# Abschaltung von SSL/TLS Versionen vor 1.2

*Veröffentlichung: IV-Sicherheitsteam, 10.04.2019*

*Zielgruppe: Administratoren der WWU*

## Einleitung

Das TLS-Protokoll (Transport Layer Security) dient der Sicherstellung von *Vertraulichkeit*, *Authentizität* und *Integrität* bei der Übertragung von Daten in unsicheren Netzwerken. Es sorgt im Internet u.a. in Form des HTTPS Protokolls für den verschlüsselten Seitenabruf. Daneben ist TLS der wichtigste Standard für Authentifizierung im Internet. TLS 1.0 ist seit 19 Jahren im Einsatz. Die veralteten Versionen 1.0 und 1.1 setzen unter anderem auf die schon lange als unsicher geltenden Hash-Verfahren MD5 und SHA-1.

Seit 2011 sind mehrere Angriffe gegen SSL/TLS bekannt geworden. Daraus folgen die Beeinträchtigung der Vertraulichkeit, z.B. das Abfangen vertraulicher Daten durch Unbefugte sowie beim Verlust der Integrität der Daten, die Veränderung von Informationen oder das Einschleusen von schädlichen Daten. Die Schwachstellen werden durch Nutzung entsprechender Cipher-Suites in TLS 1.2 behoben. Zehn Jahre nach deren Veröffentlichung unterstützen fast alle Server und Clients die aktuelle Version TLS 1.2.

Das BSI hat bereits 2014 einen Migrationsplan beschlossen, wonach seit 01.07.2017 in allen Bundesbehörden nur noch TLS 1.2 (oder neuer) zum Einsatz kommen darf.

## Empfehlung

Das IV-Sicherheitsteam der WWU hat sich in der Sitzung vom 17.10.2018 einheitlich für die umgehende Abschaltung veralteter und unsicherer Verschlüsselungsprotokolle (TLS 1.0/1.1 sowie alle SSL Versionen) ausgesprochen.

Alle Server (z.B. Webserver, Mailserver) und Dienste (z.B. VPN) der WWU, die Daten über unsichere Netze (z.B. das Internet) übertragen, müssen mindestens den Verschlüsselungsstandard TLS 1.2 unterstützen. *In Zukunft dürfen Server, die kein TLS 1.2 oder neuer unterstützen nur noch im Intranet oder per VPN genutzt werden.*

Ab 2020 werden aktuelle Browser keine Unterstützung für TLS 1.0/1.1 mehr anbieten. D.h. spätestens dann müssen alle Dienste auch im Intranet TLS 1.2 unterstützen, damit sie weiter genutzt werden können.

## Migrationsplan

Ab 01.05.2019 schreibt das WWU-CERT Betreiber von aus dem Internet noch mit SSL v2/v3 erreichbaren Servern an. Falls innerhalb von 14 Tagen keine Maßnahmen zur Absicherung durchgeführt werden, wird der Whitelisteintrag für den betroffenen Port entfernt.

Ab 01.09.2019 schreibt das WWU-CERT Betreiber von aus dem Internet noch mit TLS v1.0/1.1 erreichbaren Servern an. Falls innerhalb von 14 Tagen keine Maßnahmen zur Absicherung durchgeführt werden, wird der Whitelisteintrag für den betroffenen Port entfernt.

## Weitere Informationen

- IETF will alte TLS-Versionen "verbieten"
    - https://heise.de/-4088705
- Verschlüsselung im Web: Chrome, Firefox & Co. verabschieden sich von TLS 1.0/1.1
    - https://heise.de/-4191864

- Mindeststandard des BSI für den Einsatz des SSL/TLS-Protokolls durch Bundesbehörden
  - https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Mindeststandards/Mindeststandard_BSI_TLS_1_2_Version_1_0.pdf?__blob=publicationFile&v=8
- Mit dem SSL Server Test von Qualys kann die Konfiguration von Web-Servern getestet werden
  - https://www.ssllabs.com/ssltest/
- Mit den Kommandozeilen-Programmen `nmap` oder `gnutls-cli-debug` können die unterstützten Verschlüsselungsverfahren von Servern getestet werden
  - `nmap --script ssl-enum-ciphers -p 443 www.uni-muenster.de`
  - `gnutls-cli-debug -p 443 www.uni-muenster.de`

# Anhang

1. Bei Apache-Webservern können mittels der folgenden beispielhaften Konfiguration Clienten, die kein TLS 1.2 unterstützen, auf eine passende Fehlerseite umgeleitet werden. Beispiel aus dem Webserverpark:

```
# Der Server akzeptiert dabei Verbindungen auch mit TLS 1.0 oder 1.1, schaut aber - noch vor einer eventuellen Passwortabfrage - nach der TLS-Protokollversion und leitet ggf. auf eine Fehlerseite um.

<VirtualHost ...>
  <If "%{REQUEST_URI} !~ m#^/-# && %{SSL_PROTOCOL}=~/^(TLSv1|TLSv1[.]1)$/">
    Require all denied
    ErrorDocument 403 /-/ErrOldTLS.html
  </If>
  ...
```

2. Für nginx-Webserver kann folgende Konfiguration genutzt werden, um eine entsprechende Fehlerseite bzw. Fehlermeldung beim Aufruf des Servers für nicht-unterstützte TLS-Versionen zu erzeugen:

```
server {
    ...
   if ($ssl_protocol ~ ^TLSv1([.]1)?$) {
        return 403 "$ssl_protocol is not supported!";
  }
  ...
}
```

3. Webserver, die kein TLS 1.2 unterstützen, können z.B. durch einen Reverse-Proxy (https://httpd.apache.org/docs/2.4/howto/reverse_proxy.html) abgesichert werden.