Reverse Proxy Installation auf Debian / Ubuntu

Als Grundlage für den hier zu installierenden Reverse Proxy wurde ein **Ubuntu 16.04** aufgesetzt. Im folgenden werden alle Schritte nach der fertigen OS Installation zum einrichten des Proxys beschrieben.

Was ist ein Reverse Proxy? *Grundsätzlich handelt es sich bei einem Proxy um eine Kommunikationsschnittstelle im Netzwerk, die Anfragen entgegennimmt und stellvertretend an einen Zielrechner weiterleitet. Ein Reverse Proxy wird nun aber meistens als zusätzliche Sicherheitskomponente vor einen oder mehrere Webserver geschaltet, um Anfragen aus dem Internet stellvertretend entgegen-zunehmen und an einen Backend-Server im Hintergrund weiterzuleiten.*

Eine ausführlichere Beschreibung hier: Reverse-Proxy – Kernkomponente in Sicherheitsarchitekturen

Interessante Hardware für standalone Proxies

System Konfiguration

Schon zu Beginn wird dem Proxy-Server *eine eigene fixe IP* Adresse zugeteilt. Dies ist hierbei sehr wichtig, da der Traffic zu einem späteren Zeitpunkt von **Port 80** HTTP und **Port 443** HTTPS des Routers direkt an den Proxy per Portweiterleitung vermittelt wird.

```
# vim /etc/network/interfaces
```

```
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
auto eth0
iface eth0 inet static
address 192.168.1.6
netmask 255.255.255.0
gateway 192.168.1.1
dns-nameservers 192.168.1.1
```

Nach erfolgreicher Speicherung, wird das System neugestartet.

```
# init 6
```

Installation der Grundkomponenten

Zu Beginn werden erst einmal alle Grundpakete, welche zum einrichten unseres Proxys gebraut werden installiert. **ACHTUNG: Das Paket "`libapache2-mod-proxy-html`" ist bei Ubuntu 16.04 schon in der Base Installation enthalten!**

```
UBUNTU 14.04:
# apt-get install apache2 libapache2-mod-proxy-html libxml2-dev

UBUNTU 16.04:
# apt-get install apache2 libxml2-dev
```

Aktivierung der Proxy Komponenten aus dem Apache2 Paket.

```
# a2enmod proxy proxy_ajp proxy_http proxy_wstunnel rewrite deflate headers
proxy_balancer proxy_connect proxy_html xml2enc vhost_alias ssl
```

Nun wird Letsencript installiert, damit wir später damit auch unser eigenes SSL Zertifikat generieren können.

```
# apt-get install git-core
```

```
# cd /opt
# git clone https://github.com/letsencrypt/letsencrypt
```

Konfigurieren der Virtual-Hosts

Nun wenn wir die Basis der gebrauchten Pakete installiert haben, können wir im nächsten Schritt die Virtual-Hosts unseres Reverse Proxys definieren. Bei diesem Schritt, ist es **wichtig**, dass wir anfangs nur die *proxy_http.conf* aktivieren, da in der *proxy_http**s**.conf* bereits bei allen Virtual-Host der Zertifikatspfad angegeben ist, würde dies zu einem kritischen Fehler beim restarten des Webservers führen, da zum jetzigen Zeitpunkt noch keine SSL Zertifikate existieren, welche jedoch dort eingebunden würden..

Bereinigen der Virtual-Hosts

Zuerst werden die Standart *sites* deaktiviert und gelöscht:

```
# a2dissite 000-default.conf
# a2dissite default-ssl.conf

# rm /etc/apache2/sites-available/000-default.conf
# rm /etc/apache2/sites-available/default-ssl.conf
```

Hinzufügen der eigenen Virtual-Hosts

Nun können auch schon bereits die eigenen Virtual-Host-files, welche später vom Proxy gebraucht werden auf dem System unter **/etc/apache2/sites-available/** erstellt werden.

ACHTUNG: Folgene Virtual-Host sind reine Beispiele und müssen dementsprechend noch durch richtige Domainnamen ergänzt, abgeändert werden.

Alle anzupassenden Zeilen sind Blau markiert! PS: EXAMPLE.COM wird durch eigenen Domain-namen ersetzt!

EXAMPLE.COM.conf:

```
# vim /etc/apache2/sites-available/EXAMPLE.COM.conf
```

```
<VirtualHost *:80>
# ServerName example.com
ServerName localhost
 #
   ServerAdmin admin@example.com
   DocumentRoot /var/www/html
 #
</VirtualHost>
```

---

proxy_http.conf:

```
# vim /etc/apache2/sites-available/proxy_http.conf
```

```
#-------------------------------------------------------------------------
---------------------
#                             REDIRECTION FOR NON EXISTENT SUBDOMAINS
#-------------------------------------------------------------------------
---------------------
<VirtualHost *:80>
    ServerName example.com
    RewriteEngine On
    RewriteRule ^/?(.*) https://www.example.com/$1 [R,L]
```

```
</VirtualHost>
#------------------------------------------------------------------------------
---------------------



#------------------------------------------------------------------------------
---------------------
#                                       MAIN REDIRECTIONS
#------------------------------------------------------------------------------
---------------------
<VirtualHost *:80>
 ServerName example.com
 #
     ServerAdmin admin@example.com

     ServerAlias www.example.com

     ServerAlias piwik.example.com
     ServerAlias cloud.example.com

     ServerAlias wiki.example.com
     ServerAlias test.example.com

     RewriteEngine On
     RewriteRule ^/?(.*) https://%{SERVER_NAME}/$1 [R,L]
</VirtualHost>
```

---

proxy_https.conf:

```
# vim /etc/apache2/sites-available/proxy_https.conf
```

```
<IfModule mod_ssl.c>

#------------------------------------------------------------------------------
---------------------
#                                 VARIABLES and GENERAL SETTINGS
#------------------------------------------------------------------------------
---------------------
define blackgate_serveradmin "admin@example.com"
define blackgate_ssl_path "/etc/letsencrypt/live/blackgate.org-0001"

SSLCompression off
SSLUseStapling on
SSLStaplingCache "shmcb:logs/stapling-cache(150000)"
SSLOpenSSLConfCmd DHParameters "/etc/ssl/private/dhparam.pem"
```

```
# Requires Apache >= 2.4.11
SSLSessionTickets Off

#SSLCipherSuite "EECDH+AESGCM:EDH+AESGCM:AES256+EECDH:AES256+EDH"
SSLCipherSuite "EECDH+AESGCM:EDH+AESGCM:AES256+EECDH:AES256+EDH"
SSLProtocol All -SSLv2 -SSLv3 -TLSv1 -TLSv1.1
Header always set Strict-Transport-Security "max-age=63072000;
includeSubdomains; preload"

#-----------------------------------------------------------------------------
--------------------
#                                       MAIN SERVICES
#-----------------------------------------------------------------------------
--------------------

<VirtualHost *:443>
 ServerName www.example.com
 #
     ServerAdmin ${blackgate_serveradmin}
     SSLEngine on
     SSLCertificateFile ${blackgate_ssl_path}/cert.pem
     SSLCertificateKeyFile ${blackgate_ssl_path}/privkey.pem
     SSLCertificateChainFile ${blackgate_ssl_path}/chain.pem

     ProxyPass /error_docs !
     ErrorDocument 503 /error_docs/ServiceUnavailable.html
     ProxyPass      / http://192.168.1.21/
     ProxyPassReverse   / http://192.168.1.21/

     <Proxy http://192.168.1.21/>
             Order deny,allow
             Allow from all
     </Proxy>
</VirtualHost>

<VirtualHost *:443>
 ServerName piwik.example.com
 #
     ServerAdmin ${blackgate_serveradmin}
     SSLEngine on
     SSLCertificateFile ${blackgate_ssl_path}/cert.pem
     SSLCertificateKeyFile ${blackgate_ssl_path}/privkey.pem
     SSLCertificateChainFile ${blackgate_ssl_path}/chain.pem
     ProxyPass / http://192.168.1.11:8080/
     ProxyPassReverse / http://192.168.1.11:8080/
     ProxyPreserveHost On
     <Proxy http://192.168.1.11:8080/>
             Order deny,allow
             Allow from all
     </Proxy>
</VirtualHost>
```

```
<VirtualHost *:443>
 ServerName cloud.example.com
 #
     ServerAdmin ${blackgate_serveradmin}
     SSLEngine on
     SSLCertificateFile ${blackgate_ssl_path}/cert.pem
     SSLCertificateKeyFile ${blackgate_ssl_path}/privkey.pem
     SSLCertificateChainFile ${blackgate_ssl_path}/chain.pem
     ProxyPreserveHost On

     ProxyPass /error_docs !
     ErrorDocument 503 /error_docs/ServiceUnavailable.html

     ProxyPass / http://192.168.1.24/ retry=1 acquire=3000 Timeout=5400
Keepalive=On flushpackets=On
     ProxyPassReverse / http://192.168.1.24/
     <Proxy http://192.168.1.24/>
             Order deny,allow
             Allow from all
     </Proxy>
</VirtualHost>
<VirtualHost *:443>
 ServerName wiki.example.com
 #
     ServerAdmin ${blackgate_serveradmin}
     SSLEngine on
     SSLCertificateFile ${blackgate_ssl_path}/cert.pem
     SSLCertificateKeyFile ${blackgate_ssl_path}/privkey.pem
     SSLCertificateChainFile ${blackgate_ssl_path}/chain.pem

     ProxyPass /error_docs !
     ErrorDocument 503 /error_docs/ServiceUnavailable.html

     ProxyPass / http://192.168.1.10/
     ProxyPassReverse / http://192.168.1.10/
     ProxyPreserveHost On

     <Proxy http://192.168.1.10/>
          Require all granted
     </Proxy>
</VirtualHost>

<VirtualHost *:443>
 ServerName test.example.com
 #
     ServerAdmin ${blackgate_serveradmin}
     SSLEngine on
     SSLCertificateFile ${blackgate_ssl_path}/cert.pem
     SSLCertificateKeyFile ${blackgate_ssl_path}/privkey.pem
     SSLCertificateChainFile ${blackgate_ssl_path}/chain.pem
```

```
     ProxyPass / http://192.168.1.15/ retry=1 acquire=3000 Timeout=7200
Keepalive=On flushpackets=On
     ProxyPassReverse / http://192.168.1.15/
     <Proxy http://192.168.1.15/>
             Order deny,allow
             Allow from all
     </Proxy>

     #ProxyPassReverseCookiePath /guacamole /
</VirtualHost>


#<VirtualHost *:443>
#     ServerAlias *.example.com
#     SSLEngine on
#     SSLCipherSuite EECDH+AESGCM:EDH+AESGCM:AES256+EECDH:AES256+EDH
#     SSLProtocol All -SSLv2 -SSLv3
#     Header always set Strict-Transport-Security "max-age=63072000;
includeSubdomains; preload"
#     SSLCertificateFile /etc/letsencrypt/live/blackgate.org/cert.pem
#     SSLCertificateKeyFile /etc/letsencrypt/live/blackgate.org/privkey.pem
#     SSLCertificateChainFile /etc/letsencrypt/live/blackgate.org/chain.pem
#     RewriteEngine On
#     Redirect 301 / https://www.example.com
#</VirtualHost>


</IfModule>
# vim: syntax=apache ts=4 sw=4 sts=4 sr noet
```

---

letsencript_dummy.conf:

```
# vim /etc/apache2/sites-available/letsencript_dummy.conf
```

```
<VirtualHost *:80>
 ServerName example.com
 #
     ServerAdmin admin@example.com

     ServerAlias www.example.com
     ServerAlias piwik.example.com
     ServerAlias cloud.example.com

     ServerAlias wiki.example.com
     ServerAlias test.example.com
```

```
   DocumentRoot /var/www/html
 #
</VirtualHost>
```

Aktivieren der dummy.conf

Für den nächsten Schritt, müssen wir die **letsencript_dummy.conf** aktivieren. **Alle anderen sites bleiben deaktiviert.**

```
# a2ensite letsencript_dummy.conf
# service apache2 reload
```

LetsEncript Konfigurieren

Im ersten Schritt, wird nun zuerst ein **neues Zertifikat** für die Domäne “blackgate.org” und deren Sub-Domains des Reverse Proxys generiert. Die **Key-size** setzen wir hier für eine bessere Sicherheit auf **4096** anstatt den herkömmlichen 2048 Bit!

```
# cd /opt/letsencrypt/
```

```
# ./letsencrypt-auto certonly --rsa-key-size 4096 -d example.com -d
www.example.com -d piwik.example.com -d cloud.example.com -d
wiki.example.com -d test.example.com
```

Nach erfolgreichem Durchlauf und der Meldung, dass das Zertifikat erfolgreich unter: **/etc/letsencrypt/live/blackgate.org/cert.pem** erstellt wurde, kann mit dem nächsten Schritt weitergefahren werden.

Automatisiertes Key Update

Da das Letsencript Zertifikat nur eine Gültigkeit von drei Wochen hat, wird hier eine automatische Aktualisierung des Zertifikates empfohlen. Dies wird bei mir über einen crontab Eintrag erreicht.

```
# vim /etc/crontab
```

```
# /etc/crontab: system-wide crontab
# Unlike any other crontab you don't have to run the `crontab'
# command to install the new version when you edit this file
# and files in /etc/cron.d. These files also have username fields,
```

```
# that none of the other crontabs do.

SHELL=/bin/sh
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin

# m h dom mon dow user  command
17 *    * * *    root    cd / && run-parts --report /etc/cron.hourly
25 6    * * *    root    test -x /usr/sbin/anacron || ( cd / && run-parts --
report /etc/cron.daily )
47 6    * * 7    root    test -x /usr/sbin/anacron || ( cd / && run-parts --
report /etc/cron.weekly )
52 6    1 * *    root    test -x /usr/sbin/anacron || ( cd / && run-parts --
report /etc/cron.monthly )

0  12   * * 6   root    /opt/letsencrypt/letsencrypt-auto renew >>
/var/log/le-renew.log
#
```

Scharf schalten der Proxy Konfiguration

Wenn bis hierhin alles funktioniert hat; kann nun die proxy_dummi.conf deaktiviert werden und der eigentliche Proxy scharf geschalten werden.

```
# a2dissite letsencript_dummy.conf

# a2ensite EXAMPLE.COM.conf
# a2ensite proxy_http.conf
# a2ensite proxy_https.conf
```

```
# service apache2 reload
# rm /etc/apache2/sites-available/letsencript_dummy.conf
```

Der “proxy_https_plexdash.conf” darf erst angeschalten werden, wenn für diesen auch Zertifikate vorhanden sind. (Andernfalls Zertifikat Pfad in dieser conf Datei anpassen.)

Zusätzliche Konfigurationen

Alle hier gemachten Konfigurationsänderungen, haben keinen direkten Einfluss auf die Proxy Funktion. Sie dienen legindlich der Sicherheit und der personalisierung.

Härten des Apache-Proxys

Zum härten des Apache2 Webservers werden wir nun die security.conf Konfigurationsdatei

folgendermassen anpassen:

```
# vim /etc/apache2/conf-enabled/security.conf
```

```
# ServerTokens
ServerTokens Prod

ServerSignature Off

# Allow TRACE method
TraceEnable Off

# Setting this header will prevent other sites from embedding pages from
this
# site as frames. This defends against clickjacking attacks.
# Requires mod_headers to be enabled.
#
Header set X-Frame-Options: "sameorigin"
```

```
# service apache2 reload
```

Eigene ErrorPages definieren

Um eigene ErrorPages unter einem Apache Reverse Proxy einzubinden muss **folgendes snipped** in der Hauptkonfigurationsdatei von Apache2 *nach dem letzten </Directory> Eintrag* eingetragen werden:

snippet

```
Alias /error_docs /var/www/error_pages
ProxyPass /error_docs !
ErrorDocument 400 /error_docs/BadRequest.html
ErrorDocument 401 /error_docs/Unauthorized.html
ErrorDocument 403 /error_docs/Forbidden.html
ErrorDocument 404 /error_docs/NotFound.html
ErrorDocument 500 /error_docs/ServerError.html
ErrorDocument 503 /error_docs/server_offline.html
```

```
# vim /etc/apache2/apache2.conf
```

Nach dem speichern, werden anschliessend die besagten ErrorDocs *(Gleiche Namensgebung wie oben; z.B: BadRequest.html)* **nach /var/www/error_pages kopiert**.

ErrorPages:

- error_pages.zip

```
# chown -R www-data:www-data /var/www/error_pages/
```

```
# service apache2 reload
```

Zusätzliche Sub-Domains hinzufügen

Sollen weitere sub-Domains zu den bestehenden hinzugefügt werden, so wird folgendermassen vorgegangen:

1. Anpassen der proxy-sites und neuer Sub-Domain Namen erfassen.

```
# vim /etc/apache2/sites-available/proxy_http.conf
# vim /etc/apache2/sites-available/proxy_https.conf
```

2. Zum letsencript Binary wechseln und den letzten certonly Befehl *(Suchen mit CTRL + R)* mit der am Schluss neu angehängter Domain z.B. "**-d NEU-SUBDOM.DOMAIN.COM**" ausführen.

```
# cd /opt/letsencrypt/
# ./letsencrypt-auto certonly --rsa-key-size 4096 -d example.com -d
www.example.com -d piwik.example.com -d cloud.example.com -d
wiki.example.com -d test.example.com -d new1.example.com -d
new2.example.com
```

3. Zum Schluss muss noch der Apache Service neu geladen werden, damit das neue Zertifikat angezogen wird.

```
 # service apache2 reload
```

Setzen der korrekten Timezone

1. Die aktuelle Konfiguration kann mit **`timedatectl`** eingesehen werden.

```
# timedatectl
```

```
      Local time: Sun 2017-04-23 07:56:23 UTC
  Universal time: Sun 2017-04-23 07:56:23 UTC
        RTC time: Sun 2017-04-23 07:56:25
       Time zone: Etc/UTC (UTC, +0000)
 Network time on: yes
NTP synchronized: yes
 RTC in local TZ: no
```

2. Auflisten aller verfügbaren Timezones..

```
# timedatectl list-timezones
```

```
Africa/Abidjan
Africa/Accra
Africa/Addis_Ababa
Africa/Algiers
Africa/Asmara
Africa/Bamako
Africa/Bangui
Africa/Banjul
...
```

3. **Setzen der neuen, korrekten Timezone:** in meinem Fall: ***Zürich Schweiz***

```
# timedatectl set-timezone Europe/Zurich
```

```
 Local time: Sun 2017-04-23 09:57:37 CEST
  Universal time: Sun 2017-04-23 07:57:37 UTC
        RTC time: Sun 2017-04-23 07:57:39
       Time zone: Europe/Zurich (CEST, +0200)
 Network time on: yes
NTP synchronized: yes
 RTC in local TZ: no
```

Last update: **2017/10/27 17:05**