OpenSSL Mini-Tutorial

Dr. Günter Kolousek

2016-02-13

1. Installation unter Linux
   - Ubuntu, Debian

     `aptitude install openssl`
   - archlinux, Manjaro

     `pacman -S openssl`
2. Version feststellen

   ```
   openssl version
   openssl version -a
   ```
3. Dokumentation, Links
   - `http://openssl.org`
   - `https://www.feistyduck.com/library/openssl-cookbook/`
   - `https://wiki.archlinux.org/index.php/OpenSSL`
   - `https://www.digitalocean.com/community/tutorials/openssl-essentials-working-with-ssl-c`
   - `https://pki-tutorial.readthedocs.org/en/latest/`
   - `http://users.dcc.uchile.cl/~pcamacho/tutorial/crypto/openssl/openssl_intro.html`
4. Welche Kommandos gibt es? -> Ungültiges eingeben…

   `openssl -h`

   oder

   `man openssl`
5. Welche Subkommandos gibt es? -> Ungültiges…

   `openssl dgst -h`
6. Einen digest berechnen lassen

   `openssl dgst -sha256 mozilla.pdf`
   - sha = sha-0 -> kurz danach ersetzt durch sha-1 (160Bit)
     - sha-0 … erfolgreiche Angriffe bekannt
     - sha-1 … theoretische Angriff möglich
   - sha-2 … Familie: sha-224, sha-256, sha-384, sha-512
     - keine Angriffe bekannt
   - sha-3 … Ausschreibung von NIST
     - soll sha-2 nicht ersetzen, sondern ergänzen

7. Wie ein selbst-signiertes Zertifikat in einem Schritt generieren?

   `openssl req -x509 -nodes -days 365 -newkey rsa:1024 -keyout key.pem -out cert.pem`

   - `req` … certificate signing request
   - `-x509` … erzeuge x509 anstatt Zertifikatsrequest
     - ITU-T (internat telecommunication union) Standard für PK-Kryptographie
     - x500: nie implementierter Standard für Verzeichnisdienste (s.LDAP)
   - `-nodes` … Key nicht verschlüsseln
   - `-days 365` … 365 Tage gültig
   - `-newkey rsa:1024` … neuer Key mittels RSA mit Länge 1024 Bits
   - Was ist Unterschied zwischen .csr, .crt, .pem? → kein Unterschied im Format: alle PEM (ASCII, base64 kodiert)
     - `.csr`: Zertifizierungsanfrage
     - `.crt`: Zertifikat
     - `.pem`: kann Zertifikat und/oder privaten Key enthalten

8. (optional) nur wenn von Verisign… zu signieren

   `openssl req -new -newkey rsa:1024 -nodes -keyout key.pem -out req.pem`

   Wenn Key schon vorhanden, dann:

   `openssl req -new -key key.pem -out req.csr`

   Kann folgendermaßen überprüft werden:

   `openssl req -in req.pem -noout -verify -key key.pem`

   Info ansehen:

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

9. Wie ein Zertifikat verifizieren?

   `openssl verify cert.pem`

   Error 18 ist ok…

10. Wie testen?

    `openssl s_server -cert cert.pem -key key.pem -www`

    dann Browser starten und zu dieser URL: `https://testing:4433`

11. Wie ein entferntes Zertifikat mit der Kommandozeile abrufen?

    `openssl s_client -connect google.at:443 > google.pem`

    Dann alles außer BEGIN CERTIFICATE…END CERTIFICATE

12. Wie Informationen aus Zertifikat extrahieren?

    `openssl x509 -text -in google.pem`

    - Wie auf Einzelteile zugreifen?

      `openssl x509 -noout -in google.pem -issuer`

      oder hinten: `-subject`, `-dates`, `-hash`, `-fingerprint`

13. Wie einen private RSA Schlüssel generieren?

    `openssl genrsa -out key.pem 1024`

14. Wie einen public RSA Schlüssel generieren?

    ```
    openssl rsa -in key.pem -pubout > pubkey.pem
    ```

15. Wie ein Zertifikat erstellen?

    ```
    openssl req -x509 -new -nodes -days 365 -key key.pem -out cert.pem
    ```

16. Wie digest signieren/verifizieren?
    - generieren ohne signieren:

      ```
      openssl dgst -sha256 -out nginx.sha256 ../nginx-1.4.1.tar.gz
      ```

    - generieren mit signieren:

      ```
      openssl dgst -sha256 -sign key.pem -out nginx.sha256 ../nginx-1.4.1.tar.gz
      ```

    - verifizieren eines signierten digest:

      ```
      openssl dgst -sha256 -verify pubkey.pem -signature nginx.sha256 ../nginx-1.4.1.tar.gz
      ```

17. Wie eine Datei base64 kodieren/dekodieren?

    ```
    openssl enc -base64 -in ../nginx-1.4.1.tar.gz > nginx-1.4.1.tar.gz.enc
    less *.enc
    openssl enc -d -base64 -in nginx-1.4.1.tar.gz.enc -out nginx-1.4.1.tar.gz
    diff ../nginx-1.4.1.tar.gz nginx-1.4.1.tar.gz
    ```

18. Verschlüsseln/Entschlüsseln

    ```
    openssl enc -aes-256-cbc -salt -in nginx.sha256 -out nginx.sha256.crypt
    openssl enc -d -aes-256-cbc -salt -in nginx.sha256.crypt -out nginx.sha256.2
    diff nginx.sha256.crypt nginx.sha256.2
    ```