Sophos

- Zertifikate
  - Automatisiert Let's Encrypt Zertifikat erstellen und auf die Sophos hochladen

Zertifikate

Automatisiert Let's Encrypt Zertifikat erstellen und auf die Sophos hochladen

Einleitung

In diesem kurzen Artikel, beschreibe ich kurz, wie wir mithilfe eines **Linux Servers / LXC Containers** automatisiert **Let's Encrypt** Zertifikate erstellen können, und diese auf unsere **Sophos XG Firewall** über die API hochladen können.

Durchführung

API Zugriff aktivieren

Im ersten Schritt müssen wir überprüfen, ob der API-Zugriff auf unsere Firewall aktiviert ist. Dies können wir im **Admin-Portal** unter `Sicherung & Firmware / API` überprüfen.

Dort muss der Haken bei "*API Konfiguration*" gesetzt sein und die entsprechende IP-Adresse des Linux Servers muss eingetragen sein. Damit legen wir fest, dass die **API-Schnittstelle** generell offen ist, und der Rechner mit der entsprechenden IP-Adresse auf die API zugreifen darf.

API Benutzer anlegen

Um auf die API zugreifen zu können, brauchen wir auch einen entsprechenden Benutzer. Dafür legen wir einen Benutzer an. In meinem Beispiel heißt der Benutzer **api-acme**. Der Benutzer wird als **Administrator** mit dem **Crypto-Admin** Profil angelegt.

ACME installieren

Um jetzt den ACME-Client zu installieren, führen wir die folgenden Befehle aus:

```
apt install git curl -y
git clone https://github.com/Neilpang/acme.sh.git
cd ./acme.sh
```

In dem Verzeichnis angekommen führen wir das Skript aus und geben unsere E-Mail-Adresse an:

```
./acme.sh --install -m mail@mail.com
```

Nun brauchen wir einen API-Token von Cloudflare. Diesen können wir hier erstellen. Der API-Token braucht "*Zonen DNS Editierungsrechte*" und "*Zonen Zone Leserechte*".

Diesen Token speichern wir in einer Umgebungsvariable:

```
export CF_Token=APITOKEN CLOUDFLARE
```

Zertifikat anfordern

Jetzt fordern wir das Zertifikat an. Dazu führen wir den folgenden Befehl aus und passen natürlich noch unsere Domain an.

```
acme.sh --issue --dns dns_cf --oscp-must-staple --keylength 4096 -d domain.de -d '*.domain.de'
```

Das Skript rattert jetzt alles durch, das kann etwas dauern. Nach einer Zeit bekommen wir die Information, in welchen Pfaden sich die Zertifikatsdateien befinden.

Zertifikat hochladen

Um das Zertifikat hochzuladen, erstellen wir jetzt ein Verzeichnis und fügen dort den Inhalt der Skripte ein:

```
mkdir le2xg
cd le2xg
curl https://raw.githubusercontent.com/laitco/Snippets/main/SophosXG/le2xg.sh > le2xg.sh
curl https://raw.githubusercontent.com/laitco/Snippets/main/SophosXG/xgxml.txt > xgxml.txt
```

Wenn alles geklappt hat, müssen wir jetzt in der Sophos XG unter Zertifikate unsere Zertifikate jetzt sehen können.

> **Info:** In dem Skript müssen noch einige Informationen angepasst werden. Die Variablen für die **Router-IP**, **API-User**, **APIPlainPass** und die **LEDOMAIN** muss angepasst werden.

Da ich die Skripte nicht im **root Verzeichnis** abgelegt habe, musste das Skript bei mir etwas angepasst werden. Die Skripte sehen bei mir wie folgt aus:

le2xg.sh

```
#!/bin/bash
#Copied from https://github.com/mmccarn/sophos and done some enchancements for Wildcard Certs
# router address and port as seen from the system running Letsencrypt
ROUTER=<Router-IP>:4444

# the system where the LetsEncrypt is running
#
# 1. the 'admin' account has full api access,
#    or you can create a dedicated api user
# 2. 'api' access must be enabled at
#    Administration -> Backup & Firmware -> API
#      - Enable 'API Configuration'
#      - enter the IP addresses that should be allowed to access the API
APIUSER=api-acme
APIPLAINPASS=<Passwort>

# Complete path to xgxml.txt
XML=/opt/domain.de/le2xg/xgxml.txt

# Letsencrypt domain
# look in /etc/letsencrypt/live
LEDOMAIN=domain.de

# Letsencrypt CertificateAuthority
# CA will be created in Sophos as ${LECertAuth}-yyyymmdd
LECertAuth=LetsEncrypt-CA

# cert date
CERTDATE=$(find /root/.acme.sh/${LEDOMAIN}/${LEDOMAIN}.key -printf "%CY%Cm%Cd\n")

# XG Operation
#     add: this must be used once to initiate the certificate on the XG
#  update: this is used for updating the cert once it has been created
OPERATION=${1:-add}

# Overview -
# 1. copy & rename letsencrypt 'privkey.pem' to 'privkey.key'
# 2. replace placeholder variables in 'xgxml.txt' with values above
# 3. feed the result to curl
#    - listing the 3 files to be uploaded in the order they occur in the input
```

```
# 4. Delete the copy of privkey.pem that was created

cp /root/.acme.sh/domain.de/phillipunzen.de.key ./privkey.key
cp /root/.acme.sh/domain.de/ca.cer ./chain.pem
cp /root/.acme.sh/domain.de/fullchain.cer ./fullchain.pem

sed \
  -e "s/APIUSER/$APIUSER/" \
  -e "s/APIPLAINPASS/$APIPLAINPASS/" \
  -e "s/OPERATION/$OPERATION/" \
  -e "s/LEDOMAIN/$LEDOMAIN-$CERTDATE/" \
  -e "s/LECertAuth/$LECertAuth-$CERTDATE/" /opt/domain.de/le2xg/xgxml.txt \
| curl -k -F "reqxml=<-" \
  -F file=@./chain.pem \
  -F file=@./fullchain.pem \
  -F file=@./privkey.key \
  "https://$ROUTER/webconsole/APIController?"

 rm ./privkey.key
 rm ./chain.pem
 rm ./fullchain.pem
```

xgxml.txt

```
<?xml version="1.0" encoding="UTF-8"?>
<Request APIVersion="1702.1">

<!-- Authenticate -->
  <Login>
    <Username>APIUSER</Username>
    <Password passwordform="plain">APIPLAINPASS</Password>
  </Login>

<!-- Upload Letsencrypt Certificate Authority -->
  <Set operation="OPERATION">
    <CertificateAuthority>
      <Name>LECertAuth</Name>
      <Format>pem</Format>
      <CACertFile>chain.pem</CACertFile>
      <CAPrivateKeyFile></CAPrivateKeyFile>
```

```
      <Password></Password>
    </CertificateAuthority>
  </Set>

<!-- Upload Cert -->
  <Set operation="OPERATION">
    <Certificate>
      <Action>UploadCertificate</Action>
      <Name>LEDOMAIN</Name>
      <CertificateFormat>pem</CertificateFormat>
      <CertificateFile>fullchain.pem</CertificateFile>
      <PrivateKeyFile>privkey.key</PrivateKeyFile>
    </Certificate>
  </Set>
</Request>
```

Zertifikat automatisch erneuern

Damit die Zertifikate automatisch erneuert werden, müssen wir nur einen **Cron-Job** hinterlegen. Dazu führen wir den folgenden Befehl aus und passen die Ausgabe an:

```
crontab -e
```

```
35 0 * * * sh /opt/phillipunzen.de/le2xg/le2xg.sh > /dev/null
```