

# Let's Encrypt: Kostenlose SSL-Zertifikate auf STRATO Linux-Server nutzen

# Let's Encrypt: Kostenlose SSL-Zertifikate auf STRATO Linux-Server nutzen

Die Zertifizierungsstelle **Let's Encrypt** stellt seit Ende 2015 kostenlos und ohne Papierkram SSL/TLS-Zertifikate (X.509) aus. Außerdem ist die Einrichtung auf dem Server automatisch per Client möglich, was die Konfiguration sehr einfach macht.

Hinter Let's Encrypt steht nicht irgendein Startup, sondern die **Internet Security Research Group** mit Akamai, Cisco und Mozilla in ihren Reihen. Wir zeigen, wie Du Dir kostenlos ein SSL-Zertifikat holst und auf einem V-Server Linux von STRATO einrichten und nutzen kannst.

Let's Encrypt ist ein recht neues Angebot und hat gerade die öffentliche Beta verlassen. Beim Ausprobieren hatten wir keine großen Probleme. Es kann dennoch sein, dass sich diese Anleitung mit der Zeit noch in einigen Details ändern wird. Bei Problemen oder Unstimmigkeiten hilft Dir auch immer ein Blick in die stets aktuelle **Client-Dokumentation**.

## Die Installation umfasst folgende Schritte:

1. Voraussetzungen für Let's Encrypt .....	3
2. Auf dem Server anmelden .....	4
3. Let's Encrypt installieren .....	5
3.1 Update-Manager aktualisieren .....	5
3.2 Paketquellen prüfen .....	5
3.3 Git installieren .....	6
3.4 Let's Encrypt laden .....	6
3.5 Let's Encrypt installieren .....	7
3.6 Details festlegen .....	8
4. Einrichtung prüfen .....	10
5. Zertifikate aktuell halten .....	12
6. Let's Encrypt Client aktuell halten .....	13
7. Fazit .....	14

# 1. Voraussetzungen für Let's Encrypt

Das Besondere an Let's Encrypt sind nicht nur die kostenlosen Zertifikate. Denn wer schon einmal mit Zertifikaten jongliert hat, weiß: Auch die Einrichtung ist harte Admin-Arbeit. Bei Let's Encrypt klappen viele Schritte automatisch über das Protokoll ACME (Automated Certificate Management Environment), über das der Webserver mit dem Zertifikats-Server kommuniziert. Aktuell können das leider noch nicht alle Webserver auf allen Plattformen. Den offiziellen Client für die automatische Konfiguration gibt es nur für Linux-Server mit Apache 2 und aktuell auch für nginx – allerdings mit eingeschränktem Funktionsumfang.

Nutzer von anderen Web- oder Windows-Servern haben es nicht ganz so einfach. Eine Alternative sind Dritt-Tools wie das OpenSource-Werkzeug **Oocx.ACME** für die Zusammenarbeit mit Windows-Servern. Die Entwicklung alternativer Clients ist sehr dynamisch, einen aktuellen Überblick gibt die **Let's-Encrypt-Webseite**.

In dieser Anleitung zeigen wir Dir, wie Du Let's Encrypt auf einem STRATO V-Server Linux mit dem offiziellen Client einrichten kannst. Zum Testzeitpunkt läuft darauf Ubuntu 14.04 LTS 64 Bit.

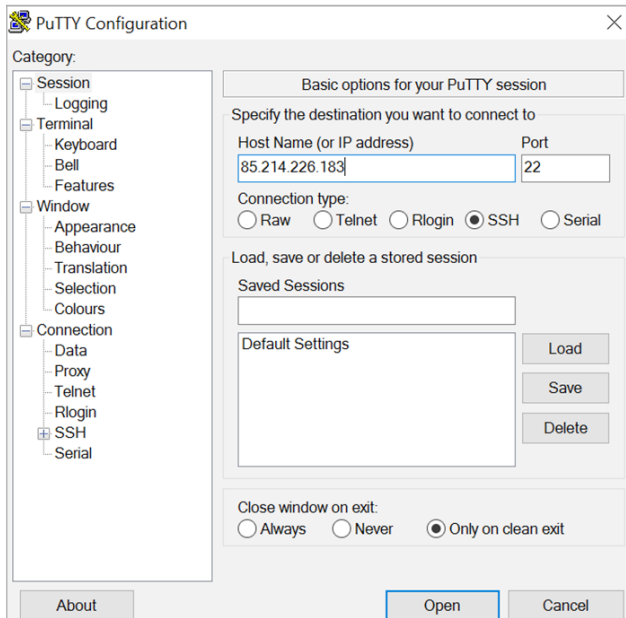
## Das brauchst Du für diese Anleitung:

- ✓ Domain
- ✓ Administrativen Zugriff auf Domain
- ✓ Laufender Linux-Server mit Apache 2
- ✓ Korrekt konfigurierte Webseite

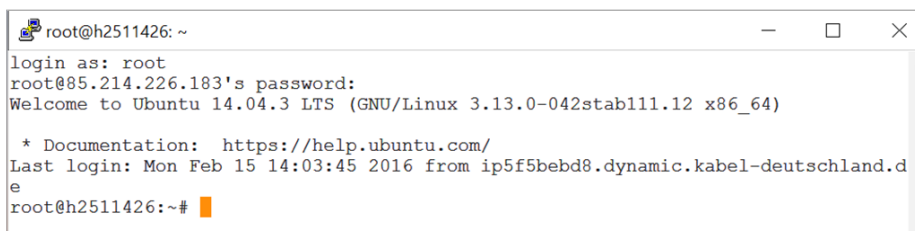
**Wichtig:** Backups sind immer ratsam. Bei STRATO sorgt die automatische Backup-Funktion BackupControl dafür, dass tägliche Backups angelegt werden. Kunden müssen sich aber selbst um konsistente Datenbank-Backups kümmern. Sichere Deine Server-Daten bevor Du mit der HTTPS-Konfiguration beginnst.

## 2. Auf dem Server anmelden

Für Let's Encrypt ist ein Root-Zugriff nötig, den Du unter Windows sehr einfach per SSH-Zugang auf den Linux-Server via **Putty** herstellen kannst.

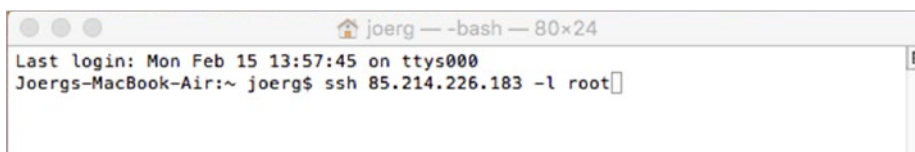


Putty öffnet eine Kommandozeile, auf der Du Dich als „root“ mit Deinem Admin-Passwort am Server anmelden kannst.



Wenn Du Linux oder OS X nutzt, brauchst Du kein zusätzliches Tool wie Putty. Du kannst Dich über die Kommandozeile per SSH mit dem Server verbinden. Der Befehl lautet dann:

**ssh IP-Adresse/Hostname -l root**



Anschließend fragt der Server nach dem Admin-Passwort. Bei korrekter Eingabe bist Du sofort mit dem Linux-Server verbunden. Aus Sicherheitsgründen wird weder bei Putty noch im Terminal-Fenster von Linux oder OS X die Passwort-Eingabe angezeigt.



```
joerg — root@h2511426: ~ — ssh 85.214.226.183 -l root — 80x24
Last login: Mon Feb 15 13:57:45 on ttys000
[Joergs-MacBook-Air:~ joerg$ ssh 85.214.226.183 -l root
]
[root@85.214.226.183's password:
]
Welcome to Ubuntu 14.04.3 LTS (GNU/Linux 3.13.0-042stab111.12 x86_64)

 * Documentation:  https://help.ubuntu.com/
Last login: Mon Feb 15 14:14:02 2016 from ip5f5bebd8.dynamic.kabel-deutschland.d
e
root@h2511426:~#
```

## 3. Let's Encrypt installieren

### 3.1 Update-Manager aktualisieren

Bevor Du den Client von Let's Encrypt installierst, solltest Du den Update-Manager-Cache aktualisieren. Dabei wird nichts installiert, nur die lokal vorhandenen Paketbeschreibungen werden auf den aktuellen Stand gebracht. Hinweis: Du kannst Let's Encrypt auch über Plesk Extensions installieren und konfigurieren.

**sudo apt-get update**

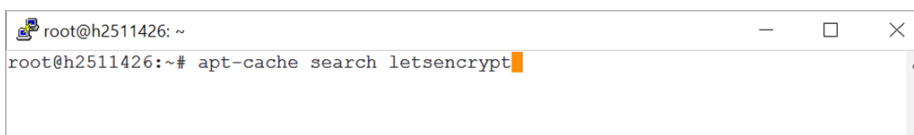


```
root@h2511426: ~
root@h2511426:~# sudo apt-get update
```

### 3.2 Paketquellen prüfen

Jetzt ist es einen Versuch wert zu prüfen, ob Let's Encrypt schon in den Paketquellen der verwendeten Linux-Distribution aufgenommen worden ist. Falls ja, kannst Du Let's Encrypt noch einfacher per *apt-get install* installieren. In unserem Fall mit Ubuntu 14.04 haben wir aber Pech.

**apt-cache search letsencrypt**

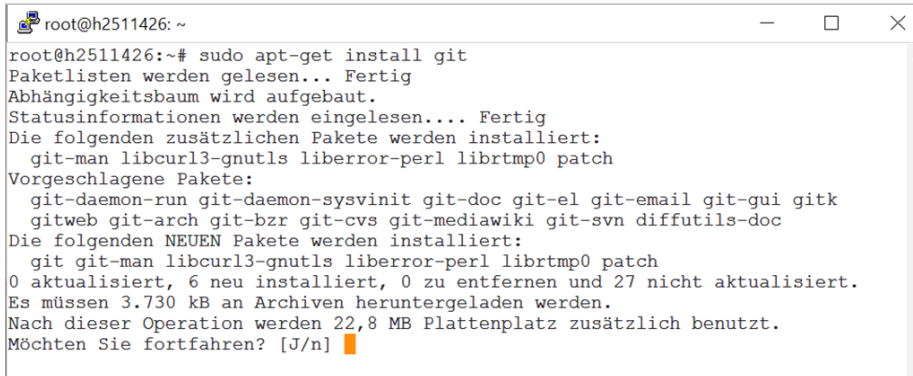


```
root@h2511426: ~
root@h2511426:~# apt-cache search letsencrypt
```

### 3.3 Git installieren

Let's Encrypt wird über Github verteilt, also musst Du einen Git-Client installieren, um an die Software zu kommen.

```
sudo apt-get install git
```

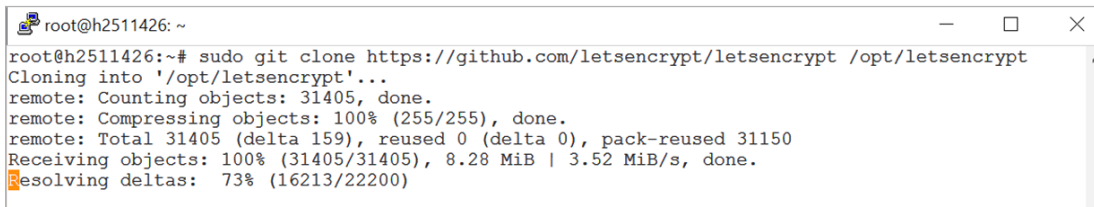


```
root@h2511426:~# sudo apt-get install git
Paketlisten werden gelesen... Fertig
Abhängigkeitsbaum wird aufgebaut.
Statusinformationen werden eingelesen... Fertig
Die folgenden zusätzlichen Pakete werden installiert:
  git-man libcurl3-gnutls liberror-perl librtmp0 patch
Vorgeschlagene Pakete:
  git-daemon-run git-daemon-sysvinit git-doc git-el git-email git-gui gitk
  gitweb git-arch git-bzr git-cvs git-mediawiki git-svn diffutils-doc
Die folgenden NEUEN Pakete werden installiert:
  git git-man libcurl3-gnutls liberror-perl librtmp0 patch
0 aktualisiert, 6 neu installiert, 0 zu entfernen und 27 nicht aktualisiert.
Es müssen 3.730 kB an Archiven heruntergeladen werden.
Nach dieser Operation werden 22,8 MB Plattenplatz zusätzlich benutzt.
Möchten Sie fortfahren? [J/n]
```

### 3.4 Let's Encrypt laden

Erst jetzt geht es an den Download des Clients von Let's Encrypt. Du platzierst mit dem nächsten Befehl eine Kopie des Github-Verzeichnisses in `/opt/letsencrypt`.

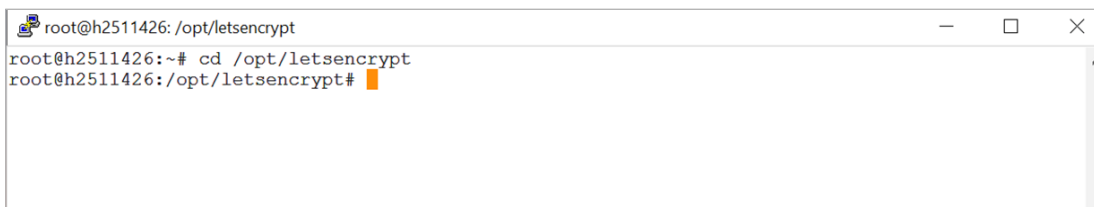
```
sudo git clone https://github.com/letsencrypt/letsencrypt /opt/letsencrypt
```



```
root@h2511426:~# sudo git clone https://github.com/letsencrypt/letsencrypt /opt/letsencrypt
Cloning into '/opt/letsencrypt'...
remote: Counting objects: 31405, done.
remote: Compressing objects: 100% (255/255), done.
remote: Total 31405 (delta 159), reused 0 (delta 0), pack-reused 31150
Receiving objects: 100% (31405/31405), 8.28 MiB | 3.52 MiB/s, done.
Resolving deltas: 73% (16213/22200)
```

Anschließend wechselst Du in das eben angelegte Verzeichnis.

```
cd /opt/letsencrypt
```



```
root@h2511426: /opt/letsencrypt
root@h2511426:~# cd /opt/letsencrypt
root@h2511426: /opt/letsencrypt#
```

### 3.5 Let's Encrypt installieren

Bis hierher könnte man den Eindruck haben, dass Let's Encrypt komplizierter ist als angepriesen, doch jetzt kommt der magische Schritt. Mit nur einem Kommando richtest Du alles ein, beginnend mit dem Client selbst. Dieser liest dann ohne Dein Zutun die Konfiguration des Webservers aus, erzeugt Schlüssel, beantragt das Zertifikat, weist die Echtheit der Domain nach, holt und installiert das Zertifikat. Wichtig zu wissen ist, dass wir in unserem Beispiel die Domain letsencrypttest.de verwenden und gleichzeitig ein Apache-Plugin von Let's Encrypt nutzen, das momentan so nur auf Debian-basierten Betriebssystemen funktioniert. Mit einem STRATO Server Linux unter Ubuntu kannst Du das Buchstabe für Buchstabe nachmachen. Für die eigene Konfiguration muss im folgenden Befehl aber unbedingt der Domain-Name angepasst werden.

Der Client kennt die Variablen für ServerName und ServerAlias aus der Apache-Konfiguration. Gebe alle Subdomains und Aliase an, etwa auch „blog.ihre-domain.de“. Der Client holt dann ein Zertifikat für alle Domains, die erste angegebene Domain wird dann als Hauptdomain für die Erzeugung des Zertifikats verwendet. Deshalb sollte am Anfang die Top-Level-Domain stehen. Wichtig: Wenn für verschiedene virtuelle Hosts auch verschiedene Zertifikate genutzt werden sollen, muss der folgende Befehl mehrfach unter Angabe der jeweiligen Domain gestartet werden.

```
./letsencrypt-auto --apache -d letsencrypttest.de -d www.letsencrypttest.de
```

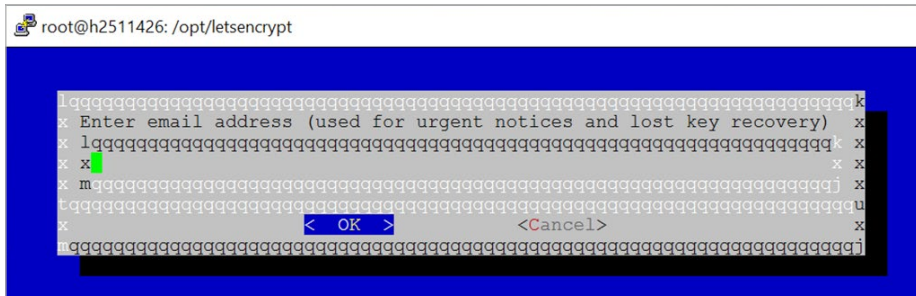
```
root@h2511426: /opt/letsencrypt
Holen: 11 ftp://ftp.stratoserver.net trusty/restricted Translation-de_DE.utf8
Holen: 12 ftp://ftp.stratoserver.net trusty/universe Translation-de_DE
Holen: 13 ftp://ftp.stratoserver.net trusty/universe Translation-de_DE.utf8
Holen: 14 ftp://ftp.stratoserver.net trusty/main Translation-de_DE
Holen: 15 ftp://ftp.stratoserver.net trusty/main Translation-de_DE.utf8
Holen: 16 ftp://ftp.stratoserver.net trusty/restricted Translation-de_DE
Holen: 17 ftp://ftp.stratoserver.net trusty/restricted Translation-de_DE.utf8
Holen: 18 ftp://ftp.stratoserver.net trusty/universe Translation-de_DE
Holen: 19 ftp://ftp.stratoserver.net trusty/universe Translation-de_DE.utf8
Holen: 20 ftp://ftp.stratoserver.net trusty/main Translation-de_DE
Holen: 21 ftp://ftp.stratoserver.net trusty/main Translation-de_DE.utf8
Holen: 22 ftp://ftp.stratoserver.net trusty/restricted Translation-de_DE
Holen: 23 ftp://ftp.stratoserver.net trusty/restricted Translation-de_DE.utf8
Holen: 24 ftp://ftp.stratoserver.net trusty/universe Translation-de_DE
Holen: 25 ftp://ftp.stratoserver.net trusty/universe Translation-de_DE.utf8
Holen: 26 ftp://ftp.stratoserver.net trusty/main Translation-de_DE
Ign ftp://ftp.stratoserver.net trusty/main Translation-de_DE
Holen: 27 ftp://ftp.stratoserver.net trusty/main Translation-de_DE.utf8
Ign ftp://ftp.stratoserver.net trusty/main Translation-de_DE.utf8
Holen: 28 ftp://ftp.stratoserver.net trusty/restricted Translation-de_DE
Ign ftp://ftp.stratoserver.net trusty/restricted Translation-de_DE
Holen: 29 ftp://ftp.stratoserver.net trusty/restricted Translation-de_DE.utf8
Ign ftp://ftp.stratoserver.net trusty/restricted Translation-de_DE.utf8
Holen: 30 ftp://ftp.stratoserver.net trusty/universe Translation-de_DE
Ign ftp://ftp.stratoserver.net trusty/universe Translation-de_DE
Holen: 31 ftp://ftp.stratoserver.net trusty/universe Translation-de_DE.utf8
Ign ftp://ftp.stratoserver.net trusty/universe Translation-de_DE.utf8
Paketlisten werden gelesen... 57%
```

Das Zertifikat unterscheidet einzelne Subdomains. Du kannst zum Beispiel auch nur für eine Subdomain wie in Blog ein Zertifikat holen. Der Befehl würde dann lauten:

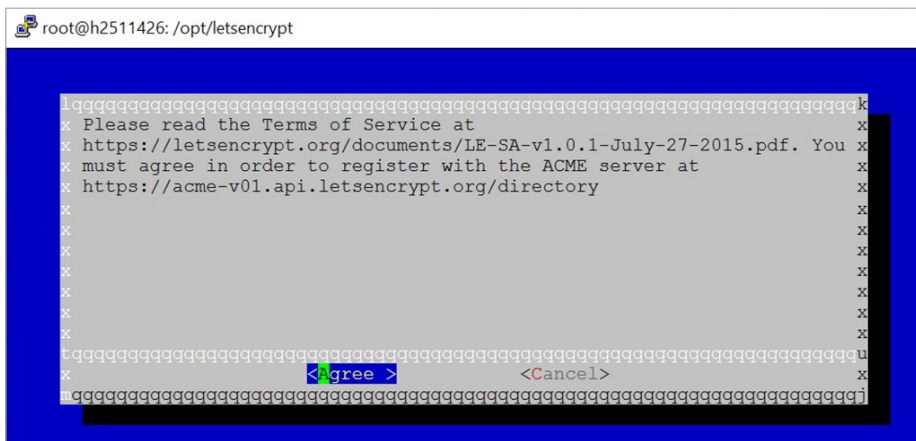
```
./letsencrypt-auto --apache -d blog.letsencrypttest.de
```

### 3.6 Details festlegen

Der letzte Befehl kann ein paar Minuten zur Abarbeitung in Anspruch nehmen, denn es wird auch noch eine virtuelle Python-Umgebung eingerichtet. Läuft alles erfolgreich durch, landest Du in einem Installer, der Dich Schritt für Schritt durch die Anpassung der Einstellungen führt. Erst musst Du eine Mailadresse angeben, die für Notfälle wie verlorene Schlüssel kontaktiert wird. Außerdem wird man auf diesem Weg benachrichtigt, kurz bevor das Zertifikat abläuft und erneuert werden muss. Am besten eignet sich hier das Mailkonto, das der Webmaster für die Administration der Seite verwendet.



Als nächstes kommt das obligatorische Abnicken der Geschäftsbedingungen.





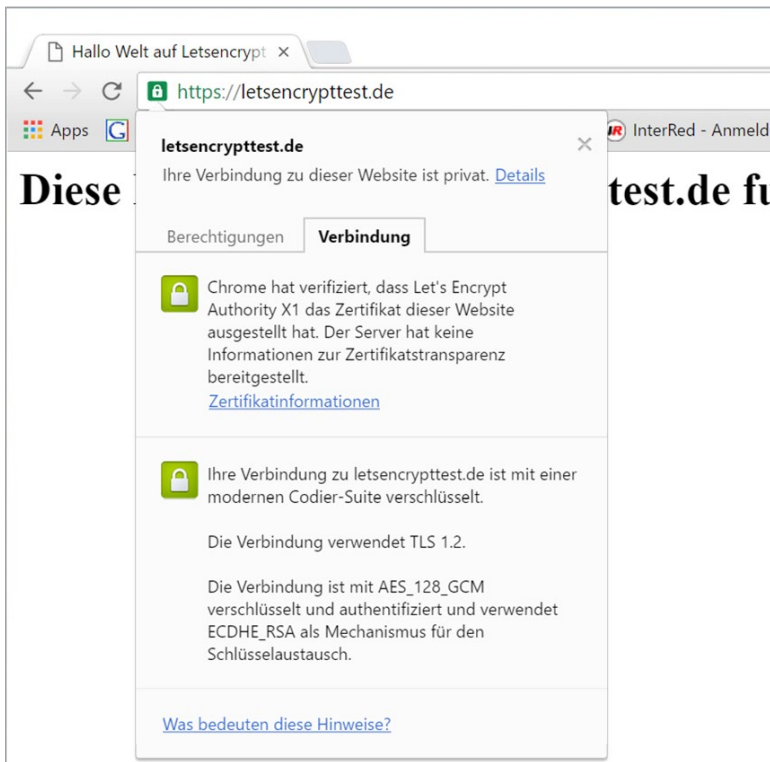


## 4. Einrichtung prüfen

Im Verzeichnis `/etc/letsencrypt/live` findest Du für jede Domain einen Unterordner, in dem die eben erzeugten SSL-Zertifikate liegen. Das sind die vier Dateien `cert.pem`, `chain.pem`, `fullchain.pem` und `privkey.pem`.

```
root@h2511426: /opt/letsencrypt
root@h2511426:/opt/letsencrypt# ls /etc/letsencrypt/live
letsencrypttest.de
root@h2511426:/opt/letsencrypt# ls /etc/letsencrypt/live/letsencrypttest.de/
cert.pem chain.pem fullchain.pem privkey.pem
root@h2511426:/opt/letsencrypt#
```

Dann bietet sich einfach ein Aufruf der Webseite im Browser an. Auch wenn Du nur die Domain in der URL-Zeile angibst, solltest Du auf der verschlüsselten Seite landen. Browser wie Firefox oder Chrome zeigen dann auch gleich ein Schloss als Kennzeichen einer sicheren Verbindung an. Teste alle Domains und Subdomains sowie Aliase durch und prüfe außerdem, ob auch die Weiterleitung funktioniert, wenn Du explizit „http“ angibst.



Die Macher von Let's Encrypt legen den Nutzern den **SSL-Test von Qualys** ans Herz. Dort musst Du lediglich die Domain eintippen und es laufen zahlreiche SSL-Tests durch.



STRATO bietet für seine Kunden als Service-Leistung SecurityScan an, ein regelmäßiger Sicherheits-Check, der über das Kunden-Server-Login bestellt werden kann.

## 5. Zertifikate aktuell halten

Zertifikate von Let's Encrypt sind aktuell nur 90 Tage gültig – das kann sich aber jederzeit ändern. Hier gilt es vorsichtig zu sein, denn wer sich die Mühe macht und seine Webseite auf HTTPS umstellt, kann sich mit einem abgelaufenen Zertifikat selbst ins Aus schießen. Die Browser blocken in diesem Fall den Zugriff auf die Webseite. Damit es nicht zu Problemen mit sicheren Webseiten kommt, solltest Du Zertifikate immer rechtzeitig vor Ablauf verlängern. Auch das kann der Client von Let's Encrypt leisten. Wichtig dabei: Das folgende Kommando klappt nur, wenn die Zertifikate in den nächsten 30 Tagen ablaufen.

**`./letsencrypt-auto renew`**



```
root@h2511426: /opt/letsencrypt
root@h2511426: /opt/letsencrypt# ./letsencrypt-auto renew
```

Mit einem zusätzlichen Flag lässt sich die Erneuerung der Zertifikate aber auch erzwingen. Dabei ignoriert der Client die Ablaufdaten.

**`./letsencrypt-auto renew --force-renew`**



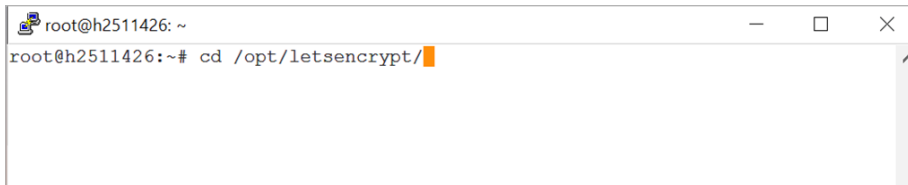
```
root@h2511426: /opt/letsencrypt
root@h2511426: /opt/letsencrypt# ./letsencrypt-auto renew --force-renew
Checking for new version...
Requesting root privileges to run letsencrypt...
/root/.local/share/letsencrypt/bin/letsencrypt --no-self-upgrade renew --force-renew
Processing /etc/letsencrypt/renewal/letsencrypttest.de.conf
new certificate deployed with reload of apache server; fullchain is /etc/letsencrypt/live/letsencrypttest.de/fullchain.pem

Congratulations, all renewals succeeded. The following certs have been renewed:
/etc/letsencrypt/live/letsencrypttest.de/fullchain.pem (success)
root@h2511426: /opt/letsencrypt#
```

## 6. Let's Encrypt Client aktuell halten

Let's Encrypt hat gerade erst die öffentliche Beta verlassen und wird dynamisch weiterentwickelt. Dementsprechend viele Updates gibt es auch für den Client. Dieser sollte auch auf dem aktuellen Stand gehalten werden. Dazu wechselt man ins Installationsverzeichnis.

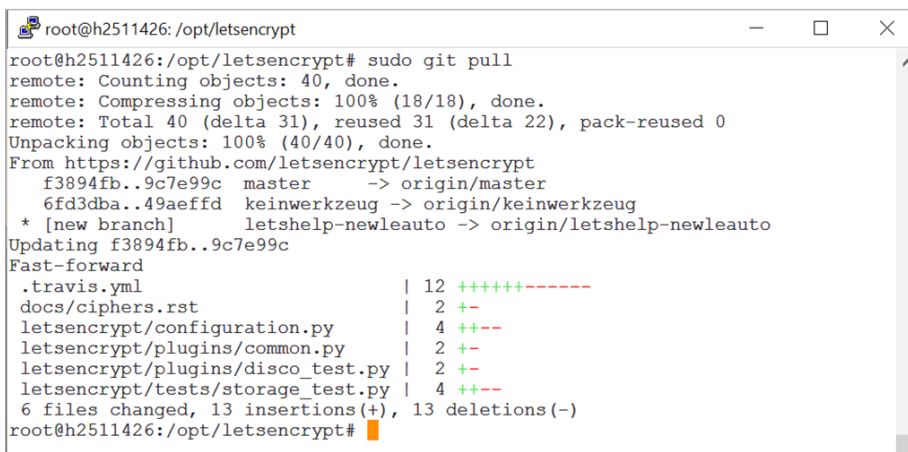
```
cd /opt/letsencrypt
```



```
root@h2511426: ~  
root@h2511426:~# cd /opt/letsencrypt/
```

Danach werden die Updates von Github gezogen.

```
sudo git pull
```



```
root@h2511426: /opt/letsencrypt  
root@h2511426:/opt/letsencrypt# sudo git pull  
remote: Counting objects: 40, done.  
remote: Compressing objects: 100% (18/18), done.  
remote: Total 40 (delta 31), reused 31 (delta 22), pack-reused 0  
Unpacking objects: 100% (40/40), done.  
From https://github.com/letsencrypt/letsencrypt  
  f3894fb..9c7e99c  master       -> origin/master  
  6fd3dba..49aeffd  keinwerkzeug -> origin/keinwerkzeug  
* [new branch]      letshelp-newleauto -> origin/letshelp-newleauto  
Updating f3894fb..9c7e99c  
Fast-forward  
 .travis.yml                | 12 ++++++-----  
 docs/ciphers.rst           |  2 +-  
 letsencrypt/configuration.py |  4 ++--  
 letsencrypt/plugins/common.py |  2 +-  
 letsencrypt/plugins/disco_test.py |  2 +-  
 letsencrypt/tests/storage_test.py |  4 ++--  
 6 files changed, 13 insertions(+), 13 deletions(-)  
root@h2511426:/opt/letsencrypt#
```

## 7. Fazit

So einfach wie mit Let's Encrypt kommt man auf dem V-Server Linux kostenlos an kein SSL/TLS-Zertifikat. In der Praxis funktioniert das schon sehr gut. Der Teufel steckt aber auch manchmal im Detail: Da es viele verschiedene Server-Konfigurationen und Anforderungen an die Verschlüsselung gibt, lohnt sich immer ein Blick in den offiziellen [User-Guide von Let's Encrypt](#).

Let's Encrypt ist aber nicht die einzige Möglichkeit, an SSL-Zertifikate zu kommen und auch nicht immer der beste Weg: Wenn Du den Umstieg auf eine SSL-verschlüsselte Webseite planst, lohnt sich auf alle Fälle auch ein Blick auf die verschiedenen Zertifikate, die STRATO kostenpflichtig anbietet. Hierbei erhältst Du einige Vorteile, wie etwa eine längere Zertifikatsgültigkeit von bis zu zwei Jahren.

Für den geschäftlichen Einsatz empfehlen wir Dir außerdem dringend, identitätsvalidierte Zertifikate oder sogar eine erweiterte Validierung zu nutzen. Hierbei wird nicht nur technisch geprüft, ob die Domain zu Deinem Server gehört, wie Let's Encrypt dies tut, sondern auch noch, ob diese Domain tatsächlich zu Dir und Deinem Geschäft gehört.