Konstrukcja urzędów certyfikacji standardu OpenSSL, zarządzanie certyfikatami

1 Wprowadzenie

Technologia SSL(*Secure Sockets Layer*) zaproponowana przez firmę Netscape Communications na potrzeby szyfrowania i uwierzytelniania komunikacji została bardzo dobrze przyjęta przez wszystkich zainteresowanych. Obecnie technologia SSL jest szeroko wykorzystywana jako mechanizm bezpiecznej komunikacji nie tylko w obrębie protokołu HTTP ale służy również do tworzenia sieci VPN*(Virtual Private Network*). Najnowsza wersja protokołu SSL to wersja 3.0. U podstaw protokołu SSL leży praktyczne wykorzystanie kryptografii asymetrycznej oraz symetrycznej. Celem ćwiczenia jest zrozumienie działania protokołu SSL oraz poznanie pakietu OpenSSL.

Słowa kluczowe: ssl, openssl, PKI, certyfikat cyfrowy, vpn

2 Protokół SSL

Zadaniem protokołu SSL jest zapewnienie szyfrowanego i/lub uwierzytelnionego kanału przesyłania danych. Biorąc pod uwagę warstwowość sieciowego modelu ISO/OSI protokół SSL można umiejscowić powyżej protokołu TCP czyli warstwy transportowej ale poniżej warstw wyższych. Protokół SSL w imieniu warstw wyższych modelu ISO/OSI wykorzystuje protokół TCP do ustanowienia kanału wymiany danych. Analizując działania protokołu SSL można wyróżnić kilka zdarzeń, które mogą zajść w trakcie korzystania z protokołu SSL. Są to:

- Uwierzytelnienie serwera wspierającego SSL – strona będąca klientem połączenia SSL ma możliwość zweryfikowania tożsamości strony serwera poprzez wykorzystanie mechanizmów kryptografii klucza publicznego i sprawdzenie czy certyfikat strony serwera jest poprawny oraz czy został podpisany przez znaną klientowi zaufaną stronę trzecia, centrum certyfikacji(*CA, ang. Certification Authority*).
- Uwierzytelnienie klienta wspierającego SSL – strona będąca serwerem ma możliwość zweryfikować tożsamość klienta żądającego nawiązania bezpiecznego połączenia. Serwer przeprowadza proces weryfikacji klienta identycznie jak zostało to opisane powyżej.
- Nawiązanie szyfrowanego połączenia – obie strony połączenia dążą do nawiązania bezpiecznego połączenia którym mogą być przesyłane poufne dane.

Aby lepiej zrozumieć proces nawiązania połączenia, poniżej wymieniono najważniejsze etapy zachodzące w protokole SSL:

1. Serwer uwierzytelniania się przed klientem.
2. Klient i serwer wymieniają między sobą informacje pozwalające uzgodnić im

najsilniejsze mechanizmy kryptograficzne jakie obie strony wspierają np. algorytm funkcji skrótu kryptograficznego, algorytm służący do szyfrowania.

3. Jeżeli zachodzi taka potrzeba klient uwierzytelnia się przed serwerem.
4. Klient i serwer używając mechanizmów kryptografii klucza publicznego uzgadniają między sobą tajny klucz określany mianem *shared secret.*
5. Zostaje ustanowiony bezpieczny kanał wymiany danych między obiema stronami.

2.1 Etap nawiązania sesji SSL

Pierwszym krokiem w celu utworzenia bezpiecznego kanału wymiany danych między serwerem a klientem jest etap przywitania(*ang. handshake*) w którym strona serwera uwierzytelnia się przed stroną klienta oraz opcjonalnie strona klienta uwierzytelnia się przed stroną serwera. Oba procesy uwierzytelnia wykorzystują mechanizmy kryptografii klucza publicznego aby w wiarygodny sposób zweryfikować tożsamość drugiej strony. Poniżej zaprezentowano kolejne etapy w procesie przywitania między stroną serwera i klienta.

1. Strona klienta wysyła do strony serwera numer obsługiwanej wersji protokołu SSL, możliwe do użycia algorytmy szyfrujące oraz zestaw danych losowych.
2. Strona serwera wysyła do strony klienta numer obsługiwanej wersji protokołu SSL, możliwe do użycia algorytmy szyfrujące, zestaw danych losowych oraz własny certyfikat zawierający publiczny klucz serwera. Może również zdarzyć się, że serwer zażąda aby klient uwierzytelnił się. W takiej sytuacji serwer wysyła również żądanie o certyfikat strony klienta.
3. Strona klienta przystępuje do weryfikacji tożsamości strony serwera. Jeżeli wystąpi problem z weryfikacją tożsamości strony serwera użytkownik jest o tym informowany.
4. Po pomyślnym uwierzytelnieniu strony serwera, strona klienta generuje tajny ciąg znaków określany mianem *premaster secret*. Dysponując certyfikatem strony serwera, strona klienta używa klucza publicznego strony serwera, szyfruje tajny ciąg *premaster secret* i wysyła go do strony serwera.
5. Opcjonalnie, serwer może zażądać aby strona klienta uwierzytelniła się. Strona klienta wysyła do strony serwera swój certyfikat z kluczem publicznym oraz podpisany swoim kluczem prywatnym losowy ciąg znaków znany stronie klienta i serwera.
6. Strona serwera przystępuje do weryfikacji tożsamości strony klienta. Jeżeli wystąpi problem z weryfikacją sesja SSL jest przerywana. Jeżeli weryfikacja zakończy się pozytywnie strona serwera używając swojego klucza prywatnego odszyfrowuje ciąg *premaster secret* i na jego podstawie generuje tajny ciąg znaków określany mianem *master secret*(strona klienta również na podstawie ciągu *premaster secret* generuje ciąg master secret).
7. Obie strony połączenia na podstawie tajnego ciągu znaków *master secret* generują ciąg *session key.* Tajny ciąg *session key* posłuży jako klucz dla algorytmu symetrycznego szyfrowania służącego do szyfrowania komunikacji między stronami.
8. Obie strony połączenia informują siebie wzajemnie, że od tej chwili komunikacja między nimi będzie szyfrowana.
9. Proces ustanowienia sesji SSL zostaje zakończony, bezpieczny kanał komunikacji został ustanowiony.

3 Infrastruktura klucza publicznego i certyfikaty cyfrowe

Protokół SSL wykorzystuje dwie technologie odnośnie mechanizmów szyfrowania. Jest to kryptografia symetryczna i kryptografia asymetryczna. Kryptografia symetryczna służy do szyfrowania danych wymienianych między stronami. Kryptografia asymetryczna służy do uzgodnienia między stronami tajnego klucza sesyjnego. Połączenie obu technologii pozwala na bezpieczne korzystanie z protokołu SSL. Omawiając protokół SSL należy również wspomnieć o Infrastrukturze klucza publicznego(*ang. PKI, Public Key Infrastructure*)oraz certyfikatach standardu X.509.

PKI to koncepcja rozbudowanego systemu kryptograficznego w którym sieć wzajemnych powiązań między elementami tego systemu gwarantuje bezpieczeństwo komunikacji a konkretnie bezpieczeństwo i możliwość weryfikacji informacji o użytkownikach tego systemu. W skład PKI wchodzą:

- Urzędy certyfikacji(*ang CA, Certification Authority*) odpowiedzialne za poświadczanie tożsamości klientów, tzw. zaufana strona trzecia(*ang. trusted third party*).
- Użytkownicy dla których wydawane są certyfikaty.
- Urzędy rejestracji(*ang. RA, Registration Authority*) odpowiedzialne za weryfikację danych o użytkownikach oraz ich rejestrację.
- Oprogramowanie oraz sprzęt niezbędny do posługiwania się certyfikatami.
- Repozytoria kluczy, certyfikatów i listy odwołanych certyfikatów(*ang. CRL, Certificate Revocation List*)

Struktura PKI przypomina drzewo, w którym w roli korzenia występuje główny urząd certyfikacji, który określa politykę certyfikacji np. dla danego kraju. Poniżej znajdują się podległe urzędy certyfikacji zajmujące się obsługą np. poszczególnych grup użytkowników lub świadczące usługi certyfikacji dla określonych zastosowań. Na samym dole tego drzewa jako liście występują użytkownicy, którzy zgłaszają się do urzędów certyfikacji w celu wydania im certyfikatów. Instytucja certyfikująca wydaje certyfikat i poświadcza tożsamość użytkownika któremu wydaje certyfikat. Użytkownik może sprawdzić jaki urząd wydał dany certyfikat, może również sprawdzić całą ścieżkę zaufania od danego użytkownika do głównego urzędu certyfikacji. Cała struktura drzewiasta tworzy hierarchię zaufania. Należy zwrócić uwagę iż w certyfikacie znajduje się tylko klucz publiczny użytkownika. Klucz prywatny(tajny), który jest zawsze matematycznie powiązany z kluczem publicznym jest znany tylko użytkownikowi. Urząd certyfikacji wydając certyfikat poświadcza, że podpis wykonany tajnym kluczem użytkownika został złożony przez tego użytkownika.

Struktura PKI podlega standaryzacji. Standard opiera się na dwóch elementach. Jednym jest standard X.509 opisujący strukturę certyfikatów oraz drugi standard określany mianem PKCS(ang. Public Key Cryptography Standards) którego autorem jest firma RSA Data Security.

Certyfikat cyfrowy jest informacją elektroniczną poświadczającą związek między danym użytkownikiem a kluczem, którego używa dany użytkownik do procedury podpisywania dokumentów. Standard X.509 definiuje budowę takiego certyfikatu i wskazuje jakie elementy powinny być zawarte w certyfikacie. Są to m.in.:

- numer wersji standardu X.509

- numer wystawionego certyfikatu
- identyfikator algorytmu podpisu certyfikatu
- nazwa funkcji skrótu kryptograficznego użyta przez wystawcę
- nazwa wystawcy certyfikatu
- daty ważności certyfikatu
- nazwa podmiotu dla którego wystawiany jest certyfikat
- parametry klucza publicznego podmiotu
- klucz publiczny podmiotu
- opcjonalne rozszerzenia
- podpis wystawcy certyfikatu

3.1 Zarządzanie PKI

Z zarządzaniem infrastrukturą PKI wiąże się kilka zadań jakie muszą być możliwe w realizacji. Są to:

- Rejestracja użytkowników(*ang. registration*) - użytkownicy składają wnioski o wydanie certyfikatów.
- Wydawanie certyfikatów(*ang. certification*) - urzędy certyfikacyjne wydają certyfikaty dla użytkowników.
- Generowanie par kluczy(*ang. key generation*) - para kluczy(publiczny i prywatny) może zostać wygenerowana przez urząd certyfikacji. Klucz publiczny zostanie później podpisany a prywatny zostanie dostarczony klientowi w bezpieczny sposób. Bezpieczniejszym rozwiązaniem jest samodzielna generacja pary kluczy i dostarczenie urzędowi do podpisu tylko publicznego klucza.
- Wzajemna certyfikacja(*ang. cross certification*) – sytuacja, gdy urzędy certyfikacji wystawiają sobie wzajemnie certyfikaty podnosząc tym samym poziom zaufania do siebie oraz co równie ważne pozwala to tworzyć relacje zaufania między użytkownikami mającymi wydane certyfikaty przez różne urzędu certyfikacji.
- Odnawianie certyfikatów(*ang. certificates update*) – urząd certyfikacji wydaje certyfikaty dla użytkowników na określony czas. Po upływie wyznaczonego okresu certyfikat przestaje być ważny i należy ponownie zwrócić się do urzędu po nowy certyfikat.
- Unieważnianie certyfikatów(*ang. revocation certificates*) Certyfikat jest wydawany na określony czas. Czasami może zdarzyć się wyjątkowa sytuacja, gdy certyfikat musi być unieważniony przed upływem tego czasu. Takie wyjątkowe sytuacje to np. ujawnienie klucza prywatnego właściciela certyfikatu, kompromitacja urzędu certyfikacji, który wydał dany certyfikat(w takiej sytuacji wszystkie wydane certyfikaty przez taki urząd muszą zostać unieważnione), zmiana informacji o użytkowniku, która powoduje, że dane w aktualnym certyfikacie będą nieaktualne.
- Odzyskiwanie klucza(*ang. key recovery*) Niekiedy może się zdarzyć sytuacja, że

użytkownik zgubi swoje klucze. Jeśli klucze były przechowywane przez urząd certyfikacji to użytkownik będzie mógł je odzyskać.

4 Oprogramowanie OpenSSL

Oprogramowanie Openssl jest ogólnie dostępną biblioteką funkcji kryptograficznych. Openssl zawiera popularną implementację protokołu SSL. Najczęściej można ją spotkać w systemach Unix/Linux, gdzie dużo oprogramowania korzysta z tej biblioteki. Openssl to nie tylko biblioteka funkcji użytecznych dla wymagającego obsługi protokołu SSL oprogramowania. Użytkownicy za pomocą tej biblioteki mogą również samodzielnie tworzyć część infrastruktury klucza publicznego poprzez utworzenie własnego centrum certyfikacji, generowanie par kluczy oraz wystawianie certyfikatów. W dalszej części opracowania zostanie pokazane w jaki sposób można utworzyć własne centrum certyfikacji, wygenerować prośbę o wystawienie certyfikatu oraz jak wystawić certyfikat.

Wraz z biblioteką Openssl dostarczany jest zestaw użytecznych skryptów, które ułatwiają realizację wyżej wymienionych zadań.

4.1 Tworzenie własnego centrum certyfikacji

Do tworzenia własnego centrum certyfikacji można wykorzystać skrypt CA.sh lub CA.pl. Oba skrypty zapewniają identyczną funkcjonalność z tym, że pierwszy został napisany w języku interpretera bash a drugi w języku perl. Aby wygenerować utworzyć własne centrum certyfikacji należy wydać komendę:

```
./CA.sh -newca
```

Poniżej zaprezentowano przykład utworzenie centrum certyfikacji:

```
testowy:/usr/share/ssl/misc # ./CA.sh -newca
CA certificate filename (or enter to create)


Making CA certificate ...
Generating a 1024 bit RSA private key
........++++++
..................++++++
writing new private key to './demoCA/private/./cakey.pem'
Enter PEM pass phrase:
Verifying - Enter PEM pass phrase:
phrase is too short, needs to be at least 4 chars
Enter PEM pass phrase:
```

```
Verifying - 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 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]:PL
State or Province Name (full name) [Some-State]:WLKP
Locality Name (eg, city) []:Poznań
Organization Name (eg, company) [Internet Widgits Pty
Ltd]:Politechnika Poznańska
Organizational Unit Name (eg, section) []:Instytut Informatyki
Common Name (eg, YOUR name) []:ca_localhost
Email Address []:ca_admin@localhost.pl
testowy:/usr/share/ssl/misc #
```

Pogrubionym drukiem zaznaczono przykładowy tekst, który został wpisany przez użytkownika. Utworzenie centrum certyfikacji spowoduje utworzenie podkatalogu demoCA. W tym katalogu będzie znajdował się plik **cacert.pem,** który jest certyfikatem dla utworzonego urzędu certyfikacji. W podkatalogu **demoCA**/**private**/ będzie znajdował się klucz prywatny urzędu certyfikacji, plik **cakey.pem**. Podczas procesu tworzenia centrum certyfikacji skrypt zapyta użytkownika o frazę hasłową, która będzie użyta do zaszyfrowania klucza prywatnego centrum certyfikacji przechowywanego w pliku **cakey.pem**. UWAGA! Zgubienie tego klucza spowoduje niemożność wystawiania certyfikatów przez to centrum certyfikacji.

4.2 Generowanie prośby o wystawienie certyfikatu

Aby uzyskać certyfikat od urzędu certyfikatu należy najpierw zgłosić żądanie wystawienia certyfikatu. Polecenie:

```
./CA.sh -newreq
```

wygeneruje prośbę o wystawienie certyfikatu w formie pliku tekstowego. Poniżej zaprezentowano przykład generacji żądania o wystawienie certyfikatu:

```
testowy:/usr/share/ssl/misc # ./CA.sh -newreq
```

```
Generating a 1024 bit RSA private key
.................++++++
.++++++
writing new private key to 'newreq.pem'
Enter PEM pass phrase:
Verifying - 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]:PL
State or Province Name (full name) [Some-State]:WLKP
Locality Name (eg, city) []:Poznań
Organization Name (eg, company) [Internet Widgits Pty Ltd]:FIRMA X
Organizational Unit Name (eg, section) []:Dział Y
Common Name (eg, YOUR name) []:nazwa_serwera_firmy_X
Email Address []:admin@firma_x.pl

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
An optional company name []:
Request (and private key) is in newreq.pem
testowy:/usr/share/ssl/misc #
```

W pliku **newreq.pem** będzie znajdowało się żądanie wydania certyfikatu oraz klucz prywatny podmiotu żądającego wystawienia certyfikatu.

4.3 Wystawienie certyfikatu

Aby wystawić certyfikat na podstawie żądania wystawienie certyfikatu należy wydać polecenie:

```
./CA.sh -sign
```

Poniżej zaprezentowano przebieg procedury wystawienia certyfikatu:

```
testowy:/usr/share/ssl/misc # ./CA.sh -sign
Using configuration from /etc/ssl/openssl.cnf
Enter pass phrase for ./demoCA/private/cakey.pem:
Check that the request matches the signature
Signature ok
Certificate Details:
        Serial Number: 1 (0x1)
        Validity
            Not Before: Jul 15 09:10:00 2006 GMT
            Not After : Jul 15 09:10:00 2007 GMT
        Subject:
            countryName               = PL
            stateOrProvinceName       = WLKP
            localityName              = Pozna\C5\84
            organizationName          = FIRMA X
            organizationalUnitName    = Dzia\C5\82 Y
            commonName                = nazwa_serwera_firmy_X
            emailAddress              = admin@firma_x.pl
        X509v3 extensions:
            X509v3 Basic Constraints:
                CA:FALSE
            Netscape Comment:
                OpenSSL Generated Certificate
            X509v3 Subject Key Identifier:

12:B8:BF:4D:BF:B9:40:C9:2E:C1:7A:DE:A8:4B:75:58:D8:C6:CD:3D
            X509v3 Authority Key Identifier:

keyid:BF:62:BF:05:7C:05:30:E5:E3:7A:90:82:0C:61:45:60:7E:D8:F3:49
```

```
DirName:/C=PL/ST=WLKP/L=Pozna\xC5\x84/O=Politechnika
Pozna\xC5\x84ska/OU=Instytut
Informatyki/CN=ca_localhost/emailAddress=ca_admin@localhost.pl
                serial:8D:A7:61:FC:60:94:39:C1


Certificate is to be certified until Jul 15 09:10:00 2007 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
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number: 1 (0x1)
        Signature Algorithm: md5WithRSAEncryption
        Issuer: C=PL, ST=WLKP, L=Pozna\xC5\x84, O=Politechnika
Pozna\xC5\x84ska, OU=Instytut Informatyki,
CN=ca_localhost/emailAddress=ca_admin@localhost.pl
        Validity
            Not Before: Jul 15 09:10:00 2006 GMT
            Not After : Jul 15 09:10:00 2007 GMT
        Subject: C=PL, ST=WLKP, L=Pozna\xC5\x84, O=FIRMA X,
OU=Dzia\xC5\x82 Y,
CN=nazwa_serwera_firmy_X/emailAddress=admin@firma_x.pl
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
            RSA Public Key: (1024 bit)
                Modulus (1024 bit):
                    00:b8:b1:2b:00:8b:de:2a:bf:93:ca:e4:a9:ac:2b:
                    7a:be:b5:3e:0c:ca:24:55:f7:65:6b:66:fa:34:d2:
                    0c:96:76:bb:01:f0:2b:24:27:99:65:11:78:51:f9:
                    6b:a7:96:ee:b7:98:45:99:46:ed:1e:13:c9:bb:ab:
                    99:c1:59:0e:ac:b8:89:9f:6a:11:b4:04:97:66:7b:
                    92:c1:19:c1:82:90:5e:df:c1:85:96:e7:9b:44:d3:
                    ee:67:8a:c1:e9:e3:14:a2:ef:0c:13:39:c0:55:19:
                    8a:3b:25:aa:49:b9:1e:ab:c4:1d:81:56:90:cc:76:
```

```
                    89:b1:ea:d8:40:e7:c7:0b:4f
                Exponent: 65537 (0x10001)
        X509v3 extensions:
            X509v3 Basic Constraints:
                CA:FALSE
            Netscape Comment:
                OpenSSL Generated Certificate
            X509v3 Subject Key Identifier:

12:B8:BF:4D:BF:B9:40:C9:2E:C1:7A:DE:A8:4B:75:58:D8:C6:CD:3D
            X509v3 Authority Key Identifier:

keyid:BF:62:BF:05:7C:05:30:E5:E3:7A:90:82:0C:61:45:60:7E:D8:F3:49

DirName:/C=PL/ST=WLKP/L=Pozna\xC5\x84/O=Politechnika
Pozna\xC5\x84ska/OU=Instytut
Informatyki/CN=ca_localhost/emailAddress=ca_admin@localhost.pl
                serial:8D:A7:61:FC:60:94:39:C1


    Signature Algorithm: md5WithRSAEncryption
        b2:bb:a6:c5:cf:5b:c7:25:42:dd:49:6d:34:f7:a0:8f:33:1d:
        2a:3f:6a:12:62:b6:48:d3:c2:1a:4a:d6:5b:14:4e:8e:98:86:
        e9:86:94:57:14:92:1a:38:46:76:0d:7f:8c:82:a0:c8:99:9c:
        60:03:f7:5a:f8:46:d7:f5:07:7a:97:19:46:5c:b8:f0:e8:ce:
        e6:0b:cd:fd:4d:22:91:28:83:af:a1:4a:b4:51:eb:bb:7d:1a:
        47:b3:8d:b5:f3:ba:dc:75:a9:5a:7d:02:43:ae:e4:37:8c:8d:
        b4:47:70:b4:86:2a:a3:d2:94:86:23:d8:8d:43:db:18:a8:1b:
        c1:39
-----BEGIN CERTIFICATE-----
MIID+jCCA2OgAwIBAgIBATANBgkqhkiG9w0BAQQFADCBrDELMAkGA1UEBhMCUEwx
DTALBgNVBAgTBFdMS1AxEDAOBgNVBAcUB1Bvem5hxYQxIDAeBgNVBAoUF1BvbGl0
ZWNobmlrYSBQb3puYcWEc2thMR0wGwYDVQQLExRJbnN0eXR1dCBJbmZvcm1hdHlr
aTEVMBMGA1UEAxQMY2FfbG9jYWxob3N0MSQwIgYJKoZIhvcNAQkBFhVjYV9hZG1p
bkBsb2NhbGhvc3QucGwwHhcNMDYwNzE1MDkxMDAwWhcNMDcwNzE1MDkxMDAwWjCB
lDELMAkGA1UEBhMCUEwxDTALBgNVBAgTBFdMS1AxEDAOBgNVBAcUB1Bvem5hxYQx
EDAOBgNVBAoTB0ZJUk1BIFgxETAPBgNVBAsUCER6aWHFgiBZMR4wHAYDVQQDFBVu
YXp3YV9zZXJ3ZXJhX2Zpcm15X1gxHzAdBgkqhkiG9w0BCQEWEGFkbWluQGZpcm1h
X3gucGwwgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBALixKwCL3iq/k8rkqawr
er61PgzKJFX3ZWtm+jTSDJZ2uwHwKyQnmWUReFH5a6eW7reYRZlG7R4TyburmcFZ
```

```
Dqy4iZ9qEbQEl2Z7ksEZwYKQXt/BhZbnm0TT7meKwenjFKLvDBM5wFUZijslqkm5
HqvEHYFWkMx2ibHq2EDnxwtPAgMBAAGjggFAMIIBPDAJBgNVHRMEAjAAMCwGCWCG
SAGG+EIBDQQfFh1PcGVuU1NMIEdlbmVyYXRlZCBDZXJ0aWZpY2F0ZTAdBgNVHQ4E
FgQUEri/Tb+5QMkuwXreqEt1WNjGzT0wgeEGA1UdIwSB2TCB1oAUv2K/BXwFMOXj
epCCDGFFYH7Y80mhgbKkga8wgawxCzAJBgNVBAYTAlBMMQ0wCwYDVQQIEwRXTEtQ
MRAwDgYDVQQHFAdQb3puYcWEMSAwHgYDVQQKFBdQb2xpdGVjaG5pa2EgUG96bmHF
hHNrYTEdMBsGA1UECxMUSW5zdHl0dXQgSW5mb3JtYXR5a2kxFTATBgNVBAMUDGNh
X2xvY2FsaG9zdDEkMCIGCSqGSIb3DQEJARYVY2FfYWRtaW5AbG9jYWxob3N0LnBs
ggkAjadh/GCUOcEwDQYJKoZIhvcNAQEEBQADgYEAsrumxc9bxyVC3UltNPegjzMd
Kj9qEmK2SNPCGkrWWxROjpiG6YaUVxSSGjhGdg1/jIKgyJmcYAP3WvhG1/UHepcZ
Rly48OjO5gvN/U0ikSiDr6FKtFHru30aR7ONtfO63HWpWn0CQ67kN4yNtEdwtIYq
o9KUhiPYjUPbGKgbwTk=
-----END CERTIFICATE-----
Signed certificate is in newcert.pem
testowy:/usr/share/ssl/misc #
```

Po zakończeniu procedury wystawienia certyfikatu w pliku **newcert.pem** znajduje się wystawiony certyfikat.

5 Podsumowanie

Biblioteka Openssl jest szeroko stosowanym oprogramowaniem, które często jest elementem obowiązkowym innych programów. Niewątpliwą zaletą tej biblioteki jest również możliwość tworzenia certyfikatów w przyjazny dla użytkownika sposób, co pozwala ja wykorzystanie wsparcia dla SSL np. w serwerach www np. serwer Apache.

6 Zadania

- Utworzenie własnego centrum certyfikacji przy pomocy biblioteki Openssl.
- Wygenerowanie żądania wystawienia certyfikatu
- Wystawienie certyfikatu
- Weryfikacja informacji o urzędzie certyfikacji i podmiocie żądającym certyfikatu za pomocą poleceń c_issuer, c_info i c_name.

7 Problemy do dyskusji

- Zalety i wady PKI

- Jakie informacje powinny znaleźć się w certyfikatach cyfrowych?
- Jakie usługi sieciowe korzystają z wsparcia biblioteki Openssl?
- Jakie cechy kryptografii publicznej i prywatnej są wykorzystywane w protokole SSL?

8 Bibliografia

- Strona projektu Openssl http://www.openssl.org
- Strona firmy SUN poświęcona protokołowi SSL http://docs.sun.com/source/816-6156-10/contents.htm#1041643
- Strona centrum certyfikacji signet http://www.signet.pl/pomoc/pki.html