OpenSSL a certifikáty

Petr Krčmář

1. června 2013

OpenSSL: o čem to bude

- Co je OpenSSL?
- Co všechno umí?
- Příklady, příklady, příklady

ROOT.CZ

OpenSSL je švýcarský nůž?

ROOT.CZ

Knihovna a utilita

- www.openssl.org
- především knihovna napsaná v C
- implementuje kryptografické funkce, protokoly SSL a TLS
- pomocí wrapperů použitelná v mnoha jazycích
- také řádková utilita
- pro použití ve skriptech a přímo na řádce

ROOT.CZ

Co OpenSSL umí?

- jednoduché benchmarky
- symetricky (de)šifrovat soubory
- hashovat soubory (MD5, SHA. . .)
- zpracovávat kódování Base64
- připojit se k serveru po SSL
- spravovat PKI certifikáty

ROOT.CZ

Benchmarky

- velmi jednoduchý benchmark
- měří rychlost jednotlivých algoritmů
- kolik zvládne procesor
- volá se parametrem `speed`
- měří počet bloků zpracovaných za tři sekundy
- mění velikost bloků od 16 až po 8192
- nakonec vypíše, kolik jich zpracoval za sekundu

Příklad

```
$ openssl speed sha1
```

ROOT.CZ

(De)šifrování souborů

- jednoduché zpracování *jednoho* souboru
- více souborů je potřeba třeba zatarovat
- použitý je symetrický (rychlý) algoritmus
- vše je chráněno jedním heslem
- na něm závisí bezpečnost šifry
- různé šifry: des3, bf, cast, des, idea...

Příklad

```
$ openssl des3 -in soubor -out zasifrovany_soubor
$ openssl des3 -d -in zasifrovany_soubor -out soubor
```

ROOT.CZ

Hashování souborů

- v zásadě stejné jako šifrování
- jen se použije jiný algoritmus
- opět mnoho variant: md2, md5, sha, sha1, sha256, sha512...
- alternativou jsou md5sum a sha*sum z coreutils

Příklad

```
$ openssl md5 soubor
$ openssl sha1 soubor
```

ROOT.CZ

Base64

- převod binárních dat do tisknutelného textu
- o třetinu větší výstup
- používá se v XML, mailu…
- výstup se skládá maximálně ze 64 různých znaků
- 8bitová slova se spojí dohromady a rozsekají po 6 bitech
- tím vznikne 64 kombinací (A-Z, a-z, 0-9, +, /)
- konec je zarovnán pomocí = na tři znaky)

Příklad

```
$ openssl base64 -in soubor -out zakodovano
$ openssl base64 -d -in zakodovano -out soubor
```

ROOT.CZ

SSL klient

- pro ladění se obvykle používá telnet (nc)
- co ale když máme port za SSL?
- OpenSSL k tomu má **interaktivního**
- existuje i s_server (ukážeme později)

Příklad

```
$ openssl s_client -connect forum.debian-linux.cz:443
GET / HTTP/1.1
Host: forum.debian-linux.cz
```

- pro STARTTLS je třeba přidat parametr starttls

Příklad

```
$ openssl s_client -connect mail.iinfo.cz:587 \
  -starttls smtp
```

Tvorba certifikátů

- především pro servery či poštu
- tvorba vlastního (self signed) certu
- žádost o podepsání autoritou
- tvorba vlastní autority
- potvrzení autoritou

Vytvoření self-signed certifikátu

```
$ openssl req -x509 -nodes -days 365 -newkey rsa:1024 \
  -keyout certifikat.pem -out certifikat.pem
$ openssl verify certifikat.pem
```

- vytvoří se klíč i certifikát, v jednom souboru .pem

ROOT.CZ

Vytvoření žádosti pro CA

- žádost předáme autoritě
- ta nám ji podepíše a vrátí
- musíme vygenerovat *dva soubory*: klíč a žádost

Vytvoření žádosti

```
$ openssl req -new -newkey rsa:1024 -nodes \
  -keyout klic.key -out zadost.csr
```

- pokud už klíč máme, stačí jednodušší syntaxe

Vytvoření žádosti

```
$ openssl req -new -key klic.key -out zadost.csr
```

Vypsání informací z žádosti

- můžeme si nechat vypsat informace z žádost

Příklad vypsání informací

```
$ openssl req -in zadost.csr -noout -text
```

- můžeme si nechat zkontrolovat podpis žádosti

Kontrola podpisu žádosti

```
$ openssl req -in zadost.csr -noout -verify \
  -key klic.key
```

ROOT.CZ

Vlastní certifikační autorita

- umožňuje vám generovat centrálně podepsané certy
- vhodné pro organizace, které nechtějí platit jiné CA
- do klientů pak stačí naimportovat kořenový certifikát CA
- je pak možné podle interních směrnic vydávat certifikáty

ROOT.CZ

Příklad tvorby vlastní CA (1/2)

- vytvoříme si adresářovou strukturu

Příklad

```
$ mkdir mojeCA mojeCA/private \
  mojeCA/certs mojeCA/newcerts mojeCA/crl
```

- zkopírujeme si výchozí konfigurační soubor

Příklad

```
$ cp /etc/ssl/openssl.cnf /tmp/mojeCA
```

- zeditujeme ho a upravíme (nezapomenout na req_distinguished_name)

ROOT.CZ

Příklad tvorby vlastní CA (2/2)

- vytvoříme si databázi pro OpenSSL
- zadáme pořadové číslo prvního certu

Příklad

```
$ touch /tmp/mojeCA/index.txt
$ echo 01 > /tmp/mojeCA/serial
```

ROOT.CZ

Generování klíče a kořenového certifikátu

- vytvoříme klíč a certifkát pro naši CA
- platnost bude pět let

Příklad

```
$ openssl req -config openssl.cnf -new -x509 \
  -extensions v3_ca -keyout private/mojeCA.key \
  -out certs/mojeCA.crt -days 1825
```

ROOT.CZ

Podpis žádosti o vydání certifikátu

- podepíšeme pomocí své CA

Příklad

```
$ openssl ca -config openssl.conf -batch -notext \
-keyfile private/mojeCA.key -cert certs/mojeCA.crt \
-in ../zadost.csr -out ../podepsana.cer
```

- pokud potřebujeme PEM (pro další příklad)

Příklad

```
$ cat klic.key podepsana.cer > muj_server.pem
```

ROOT.CZ

SSL server

- opak s_client
- umožňuje emulovat druhou stranu
- je potřeba si nejprve vygenerovat certifikát
- ten už máme :-)
- server se otevře na **https**://127.0.0.1:4433

Příklad

```
$ openssl s_server -cert certifikat.pem -www
```

- pro předání klíčů web serveru si přečtěte manuál
- různé servery chtějí součásti v různých souborech

ROOT.CZ

Děkuji za pozornost

Dotazy?

ROOT.CZ