organizationName      :PRINTABLE:'radarhack.com'
organizationalUnitName:T61STRING:'r&d'
commonName            :PRINTABLE:'xxradar'
emailAddress          :IA5STRING:'xxradar@radarhack.com'
                                  ….(must match an email account in outlook)
Certificate is to be certified until Sep 15 18:34:59 2004 GMT (365 days)
Sign the certificate? [y/n]:y


1 out of 1 certificate requests certified, commit? [y/n]y
Write out database with 1 new entries
Data Base Updated
```

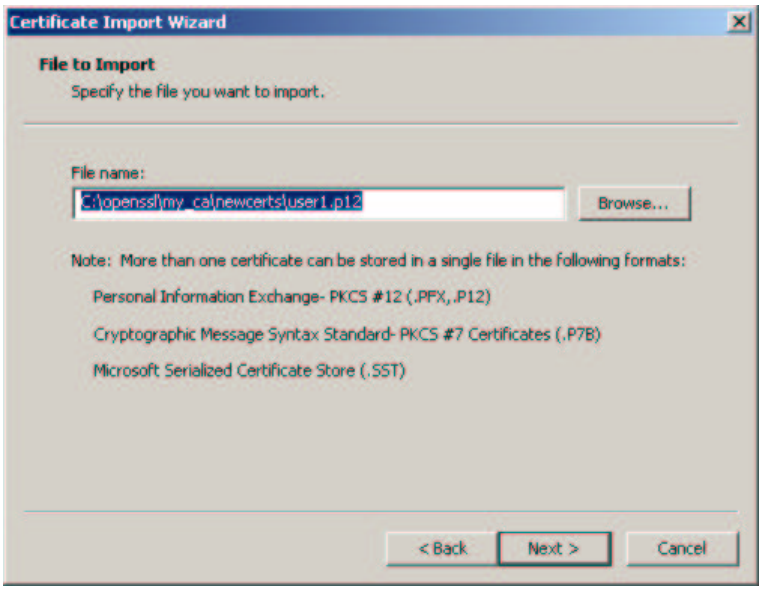
Converting certificates from PEM to DER or PKCS#12, to make them Microsoft ready.

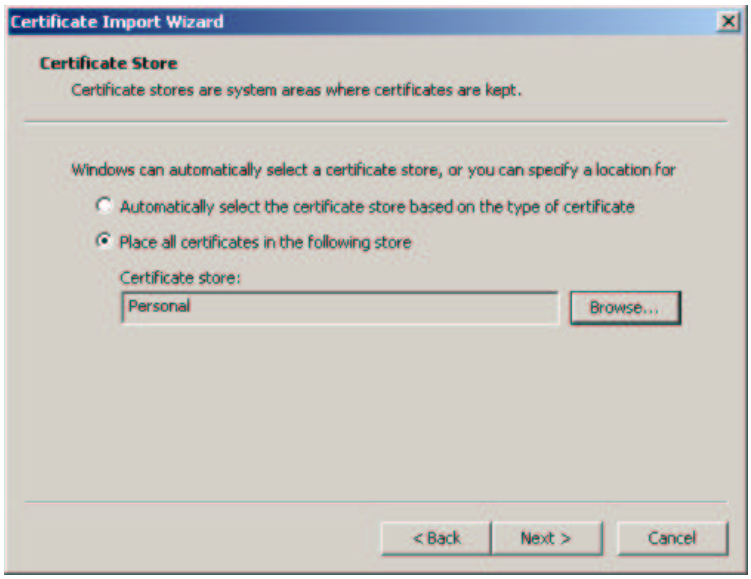
```
C:\openssl>openssl pkcs12 -export -in ./my_ca/newcerts/user1.pem -inkey
./my_ca/private/user1.key -out ./my_ca/newcerts/user1.p12
Loading 'screen' into random state - done
Enter PEM pass phrase:   (to unlock user1.key)
Enter Export Password: (to protect PKCS#12 file)
Verifying password - Enter Export Password:
```

Do the same thing for a user2.

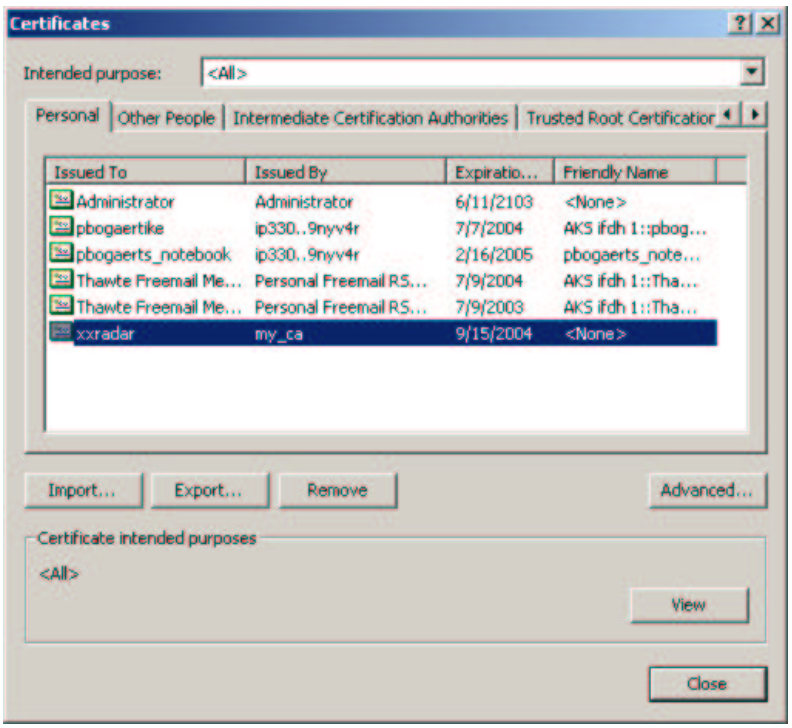
Installing the certificates in your email client.

Once you have an email account for a user, install the certificate in the personal certificate store, by double clicking the PKCS#12 file.





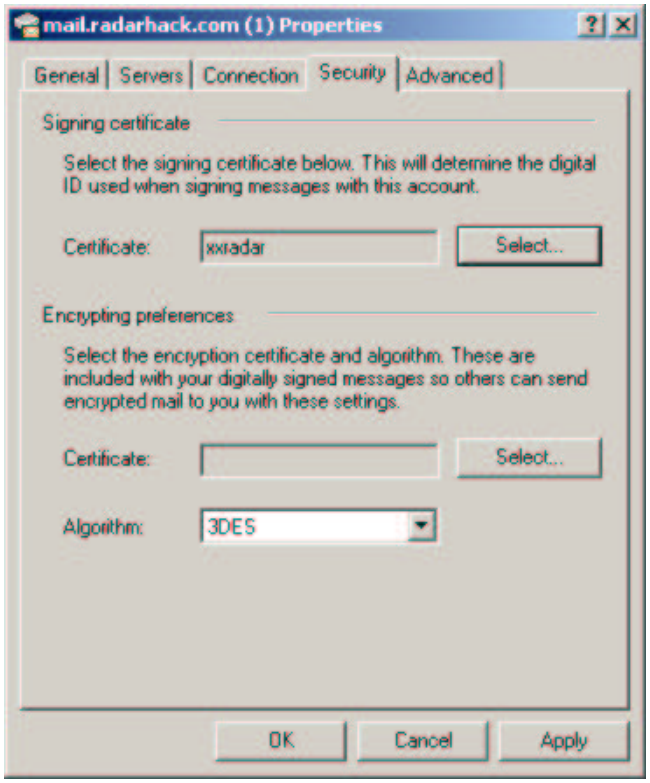
Verify the certificate by checking IE.



Double click the certificate and take a look.

You can check the certification path as shown.

Last but not least, use the certificate for encryption/signing on the mail account.



Remember to-do the same thing for a second user. You can create a second email account in the same outlook client, if you have two email accounts available. HAVE FUN!