Um IPv6 Webseiten per http oder https für IPv4 Anfragen oder IPv4 Seiten für IPv6 Anfragen erreichbar zu machen, bietet sich ein Linux Gateway / Proxy an. Linux, in diesem Fall ein CentOS 7 System, bietet u.a. folgende Möglichkeiten ein solches Gateway bzw. einen solchen Proxy Server zu realisieren:

socat

Socat (für SOcket CAT) ist ein mächtiges Tool, das zwei bidirektionale Byte-Streams anlegt und Daten zwischen diesen überträgt. Datenkanäle können Dateien, Pipelines, Geräte (Terminal oder Modem, etc.) oder Sockets (Unix, IPv4, IPv6, Raw, UDP, TCP, SSL) sein. Zunächst muss das Paket *„socat“* per yum installiert werden:

```
[root@proxy ~]# yum install socat -y
```

Dann bitte folgendes Verzeichnis anlegen:

```
[root@proxy ~]# mkdir /etc/socat
```

Jetzt werden zwei Shell Skripte erzeugt. Ein Skript für die Umleitung des Ports 80 und ein Skript für die Umleitung des Ports 443. In beiden Fällen wird eine IPv4 Anfrage an einen IPv6 Webserver direkt weitergeleitet.

```
[root@proxy ~]# vi /etc/socat/80_socat
```

```
#!/bin/bash
# /etc/socat/80_socat
# Socat-Script Port 80

# TCP Port 80
```

```
/usr/bin/socat TCP4-LISTEN:80,fork TCP6:[IPv6-Adresse des Zielrechners]:80
```

```
[root@proxy ~]# vi /etc/socat/443_socat
```

```
#!/bin/bash
# /etc/socat/443_socat
# Socat-Script Port 443

# TCP Port 443
```

```
/usr/bin/socat TCP4-LISTEN:443,fork TCP6:[IPv6-Adresse des Zielrechners]:443
```

Die beiden soeben erzeugten Dateien ausführbar machen:

```
[root@proxy ~]# chmod 755 /etc/socat/80_socat
[root@proxy ~]# chmod 755 /etc/socat/443_socat
```

Jetzt wird ein für jeden der beiden Ports ein Dienst generiert.

1. Port 80

```
[root@proxy ~]# vi /etc/systemd/system/80_socat.service
```

```
[Unit]
Description=socat Service 80
After=network.target

[Service]
Type=simple
User=root
ExecStart=/etc/socat/80_socat
Restart=on-abort

[Install]
```

```
WantedBy=multi-user.target
```

2. Port 443

```
[root@proxy ~]# vi /etc/systemd/system/443_socat.service
```

```
[Unit]
Description=socat Service 443
After=network.target

[Service]
Type=simple
User=root
ExecStart=/etc/socat/443_socat
Restart=on-abort

[Install]
```

```
WantedBy=multi-user.target
```

Der *systemctl* Daemon muss neu geladen werden um die beiden *„Socat-Dienste“* starten zu können.

```
[root@proxy ~]# systemctl daemon-reload
```

```
[root@proxy ~]# systemctl start 80_socat
```

```
[root@proxy ~]# systemctl start 443_socat
```

Damit diese Dienste beim Start des Rechners geladen werden:

```
[root@proxy ~]# systemctl enable 80_socat
```

```
[root@proxy ~]# systemctl enable 443_socat
```

Bei IPv4 Anfragen für Port 80 und 443 werden diese nun auf einen IPv6 Webserver umgeleitet. Natürlich kann *socat* auch mit anderen Ports umgehen!

Wichtig: Im Apache Log des Ziel-Servers taucht nur die IPv6 Adresse des Quell-Servers auf!

Apache als Reverse Proxy

Der Apache HTTP Server der Apache Software Foundation und einer der meistbenutzten Webserver im Internet kann ebenfalls als Reverse Proxy eingesetzt werden.

Zunächst müssen folgende Pakete installiert werden:

```
[root@proxy ~]# yum install httpd mod_ssl mod_proxy_html
```

Das Modul *„mod_proxy_html”* schaltet das Rewriting für HTML links an damit diese funktionieren können.

```
[root@proxy  ~]# cp /usr/share/doc/httpd-2.4.6/proxy-html.conf /etc/httpd/conf.d/
```

Jetzt wird eine Reverse Proxy Konfiguration benötigt. Dazu wird die Datei */etc/httpd/conf.d/reverse-proxy.conf* mit folgendem Inhalt erzeugt:

```
#Port 80
ProxyRequests Off
ProxyPass / http://[IPv6-Adresse des Zielrechners]:80 connectiontimeout=5 timeout=30
ProxyPassReverse / http://[IPv6-Adresse des Zielrechners]:80

#Port 443
ProxyRequests Off
ProxyPass / https://[IPv6-Adresse des Zielrechners]:443 connectiontimeout=5
timeout=30
```

```
ProxyPassReverse / https://[IPv6-Adresse des Zielrechners]:443
```

Damit der Apache Dienst beim Start des Rechners geladen werden kann:

```
[root@proxy ~]# systemctl enable httpd
```

Den *httpd* Daemon starten:

```
[root@proxy ~]# systemctl start httpd
```

Die Datei */etc/httpd/conf.d/ssl.conf* sowie die SSL Zertifikate im Verzeichnis */etc/pki* des Ziel-Servers sollten auf den Proxy-Server exakt so in Kopie vorliegen.

Squid als Reverse Proxy

Squid kann nicht nur als Forward Proxy Server und Web-Cache, sondern auch als Reverse Proxy eingesetzt werden. Vor allem aufgrund seiner guten Skalierbarkeit und der ausgezeichneten Unterstützung für HTTP/HTTPS ist er für die Aufgabe prädestiniert.

Squid Paket installieren:

```
[root@proxy ~]# yum install squid
```

Damit der Proxy Dienst beim Start des Rechners geladen werden kann:

```
[root@proxy ~]# systemctl enable squid
```

Die entsprechend angepasste Squid Proxy Konfiguration wird nun benötigt. Die Datei */etc/squid/squid.conf* sollte dann wie folgt aussehen:

```
# Recommended minimum configuration:
#
# Example rule allowing access from your local networks.
# Adapt to list your (internal) IP networks from where browsing
# should be allowed
acl localnet src 10.0.0.0/8     # RFC1918 possible internal network
acl localnet src 172.16.0.0/12  # RFC1918 possible internal network
acl localnet src 192.168.0.0/16 # RFC1918 possible internal network
acl localnet src fc00::/7       # RFC 4193 local private network range
acl localnet src fe80::/10      # RFC 4291 link-local (directly plugged) machines
acl SSL_ports port 443          # https
acl Safe_ports port 80          # http
acl Safe_ports port 21          # ftp
acl Safe_ports port 443         # https
acl Safe_ports port 70          # gopher
acl Safe_ports port 210         # wais
acl Safe_ports port 1025-65535  # unregistered ports
acl Safe_ports port 280         # http-mgmt
acl Safe_ports port 488         # gss-http
acl Safe_ports port 591         # filemaker
acl Safe_ports port 777         # multiling http
acl CONNECT method CONNECT

# Recommended minimum Access Permission configuration:
#
# Deny requests to certain unsafe ports
http_access deny !Safe_ports
# Deny CONNECT to other than secure SSL ports
http_access deny CONNECT !SSL_ports
# Only allow cachemgr access from localhost
http_access allow localhost manager
http_access deny manager
#http_access deny to_localhost

# Squid normally listens to ports 80 and 443
http_port 80 accel defaultsite=www.domain.de vhost
https_port 443 accel cert=/etc/pki/tls/certs/domain.de.pem
key=/etc/pki/tls/private/domain.de.key.pem cafile=/etc/pki/CA/certs/domain.de.ca.pem
```

```
defaultsite=www.domain.de vhost

# Uncomment and adjust the following to add a disk cache directory.
cache_dir ufs /var/spool/squid 500 16 256
# Leave coredumps in the first cache dir
coredump_dir /var/spool/squid

# Add any of your own refresh_pattern entries above these.
#
refresh_pattern ^ftp:           1440    20%     10080
refresh_pattern ^gopher:        1440    0%      1440
refresh_pattern -i (/cgi-bin/|\?) 0     0%      0
refresh_pattern .               0       20%     4320

# First (HTTP) Peer
cache_peer [IPv6-Adresse des Zielrechners] parent 80 0 no-query originserver
login=PASS name=80

# Second (HTTPS) Peer
cache_peer [IPv6-Adresse des Zielrechners] parent 443 0 no-query originserver  ssl
sslflags=DONT_VERIFY_PEER login=PASS connection-auth=off name=443

#ACL proxy
acl proxy_acl dstdomain .domain.de .domain.com
http_access allow proxy_acl
cache_peer_access 443 allow proxy_acl
cache_peer_access 80 allow proxy_acl

# Additional ACL definitions
acl manager proto cache_object
acl purge method PURGE
acl CONNECT method CONNECT

# Restrictions
http_access allow manager localhost
http_access deny manager
http_access allow purge localhost
http_access deny purge
```

```
http_access deny all

# Disable caching
# cache deny all

# memory cache size
cache_mem 256 MB

# define hostname
visible_hostname proxy.domain.de

#logformat <name> <format specification>
logformat apache %{%d.%m %H:%M:%S}tl  %>a %Ss %ru

#access_log
access_log  /var/log/squid/access.log apache
```

```
cache_mgr webmaster@domain.de
```

Wichtig ist, dass alle SSL Zertifikate (Ziel-Server-Zertifikat, Ziel-Server-Zertifikat Schlüssel und CA Zertifikat) im Verzeichnis */etc/pki* des Ziel-Servers auf den Proxy-Server wieder exakt so in Kopie vorliegen. Die Pfadangaben und Namen der Zertifikat Dateien in der Zeile unten müssen auf dem Proxy Server korrekt vorliegen.

```
https_port 443 accel cert=/etc/pki/tls/certs/domain.de.pem
key=/etc/pki/tls/private/domain.de.key.pem cafile=/etc/pki/CA/certs/domain.de.ca.pem
defaultsite=www.domain.de vhost
```

Squid starten und testen:

```
[root@proxy ~]# systemctl start squid
```