

Beantragung von Serverzertifikaten an der Universität Passau

Serverzertifikate werden ab 01.01.2023 nur noch über die GÉANT Trusted Certificate Services (TCS) verteilt. Hierzu steht ein Webportal bereit, welches von Nutzern im Selfservice-Verfahren betrieben wird.

Zuvor muss offline ggf. ein privater Schlüssel sowie ein Certificate Signing Request (CSR) erzeugt werden. Dies wird hier am Beispiel der OpenSSL-Software erklärt. Wenn Sie eine andere Software verwenden, konsultieren Sie bitte die Dokumentation Ihrer Software.

Wichtige Information: Das Zertifikatsfeld „OU“ kann nach wie vor im Zertifikatsantrag ausgefüllt werden, wird aber nicht mehr in das endgültige Zertifikat übernommen.

Beantragen eines Serverzertifikats mit eigenem Zertifikatsantrag (CSR)

1. Erzeugung eines private Key

Dies ist nur notwendig, wenn für den betreffenden Server noch kein entsprechender Key (RSA, mindestens 2048 Bit Länge) vorliegt.

```
openssl genrsa -out <secretkeyfile> <laenge>
```

Hierbei ist <secretkeyfile> der Name der Datei, in die der private Key geschrieben wird, <laenge> die gewünschte Schlüssellänge (mindestens 2048).

2. Erzeugung des CSR

Der Zertifikatsantrag (CSR) muss im PKCS#10-Format als PEM-kodierte Datei vorliegen und kann mit folgendem Kommando erzeugt werden:

```
openssl req -new -key <secretkeyfile> -out <csrfile> \  
-subj <subjectstring>
```

<secretkeyfile> ist der Name der Datei, in der sich der private Key bereits befinden muß. <csrfile> ist der Name der Datei, in die der CSR geschrieben wird.

<subjectstring> ist der zu zertifizierende Name in folgendem Format (bitte für das OpenSSL-Kommando in Anführungszeichen einschließen):

```
/C=DE/ST=Bayern/L=Passau/O=Universitaet Passau/OU=<einrichtungsname>/CN=<servername>
```

Hierbei sollte für <einrichtungsname> die Bezeichnung der Einrichtung angegeben werden, die den Server betreibt, <servername> ist der Name des Servers als FQDN. <servername> muss auf ".uni-passau.de" enden! Wir behalten uns vor, den Einrichtungsnamen zwecks Standardisierung ggf. anzupassen. Die anderen Teile des <subjectstring> dürfen nicht variiert werden.

Bitte beachten Sie, dass Umlaute im <subjectstring> nicht zugelassen sind. Bitte verwenden Sie ggf. die übliche Substitution:

Ä → Ae, Ö → Oe, Ü → Ue, ä → ae, ö → oe, ü → ue, ß → ss.

Wie kann man einen CSR für mehrere FQDNs erzeugen?

Verwenden Sie die OpenSSL-Konfigurationsdatei "uni-passau-req.cnf", die Sie unter folgendem Link downloaden können:

```
http://www.rz.uni-passau.de/fileadmin/Dateien/Dokumente/Rank/uni-passau-req.cnf
```

Bitte passen Sie die Einträge für "OU" und "CN" für Ihren Server an. Mit den Einträgen "DNS.2", "DNS.3", ... können Sie weitere FQDNs angeben, für die das Zertifikat gelten soll.

Bitte beachten Sie, dass der bei "CN" angegebene Servername unbedingt auch in "DNS.1" angegeben werden muss. Nicht benötigte "DNS."-Einträge löschen Sie bitte.

Den CSR können Sie dann folgendermaßen erzeugen:

```
openssl req -new -key <secretkeyfile> -out <csrfile> \
-config uni-passau-req.cnf
```

3. Online-Beantragung des Zertifikats

Gehen Sie mit einem Webbrowser auf die Seite

```
http://www.zim.uni-passau.de/dienstleistungen/netzwerk-und-server/server-zertifizierungen/online-antrag/
```

Sie gelangen dann über einen weiteren Link auf die Einstiegsseite der Nutzerschnittstelle für die Beantragung von Zertifikaten:

Welcome to SSL Certificate Management

Before enrolling or managing existing certificates you must authenticate.

Identity Provider

You can select to authenticate with your company's identity provider.

Your Institution

-  Why do I need to authenticate?
-  How do I use my passphrase?
-  How do I revoke my certificate?

Wählen Sie den Punkt "Your Institution" und suchen Sie nach "uni-passau.de". Wählen Sie das Suchergebnis "University of Passau".

Find Your Institution

Your university, organization or company



Examples: Science Institute, Lee@uni.edu, UCLA



Remember this choice

[Learn More](#)

University Of Passau

uni-passau.de

Sollten Sie noch kein Zertifikat bezogen haben, werden Sie mit über ein Formular zur Beantragung eines neuen Zertifikates aufgefordert. Hier ist folgender Bereich entscheidend und muss von Ihnen ausgefüllt werden:

 Upload CSR

CSR *

Common Name

Subject Alternative Names

External Requesters

“Upload CSR” können Sie einen erstellten Zertifikatsantrag als Textdatei hochladen. Alternativ können Sie den Inhalt der Textdatei auch manuell in das Feld “CSR” kopieren. Der Antrag wird automatisch validiert und die Felder “Common Name” und “Subject Alternative Names” werden mit den Angaben des Antrags befüllt.

Sie können optional “External Requesters” mit hinterlegen (E-Mail Adressen). Diese werden automatisch auch durch die Zertifizierungsstelle benachrichtigt, falls Aktionen notwendig werden (z.B. Abholung oder Erneuerung).

Sobald alle Angaben erfolgt sind schicken Sie den Antrag mit “Submit” ab. Sobald der Antrag von einem Mitarbeiter bestätigt wird, erhalten Sie das Zertifikat und alle notwendigen Dateien (Root-CA, Zertifikatskette, etc.) per E-Mail von “support@cert-manager.com”. Alle in der Mail enthaltenen Links verweisen wieder zum Portal “<https://cert-manager.com/customer/DFN>”. In der Liste erhalten Sie eine Auswahl an Formaten, die je nach verwendeter Software benötigt werden könnten.

Please choose the correct download format. For Apache/nginx use "as Ce:

Sobald
oder

Available formats:

- as Certificate only, PEM encoded: <https://cert-manager.com/customer/DFN/ssl?action=download&format=PEM>
- as Certificate (w/ issuer after), PEM encoded: <https://cert-manager.com/customer/DFN/ssl?action=download&format=PEM&issuer=after>
- as Certificate (w/ chain), PEM encoded: <https://cert-manager.com/customer/DFN/ssl?action=download&format=PEM&chain=true>
- as PKCS#7: <https://cert-manager.com/customer/DFN/ssl?action=download&format=PKCS7>
- as PKCS#7, PEM encoded: <https://cert-manager.com/customer/DFN/ssl?action=download&format=PKCS7&format=PEM>

Sie ein
mehrere

Issuing CA certificates only:

- as Root/Intermediate(s) only, PEM encoded: [https://cert-manager.com/customer/DFN/ssl?action=download&format=PEM&type=Root/Intermediate\(s\)only](https://cert-manager.com/customer/DFN/ssl?action=download&format=PEM&type=Root/Intermediate(s)only)
- as Intermediate(s)/Root only, PEM encoded: [https://cert-manager.com/customer/DFN/ssl?action=download&format=PEM&type=Intermediate\(s\)/Rootonly](https://cert-manager.com/customer/DFN/ssl?action=download&format=PEM&type=Intermediate(s)/Rootonly)

Sincerely Yours,
DFN-PKI-Team

Zertifikate unter Ihrer Verwaltung besitzen, finden Sie im gleichen Portal eine Liste Ihrer Zertifikate. Hier können Sie jederzeit das Zertifikat erneut herunterladen, erneuern, oder widerrufen.

Konfigurationshinweise

Unter <https://ssl-config.mozilla.org> können Sie bei Bedarf Konfigurationsschnipsel für eine valide SSL Konfiguration beziehen. Sie benötigen lediglich das Wissen über den zu verwendenden Webserver, sowie das gewünschte Kompatibilitätsniveau. Beispiel „nginx“:

Server Software

- Apache
- AWS ALB
- AWS ELB
- Caddy
- Dovecot
- Exim
- Go
- HAProxy
- Jetty
- lighttpd
- MySQL
- nginx
- Oracle HTTP
- Postfix
- PostgreSQL
- ProFTPD
- Redis
- Squid
- Tomcat
- Traefik

Mozilla Configuration

- Modern
Services with clients that support TLS 1.3 and don't need backward compatibility
- Intermediate
General-purpose servers with a variety of clients, recommended for almost all systems
- Old
Compatible with a number of very old clients, and should be used only as a last resort

Environment

Server Version 1.17.7

OpenSSL Version 1.1.1k

Miscellaneous

 HTTP Strict Transport Security

This also redirects to HTTPS, if possible

 OCSP Stapling

nginx 1.17.7, intermediate config, OpenSSL 1.1.1k

Supports Firefox 27, Android 4.4.2, Chrome 31, Edge, IE 11 on Windows 7, Java 8u31, OpenSSL 1.0.1, Opera 20, and Safari 9

```
# generated 2022-12-21, Mozilla Guideline v5.6, nginx 1.17.7, OpenSSL 1.1.1k, intermediate configuration
# https://ssl-config.mozilla.org/#server=nginx&version=1.17.7&config=intermediate&openssl=1.1.1k&guideline=5.6
server {
    listen 80 default_server;
    listen [::]:80 default_server;

    location / {
        return 301 https://$host$request_uri;
    }
}

server {
    listen 443 ssl http2;
    listen [::]:443 ssl http2;

    ssl_certificate /path/to/signed_cert_plus_intermediates;
    ssl_certificate_key /path/to/private_key;
    ssl_session_timeout 1d;
    ssl_session_cache shared:MozSSL:10m; # about 40000 sessions
    ssl_session_tickets off;

    # curl https://ssl-config.mozilla.org/ffdhe2048.txt > /path/to/dhparam
    ssl_dhparam /path/to/dhparam;

    # intermediate configuration
    ssl_protocols TLSv1.2 TLSv1.3;
    ssl_ciphers ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-RSA-AES128-GCM-SHA256:ECDHE-ECDSA-AES256-GCM-SHA384:ECDHE-RSA-AES256-GCM-SHA384:ECDHE-ECDSA-CHACHA20-POLY1305:ECDHE-RSA-CHACHA20-POLY1305:DHE-RSA-AES128-GCM-SHA256:DHE-RSA-AES256-GCM-SHA384;
    ssl_prefer_server_ciphers off;

    # HSTS (ngx_http_headers_module is required) (63072000 seconds)
    add_header Strict-Transport-Security "max-age=63072000" always;
```