TLS und HTTPS

Wie ein Hoster jede Exponential-Website nur über HTTPS ausliefert - wer TLS beendet, Umleitung von einfachem HTTP, HSTS im Kernel und in Velocity, sichere Sitzungscookies, Proxys und Load Balancer, die TLS beenden…

Source file: `doc/ger-de/hilfe/hosting/sicherheit/tls-und-https.md` Last changed 2026-11-04
https://doc.exponential.earth/learninminutes/exphelp/viewer/help/hosting/security/tls-and-https.md

Diese Seite ist für Systemadministratoren, die Exponential-Websites über HTTPS ausliefern, und für Support-Mitarbeitende, die den üblichen Symptomen begegnen: Links mit `http://` hinter einem TLS-Proxy, Schleifen bei der Anmeldung oder eine Warnung des Browsers. Sie sagt, wer TLS beenden kann, welche Einstellungen von Exponential zu dieser Wahl passen müssen und wie Sie das Ergebnis prüfen. Die Referenz ist 13.9 Sicherheitsheader und HTTPS und 13.10 Hinter einem Proxy.

Inhalt

1. Wer TLS beendet
2. HTTPS überall, Schritt für Schritt
3. HSTS
4. Sichere Cookies
5. Proxys, die TLS beenden
6. Zertifikate überwachen
7. Prüfungen
8. Was schiefgeht
9. Checkliste

Wer TLS beendet

Einrichtung	TLS wird beendet von	Zertifikat verwaltet von
Apache mit PHP-FPM	Apache `mod_ssl`	dem Panel oder einem ACME-Client auf dem Server
nginx mit PHP-FPM	nginx `ssl`	dem Panel oder einem ACME-Client auf dem Server
Velocity direkt vor den Besuchern	Velocity selbst, mit HTTP/2	Velocity: ein Zertifikat, das Sie ihm geben, ein selbstsignierter Ersatz oder ACME (Modus `letsencrypt`)
Velocity oder PHP-FPM hinter einem Proxy, Load Balancer oder CDN	dem Proxy	dem Betreiber des Proxys

Velocity braucht für HTTPS kein `mod_ssl`, kein stunnel und keinen Proxy; es prüft ein Zertifikatspaar, bevor es es nutzt, tauscht ein erneuertes Paar innerhalb von `watchInterval` (60 Sekunden) im Betrieb aus und liefert lieber einen selbstsignierten Ersatz als ein kaputtes Paar aus (8.3.9). Details für Hoster: HTTPS-Zertifikate in Velocity; Zertifikate in Panels: TLS-Zertifikate in Panels.

HTTPS überall, Schritt für Schritt

1. **Liefern Sie jeden Namen der Website über HTTPS aus**, einschließlich `www.` und des Hostnamens der Administration.
2. **Leiten Sie einfaches HTTP auf HTTPS um**, in Apache oder nginx (Abschnitte 8.5 und 8.6 von Kapitel 8). Velocity lauscht auf beiden Ports und hat keine Einstellung, die einen auf den anderen umleitet; setzen Sie die Umleitung davor, oder schließen Sie den einfachen Port nach außen. Die eigenen `[SSLZoneSettings]` von Exponential in `site.ini` können einzelne Teilbäume umschalten, für eine Website, die noch nicht nur HTTPS sein kann.
3. **Lassen Sie die Website HTTPS-Adressen bauen**: `site.ini` `[SiteSettings]` `SiteURL` und die URLs der Siteaccesses.
4. **Setzen Sie** `[Session]` `CookieSecure=true` (siehe Sichere Cookies).
5. **Schalten Sie HSTS ein**, zuerst kurz (siehe HSTS).
6. **Überwachen Sie den Ablauf von Zertifikaten** (siehe Zertifikate überwachen).

HSTS

Ein Browser, der `Strict-Transport-Security` gesehen hat, verweigert für den ganzen Zeitraum einfaches HTTP zu diesem Host, selbst wenn Sie den Wert später senken. Beginnen Sie kurz, erhöhen Sie auf ein Jahr erst, wenn jeder Name dauerhaft über HTTPS ausgeliefert wird.

Wer ihn sendet	Einstellung	Ausgeliefert
Velocity, bei jeder HTTPS-Antwort	`velocity.ini [HTTPSSettings] HSTSMaxAge, HSTSIncludeSubDomains, HSTSPreload`	`300`, `disabled`, `disabled`
der Kernel, unter Apache oder nginx	`site.ini [HTTPHeaderSettings] SecurityHeaders[Strict-Transport-Security]`	leer (aus); nur über HTTPS gesendet

INI

```ini
# settings/override/site.ini.append.php, once every name is HTTPS for good
[HTTPHeaderSettings]
SecurityHeaders[Strict-Transport-Security]=max-age=31536000
```

Preload braucht `HSTSIncludeSubDomains` und mindestens ein Jahr (8.3.9, Strict-Transport-Security).

> **Warnung** `includeSubDomains` gilt für jede Subdomain des Namens, auch für solche, die andere Teams oder andere Kunden über einfaches HTTP betreiben. Vereinbaren Sie es mit dem Inhaber der Domain, bevor Sie es einschalten.

Sichere Cookies

`settings/site.ini` `[Session]`, mit den Standards, die gelten, solange die Schlüssel nicht gesetzt sind (13.7):

Schlüssel	Standard	Für eine Website nur mit HTTPS
`CookieSecure`	`auto`: Secure, wenn die Anfrage selbst über HTTPS kam	`true`
`CookieHttponly`	`true`	beibehalten
`CookieSameSite`	`Lax`	beibehalten; `None` nur mit `Secure`

INI

```ini
# settings/override/site.ini.append.php
[Session]
CookieSecure=true
```

`CookieSecure=true` stellt sicher, dass das Sitzungscookie nie über einfaches HTTP gesendet wird, selbst bei einer Anfrage, die fälschlich ohne TLS ankommt.

Proxys, die TLS beenden

Wenn ein Proxy TLS beendet, sieht PHP eine einfache HTTP-Anfrage und erfährt von HTTPS nur aus Headern wie `X-Forwarded-Proto`. Jeder Besucher kann diese Header auch senden, deshalb glaubt der Kernel ihnen seit 6.0.15 **nur, wenn die Verbindung von einer Adresse in `site.ini` `[HTTPHeaderSettings]` `TrustedProxies[]` kommt**. Der Standard sind die Loopback-Adressen `127.0.0.1` und `::1`.

Einrichtung	Was zu tun ist
Kein Proxy	nichts; weitergeleitete Header, die ein Besucher sendet, werden ignoriert
Ein Proxy auf derselben Maschine, der die Header setzt	nichts; lassen Sie den Proxy `X-Forwarded-Proto` selbst setzen
Ein Proxy, Load Balancer oder CDN auf einer anderen Adresse	führen Sie seine Adressen oder Bereiche auf, und nur diese; schirmen Sie den Anwendungsserver mit der Firewall ab, damit er nicht an ihnen vorbei erreichbar ist
Ein Frontend, das `REMOTE_ADDR` umschreibt (`mod_remoteip`, nginx `realip`, wie bei nginx vor Apache in Plesk)	nichts; `ClientIpByCustomHTTPHeader` bleibt `false`

INI

```ini
# settings/override/site.ini.append.php: a load balancer on a private subnet
[HTTPHeaderSettings]
TrustedProxies[]
TrustedProxies[]=10.0.0.0/16
ClientIpByCustomHTTPHeader=X-Forwarded-For
```

Die Adresse des Besuchers wird von rechts aus `X-Forwarded-For` gelesen, sodass ein gefälschter Eintrag weiter links nie genutzt wird. Einträge sind Adressen oder Bereiche; Hostnamen werden nicht aufgelöst, und ein Kommentar in derselben Zeile macht den Eintrag ungültig.

Unter Velocity führen Sie die Proxys in Velocitys eigenem `Q.webserver.proxy.trusted` auf, in einem Snippet unter `/etc/vc/conf-available/`, eingeschaltet mit `exp:velocity conf enable <name>`; Velocity gibt dann den Besucher als `REMOTE_ADDR` weiter und setzt

`$_SERVER['HTTPS']` selbst (ab Engine-Release 0.0.4.44 nur für vertrauenswürdige Gegenstellen). Details: 13.10, Unter Velocity und Weitergeleiteten Headern wird nur von konfigurierten Proxys vertraut.

Zertifikate überwachen

- Velocity: `php bin/php/console exp:velocity ssl show --json --allow-root-user`, und beobachten Sie `daysLeft`. Ein CA-Zertifikat mit 90 Tagen wird erneuert, wenn etwa 30 Tage übrig sind; weniger als 14 übrige Tage bedeuten also, dass die Erneuerung seit mehr als zwei Wochen scheitert (8.3.9, Zertifikate untersuchen).
- Apache oder nginx: der Timer Ihres ACME-Clients oder die Erneuerung des Panels, plus eine externe Prüfung des Ablaufs.
- Halten Sie NTP am Laufen: Zertifikate werden gegen die Systemuhr geprüft.

Wenn es schon passiert ist: Ablauf von Zertifikaten.

Prüfungen

Shell

```
# headers on a page
curl -sI https://example.com/ | grep -i -E 'strict-transport|x-content-type|x-frame|referrer|permissions|content-security'
# the session cookie: expect Secure; HttpOnly; SameSite=Lax
curl -sI https://example.com/user/login | grep -i set-cookie
# directly, no proxy: a visitor's own header must change nothing (the redirect stays http)
curl -s -o /dev/null -w '%{redirect_url}\n' -H 'X-Forwarded-Proto: https' http://example.com/user/logout
```

Was schiefgeht

Symptom	Ursache
Absolute URLs und Umleitungen sagen `http://` hinter einem HTTPS-Proxy; eine Umleitungsschleife auf der Anmeldeseite	der Proxy steht nicht in `TrustedProxies[]` (oder nicht in der Liste von Velocity), oder er sendet kein `X-Forwarded-Proto`
Angemeldete Benutzer werden mit `CookieSecure=true` auf jeder Seite abgemeldet	dasselbe: Der Kernel hält die Anfrage für einfaches HTTP
Jeder Besucher hat im Audit und bei der Sperre die Adresse des Proxys	`ClientIpByCustomHTTPHeader` nicht gesetzt, oder dem Proxy wird nicht vertraut
Ein Besucher kann weiterhin Schema oder Adresse wählen	ein vertrauenswürdiger Bereich, der viel weiter ist als Ihre Proxys, oder der Anwendungsserver ist direkt erreichbar
Der Browser warnt auf einer Velocity-Website vor dem Zertifikat	Velocity liefert seinen selbstsignierten Ersatz aus: Das echte Paar ist abgelaufen, passt nicht oder ist noch nicht ausgestellt; `exp:velocity ssl show`

Checkliste

☐ Jeder Name der Website liefert HTTPS aus; einfaches HTTP leitet um.

☐ `SiteURL` und die URLs der Siteaccesses nutzen `https://`.

☐ `[Session] CookieSecure=true`.

☐ HSTS an, zuerst kurz; auf ein Jahr erhöht, sobald es stabil ist; `includeSubDomains` mit dem Inhaber der Domain vereinbart.

☐ Hinter einem Proxy: `TrustedProxies[]` (oder `Q.webserver.proxy.trusted`) nennt genau Ihre Proxys; die Prüfung mit `X-Forwarded-Proto` oben ändert nichts.

☐ Ablauf der Zertifikate überwacht; NTP läuft.

Verwandte Seiten

- Sicherheit für Hoster: Überblick
- HTTPS-Zertifikate in Velocity
- TLS-Zertifikate in Panels
- Varnish und Reverse Proxys
- Ablauf von Zertifikaten
- Kapitel 13: Sicherheitshärtung für die Produktion
- Freie Dokumentation: Diese Seite darf unter der GNU FDL 1.3 frei kopiert, übersetzt und verbessert werden.

> **Werden Sie Exponential-Partner**
>
> Partner teilen ihre Informationen, stellen Mittel bereit, geben Beiträge zurück und entwickeln die Produkte mit uns, und so wächst Exponential. Eine Partner-Website und ein Dashboard sind für Mitte 2027 geplant: Werden Sie noch heute Partner, damit wir schneller wachsen und neue Funktionen früher liefern.
>
> Partner: hier beginnen | Wie Sie sich bewerben

Exponential Copyright © 1998-2026 Exponential Foundation and 7x
Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".
This documentation is free documentation, licensed under the GNU Free Documentation License 1.3 or later:
https://www.gnu.org/licenses/fdl-1.3.html.