Server-Zertifikatsinstallation unter LINUX/UNIX

Beispielhaft sei hier die Verwendung eines Server-Zertifikates unter einem Apache2.0-Webserver erläutert.

Für den sicheren Zugriff bietet sich das sichere HTTPs-Protokoll an. Dort, wo sensible Daten, wie z.B. ein Login-Paßwort verwendet wird, sollte die Verbindung über das Netz grundsätzlich verschlüsselt erfolgen. Deshalb bieten Grundinstallationen häufig die Möglichkeit an, mit selbstsignierten Zertifikaten zu arbeiten, nur um die technischen Voraussetzungen zu schaffen, die den Einsatz von SSL- bzw. TSL-Verbindungen zu ermöglichen.

Das folgende Beispiel basiert auf dem einseitigen Zertifikatsaustausch, bei dem der Webserver dem Browser des Anwenders ein Zertifikat anbietet, das bei erfolgreicher Überprüfung bzw. Akzeptanz durch den Client-Browser zu einem sicheren Verbindungsaufbau führt. Meist ist im weiteren Zugriff auf Daten bzw. Anwendungen ein Passwort-Dialog zur Autorisierung zu finden,mit dem dann, unabhängig vom Client-Zertifikat, schliesslich der Zugang gewährt oder abgelehnt wird. OpenSSL-Installation

OpenSSL installieren Sie entsprechend des mitgelieferten READMEs und des entspr. Windows-Setup bzw. unter Unix durch Installation des entsprechenden Ports bzw. Pakets aus den verschiedenen Distributionen oder durch Kompilierung des Quellcodes:

```
>./configure
>make
>make install
```

Nach erfolgreicher Installation finden Sie im Verzeichnis /usr/local/ssl die installierte Software. Die erfolgreiche Installation testen Sie beispielsweise mit *$ openssl* und anschliessender Eingabe *Help*

hier der Output

```
Standard commands

asn1parse      ca             ciphers        crl            crl2pkcs7
dgst           dh             dhparam        dsa            dsaparam
enc            engine         errstr         gendh          gendsa
genrsa         nseq           ocsp           passwd         pkcs12
pkcs7          pkcs8          prime          rand           req
rsa            rsautl         s_client       s_server       s_time
sess_id        smime          speed          spkac          verify
```

```
version         x509

Message Digest commands (see the `dgst' command for more details)

md2             md4             md5             rmd160          sha
sha1

Cipher commands (see the `enc' command for more details)

aes-128-cbc     aes-128-ecb     aes-192-cbc     aes-192-ecb     aes-256-cbc
aes-256-ecb     base64          bf              bf-cbc          bf-cfb
bf-ecb          bf-ofb          cast            cast-cbc        cast5-cbc
cast5-cfb       cast5-ecb       cast5-ofb       des             des-cbc
des-cfb         des-ecb         des-ede         des-ede-cbc     des-ede-cfb
des-ede-ofb     des-ede3        des-ede3-cbc    des-ede3-cfb    des-ede3-ofb
des-ofb         des3            desx            rc2             rc2-40-cbc
rc2-64-cbc      rc2-cbc         rc2-cfb         rc2-ecb         rc2-ofb
```

Zertifikatserstellung

Zertifikatsanträge müssen in einem maschinell verarbeitbaren Format bei der RA eingereicht werden, dem Public Key Cryptography Standard Nr 10, PKCS#10, und werden bei Verwendung eines geeigneten Werkzeuges automatisch erstellt.

Der Zertifikatsantrag setzt sich aus 4 Teilen zusammen:

1. dem „Subject Distinguished Name“, kurz DN, die teilweise durch die Policy vorgegeben sind, wie Landesbezeichnung, Country, C=de; die Ortsbezeichnung O=Technische Universitaet Clausthal; die OU-Attribute, z.B. OU=Rechenzentrum
2. den optionalen Attributen
3. einer digitalen Signatur mit dem privaten Schlüssel (der später auch auf dem Server abgelegt wird)
4. einer Kennzeichnung des verwendeten Signaturalgorithmus

Bei Verwendung folgender Konfigurationsdatei für OpenSSL sind diese Angaben bereits voreingestellt und können übernommen werden:

Den Wert bei *0.organizationalUnitName_default* (als Vorgabe steht dort *Rechenzentrum*) sollten Sie in der Konfigurationsdatei in den Namen Ihres Instituts bzw. Ihrer Einrichtung ändern!

rzkey.conf

```
#
```

```
# rzkey.conf
#
# Verwenden Sie diese Datei als Konfigurationsdatei fuer
Zertifikatsantraege.
# Das typische Kommando zur Erstellung eines solchen Server-
Zertifikatantrags
# lautet:
#
# openssl req -config rzkey.conf -newkey rsa:4096 -sha256 -outform PEM
-out certreq.pem
#
#
# This definition stops the following lines choking if HOME isn't
# defined.
HOME            = .
RANDFILE        = $ENV::HOME/.rnd



#####################################################################
[ req ]
default_bits        = 4096
default_keyfile     = server-key.pem
distinguished_name = req_distinguished_name
attributes      = req_attributes



# This sets a mask for permitted string types. There are several
options.
# default: PrintableString, T61String, BMPString.
# pkix   : PrintableString, BMPString.
# utf8only: only UTF8Strings.
# nombstr : PrintableString, T61String (no BMPStrings or UTF8Strings).
# MASK:XXXX a literal mask value.
# WARNING: current versions of Netscape crash on BMPStrings or
UTF8Strings
# so use this option with caution!
string_mask = nombstr

# req_extensions = v3_req # The extensions to add to a certificate
request

[ req_distinguished_name ]
countryName             = Laendername (bitte nicht aendern)
countryName_default         = DE
countryName_min             = 2
countryName_max             = 2
```

```
stateOrProvinceName             = Bundesland (bitte nicht aendern)
stateOrProvinceName_default     = Niedersachsen

localityName                    = Ortsname (bitte nicht aendern)
localityName_default            = Clausthal

.organizationName     = Name der Organisation (bitte nicht aendern)
.organizationName_default = Technische Universitaet Clausthal


.organizationalUnitName       = Abteilung
.organizationalUnitName_default   = Rechenzentrum

1.organizationalUnitName       = Unterabteilung
1.organizationalUnitName_default   =

2.organizationalUnitName       = Gruppe
2.organizationalUnitName_default   =

commonName         = Voll qualifizierter DNS-Name des Servers, unter
dem die Website angesprochen werden soll
commonName_max         = 64

emailAddress           = Email-Adresse des Betreibers (Beispiel:
webmaster@tu-clausthal.de)
emailAddress_max       = 40
emailAddress_default       = webmaster@tu-clausthal.de

# SET-ex3           = SET extension number 3

[ req_attributes ]

# unstructuredName      = An optional company name


[ v3_req ]

# Extensions to add to a certificate request

basicConstraints = CA:FALSE
keyUsage = nonRepudiation, digitalSignature, keyEncipherment
```

Wenn Sie ein Zertifikat für einen Windows-Dienst benötigen, finden Sie hier eine Anleitung.

Falls Sie Ihren Dienst bisher ohne Zertifikat betreiben, müssen Sie ein entsprechendes Schlüsselpaar nach dem RSA-Verfahren erzeugen. Das folgende openssl-Kommando führt die Schlüsselgenerierung,

die Abfrage eines Passwortes für diesen Schlüssel, die Attribut-Abfrage und die anschliessende Erstellung des Zertifikatantrages durch:

> openssl req -config rzkey.conf -newkey rsa:4096 -sha256 -outform pem -out certreq.pem

Die einzelnen Befehlsteile bedeuten folgendes:

1. Programmname des Openssl-Kommandos, evtl. muss der komplette Pfad, z.B. /usr/local/bin/openssl angegeben werden.
2. req: erstellt einen PKCS#10-Antrag
3. -config: verweist auf die ssl-Konfig-Datei für Voreinstellungen
4. -newkey rsa:4096: neues Schlüsselpaar mit RSA und einer Key-Länge von 4096 Bit
5. sha256: „Secure Hash Algorithm Nr. 2“ wird als Streumechanismus für die Erstellung kryptographisch sicherer Prüfsummen verwendet.
6. -outform PEM: Privacy Enhanced Mail Ausgabeformat
7. -out certreq.pem: speichert den Antrag unter certreq.pem. Achtung: die Konfigurationsdatei enthält die Direktive, den neu zu erstellenden privaten Schlüssel in server-key.pem zu speichern. Sollen mehrere Anträge hintereinander erstellt werden, müssen die Dateien nach jedem Durchgang umbenannt werden, um ein Überschreiben zu vermeiden!

Läuft der Dienst mit einem selbst signierten Zertifikat und haben Sie bereits einen RSA-Schlüssel erstellt mit einer Länge von 4096 Bit, können Sie diesen Schlüssel für Ihr neues Zertifikat weiter verwenden. Mit folgendem Kommando kann dann eine Zertifizierungsanfrage für den vorhandenen Key (server.key) erstellt werden:

> openssl -req -config rzkey.conf -key server.key -keyform PEM -sha256 -outform pem -out certreq.pem

Mit -keyform PEM bzw. -keyform DER sollten Sie das Format des vorhandenen Schlüssels berücksichtigen.

Laut Heise: „SHA-1-Zertifikate sollen nicht sofort, sondern im Laufe der nächsten zwei bis drei Jahre aus dem Netz verbannt werden. Um den Server-Betreibern einen Anreiz zum Wechsel auf ein moderneres Hash-Verfahren wie SHA-256 zu schaffen, soll Chrome künftig seine Nutzer auf SHA-1-Sites ein spezielles Symbol anzeigen, dass über das niedrigere Sicherheitsniveau informiert. Kommt ein SHA-1-Zertifikat zum Einsatz, das noch nach dem 1. Januar 2017 gültig ist, verschärft Chrome die Maßnahmen und betrachtet die vom Server ausgelieferten Inhalte als Mixed Content. Bindet eine HTTPS-Site, die alles richtig macht, Inhalte von dem betroffenen Server ein, zeigt Chrome eine Warnung an – genau so, als würde die Site HTTP-Inhalte einbetten. Wann genau diese Änderungen zum Tragen kommen, ist noch nicht bekannt.“

In den FAQs der DFN-PKI ist sind zu dem Thema „Fragen und Antworten zur Umstellung von SHA-1 auf SHA-2 in der DFN-PKI“ ausführliche Hinweise zu finden.

Installation von Apache

Im wentlichen muss nur das Konfig-File *httpd.conf* (bei der Suche hilft ein *locate httpd.conf*) angepasst werden:

Als Gruppe sollten Sie eine group angeben, die für keinen anderen Dienst auf dem Server verwendet wird.

Ersetzen Sie die Mailadresse im Eintrag ServerAdmin durch die im Zertifikatsantrag angegebene administrative Emailadresse

```
ServerAdmin webmaster@.tu-clausthal.de
```

In der Konfigurationsdatei ssl.conf werden die Einstellungen für die https-Verbindungen vorgenommen. Auch hier sind wieder ähnliche Einstellungen wie in der httpd.conf vorzunehmen, allerdings sollte der Server auf dem Standard-Port 443 für verschlüsselte Verbindungen lauschen.

```
ServerName test.bla.tu-clausthal.de:443
```

Der Servername ist dejenige, welcher als DN im Zertifikatsantrag gesetzt wurde.

Suchen Sie den Eintrag SSLCertificateFile und ersetzen Sie die Pfadangabe durch den vollständigen Pfad auf das von der CA ausgestellte Server-Zertifikat.

```
SSLCertificateFile /usr/local/apache2/conf/ssl.crt/server.crt
```

Damit beim Start des Servers nicht jedesmal das Passwort eingegeben werden muss, kann mit

```
openssl rsa -in server.key -out server.key_nopassphrase
```

der Schlüssel dekodiert (von seiner Passphrase befreit) werden.

Suchen Sie den Eintrag SSLCertificateKeyFile und ersetzen Sie diesen Eintrag durch den vollständigen Pfad auf den dekodierten privaten RSA-Schlüssel, der bei der Erstellung des PKCS#10-Antrages generiert wurde.

```
SSLCertificateKeyFile
/usr/local/apache2/conf/ssl.key/server.key_nopassphrase
```

Laden Sie mit einem Browser die Zertifikatskette im „PEM“-Format herunter und speichern Sie diese Zertifikate in das Verzeichnis /usr/local/apache2/SSL/ssl.crt/. Dann weisen Sie den Apache-Webserver dazu an, diese Zertifikatskette zu verwenden:

```
SSLCertificateChainFile /usr/local/apache2/SSL/ssl.crt
```

In der Standardeinstallung der Apache-SSL-Konfiguration kann die Zeile SSLCipherSuite durch folgendes ersetzt werden:

```
SSLHonorCipherOrder on
SSLCipherSuite "HIGH:MEDIUM!aNULL !eNULL !LOW !3DES !MD5 !EXP !PSK !SRP !DSS
!RC4"
```

Hierdurch werden unsichere Authentifikations- und Verschlüsselungsmechanismen ausgeschlossen.

Browsertest

Mit den oben angegebenen Einstellungen sollte neben dem Zugriff per http auch ein verschlüsselter Zugriff möglich sein. Starten Sie den Server mit

>/usr/local/apache2/bin/apachectl startssl

damit das SSL/TSL-Protokoll aktiviert wird. In den Logfiles des Apache könne Sie ggfls. nachsehen, was einen erfolgreichen Start des Servers verhindert hat;)

Kommt die Verbindung mit dem Server zustande, können Sie das angebotene Zertifikat untersuchen. Es werden die enthaltenen CA-Informationen der TU-Clausthal-CA und die Attribute des Server-Zertifikates angezeigt. Falls die Identität des Servers als vertrauenswürdige Website nicht bestätigt werden sollte, haben Sie vergessen, in Ihren Browser die CA-Zertifikate zu importieren. Sie können dies über die TU Clausthal CA Einstiegsseite wiederholen.

Wenn Sie insbesondere bei der Konfiguration ihres Webservers die Option SSLCertificateChainFile benutzt haben, genügt es, das Root-Zertifikat der DFN-PKI zu importieren. Der Browser kann dann mit Hilfe der von Ihrem Server im Zuge des SSL/TLS-Verbindungsaufbaus an ihn gesendeten Zertifikate aus dem SSLCertificateChainFile auf die Authentizität Ihres Servers schließen. Es konnte ein sogenannter „Trusted Path“ zwischen Ihrem Server und der DFN-PKI hergestellt werden, obwohl die dazwischenliegenden Zertifikate der TU Clausthal CA nicht in den Browser importiert wurden.

Bei richtiger Konfiguration von Servern anderer Einrichtungen innerhalb der DFN-PKI, aber außerhalb der UHH würden sich die Browser dann auch nicht mehr mit der obigen Meldung beschweren, da auch zu ihnen bei Vorhandensein des Wurzelzertifikats der DFN-PKI eben dieser „Trusted Path“ vom Browser abgeleitet werden kann.

Probleme beim SSL/TSL-Aufbau können z.B. mit folgenden Unix-Kommandos untersucht werden:

>openssl s_client -host server1.rz.tu-clausthal.de -port 443

Die Ausgabe des OpenSSL-Kommandos mit „s-client“ enthält wichtige Hinweise:

- CONNECTED: Verbindungsaufbau wurde hergestellt
- Certificate chain:
- No client certificate CA names sent: einseitige Authentifizierung des Servers
- New,TLSv1/SSLv3,...
- Server public key is 2048 bit:

- Protocol: TLSv1

Tips und Links zu Apache- und OpenSSL-Dokus:

http://www.openssl.org

vollständige Apache-Dokumentation

An dieser Stelle einen großen Dank an die Mitarbeiter der UHH (Uni Hamburg), die dort die DFN-CA betreiben und an der Dokumentation ihres Dienstes gearbeitet haben. Struktur und ein großer Teil der Inhalte dieser Seite ist von dort übernommen worden.

Linux