Pro úspěšné provedení příkladu je nutné mít

OpenSSL

Doporučuje se ještě

Total Commander

Vytvoření certifikátu kořenové certifikační autority – self-signed

openssl req -new -x509 -days 365 -key ca.key -out ca.crt

Country Name (2 letter code) [AU]:CZ

State or Province Name (full name) [Some-State]:Morava

Locality Name (eg, city) []:Brno

Organization Name (eg, company) [Internet Widgits Pty Ltd]:VUTBR

Organizational Unit Name (eg, section) []:VUTBR

Common Name (e.g. server FQDN or YOUR name) []:KIB

Email Address []:ibreiten@fit.vutbr.cz

Vytvoření požadavku pro získání certifikátu – opět self-signed, abychom dokázali, že vlastníme privátní klíč

openssl req -new -key srv.key -out srv.csr

Country Name (2 letter code) [AU]:CZ

State or Province Name (full name) [Some-State]:Brno

Locality Name (eg, city) []:Brno

Organization Name (eg, company) [Internet Widgits Pty Ltd]:VUT FP KIB

Organizational Unit Name (eg, section) []:

Common Name (e.g. server FQDN or YOUR name) []:kib.fp.vut.cz

Email Address []:ibreiten@fit.vutbr.cz

Please enter the following 'extra' attributes

to be sent with your certificate request

A challenge password []:

An optional company name []:

Vytvoření a podepsání certifikátu certifikační autoritou

openssl x509 -req -days 365 -in srv.csr -CA ca.crt -CAkey ca.key -set_serial 01 -out srv.crt

---

Klient chce bezpečně komunikovat se serverem, iniciuje tedy komunikaci a žádá server o certifikát.

Server zasílá certifikát. (Zkopírování certifikátu do klientské složky)

Klient kontroluje důvěryhodnost certifikátu vůči Certifikační autoritě

openssl verify -verbose -CAfile ca.crt srv.crt

Klient nyní serveru důvěřuje. Vytvoří tak heslo pro symetrickou šifru, pomocí kterého pak bude šifrována zbylá komunikace.

openssl dgst -md5 -hex key-aes.txt > key.aes

Klient zašifruje zprávu symetrickou šifrou

openssl enc -aes-128-cbc -pass pass:key.aes -in msg.txt -out secret.enc

---

Klient chce zprávu předat serveru. Musí tak extrahovat veřejný klíč z certifikátu

openssl x509 -pubkey -noout -in srv.crt > srv-pub.key

Když má klient klíč, může šifrovat

openssl rsautl -encrypt -in key.aes -inkey srv-pub.key -pubin -out encryptedMsg1.rsa

openssl rsautl -encrypt -in secret.enc -inkey srv-pub.key -pubin -out encryptedMsg2.rsa

Tato data pak posílá serveru

Server data přijímá a nejdříve dešifruje zprávu 1 pro získání hesla, pak zprávu 2 pro získání zprávy zašifrovanou symetrickou šifrou

openssl rsautl -decrypt -in encryptedMsg1.rsa -inkey srv.key -out key.aes

openssl rsautl -decrypt -in encryptedMsg2.rsa -inkey srv.key -out secret.enc

---

Server úspěšně dešifroval data šifrovaná asymetrickou šifrou, zbývá tak dešifrovat zprávu šifrovanou symetrickou šifrou

openssl enc -d -aes-128-cbc -pass pass:key.aes -in secret.enc -out msg.txt

Odeslání a příjem dat tak proběhl v pořádku

Nyní server chce klientu potvrdit, že data přijal a vše je v pořádku, vytvoří tak potvrzující zprávu pro klienta a tu podepíše

openssl dgst -sha1 -sign srv.key -out resp.sign resp.txt

zpráva a podpis jsou odeslány klientovi

Klient přijímá odpověď a kontroluje, zda není podvržená

openssl dgst -sha1 -verify srv-pub.key -signature resp.sign resp.txt

---

Klient a server provedli bezpečnou komunikaci, kdy využili jak symetrickou, tak asymetrickou kryptografii.