FR

Let's Encrypt et ACME sur OPNsense

Certificats TLS gratuits pour tous vos services internes

Serie Stack OPNsense BOTUM — Billet 14/15

Mars 2026

BILLET 14

Sommaire

Let's Encrypt et ACME sur OPNsense : certificats TLS gratuits pour tous vos services internes

Dans ce billet 14 de la Serie Stack OPNsense BOTUM, nous mettons en place **Let's Encrypt** via le protocole ACME directement sur OPNsense. Objectif : chaque service interne (Grafana, Wazuh, AdGuard, HAProxy) dispose d'un certificat TLS valide, renouvel automatiquement, sans ouvrir le port 80.

1. Pourquoi TLS meme pour les services internes ?

La plupart des administrateurs negligent TLS sur le reseau interne : *"personne ne peut intercepter le trafic LAN de toute facon"*. C'est une erreur.

Raisons critiques :

• **HSTS (HTTP Strict Transport Security)** : Les navigateurs modernes refusent les connexions HTTP vers des domaines ayant deja repondu en HTTPS. Sans TLS interne, vos services deviennent inaccessibles apres une premiere visite HTTPS.

• **Confiance navigateur** : Les alertes de securite 'certificat invalide' forment les utilisateurs a ignorer les avertissements — ce qui les rend vulnerables aux vraies attaques.

• **Zero-Trust interne** : Un attaquant sur le VLAN IoT (Episode 2) peut intercepter le trafic non chiffre entre services. TLS interne + MTLS = defense en profondeur.

• **APIs et webhooks** : Grafana, Wazuh, AdGuard Home exposent des APIs. Sans TLS, les tokens d'authentification circulent en clair.

• **Let's Encrypt = gratuit** : Aucune raison economique de ne pas l'utiliser. Le plugin ACME d'OPNsense automatise tout.

2. Installer le plugin os-acme-client sur OPNsense

Le plugin **os-acme-client** est disponible directement dans le gestionnaire de plugins OPNsense. Il integre le client ACME acme.sh et gere nativement Let's Encrypt.

```
# Via l'interface Web OPNsense :
# System > Firmware > Plugins > rechercher "acme"
# Installer : os-acme-client
# Redemarrer OPNsense apres installation

# Verification via SSH OPNsense :
pkg info | grep acme
# -> os-acme-client-x.x_x   ACME client plugin

# Le menu apparait dans :
# Services > ACME Client > Settings / Accounts / Certificates / Automations
```

Apres installation, creer un compte ACME (adresse email pour les notifications d'expiration) :

```
# Services > ACME Client > Accounts > Add
Name        : botum-letsencrypt
E-Mail      : admin@botum.ca
ACME Server : Let's Encrypt (production)
              https://acme-v02.api.letsencrypt.org/directory
# Note : utiliser "staging" pour les tests (limite 5 certificats/semaine en prod)
```

3. Valider le domaine : challenge DNS sans port 80

La validation **DNS-01** est la seule methode qui fonctionne sans exposer le port 80 au public — indispensable pour les services internes.

3.1 Cloudflare (recommande)

```
# 1. Creer un token API Cloudflare (minimal permissions) :
# Cloudflare Dashboard > My Profile > API Tokens > Create Token
# Permissions : Zone > DNS > Edit
# Zone Resources : Include > Specific zone > botum.ca
# Copier le token genere

# 2. OPNsense > Services > ACME Client > Challenge Types > Add
Name          : cloudflare-dns
Challenge     : DNS-01
DNS Service   : Cloudflare
CF_Token      : [votre-token-api-cloudflare]
CF_Account_ID : [votre-account-id] (visible dans Cloudflare Dashboard > droite)

# 3. Alternative : CF_Key + CF_Email (ancienne methode Global API Key)
# Deconseille - preferer le token API scope minimal
```

3.2 OVH / Gandi

```
# OVH - Services > ACME Client > Challenge Types > Add
Name         : ovh-dns
Challenge    : DNS-01
DNS Service  : OVH
OVH_AK       : [Application Key depuis api.ovh.com]
OVH_AS       : [Application Secret]
OVH_CK       : [Consumer Key]
OVH_ENDPOINT : ovh-ca   # ou ovh-eu selon region

# Gandi
Name         : gandi-dns
Challenge    : DNS-01
DNS Service  : Gandi LiveDNS
GANDI_LIVEDNS_KEY : [cle API Gandi]

# Generer la cle API Gandi :
# account.gandi.net > Security > API Keys > Create
```

4. Generer un certificat wildcard *.botum.ca

Un certificat **wildcard** couvre tous les sous-domaines : *grafana.botum.ca*, *wazuh.botum.ca*, *adguard.botum.ca*, etc. Un seul certificat, renouvement centralise.

```
# Services > ACME Client > Certificates > Add
Name               : wildcard-botum-ca
Common Name        : *.botum.ca
Challenge Type     : cloudflare-dns  (cree section 3)
Key Length         : 4096 bit (ou ec-384 pour ECDSA)
OCSP Must-Staple   : desactive (optionnel)
Auto Renew         : activer
Renew before (d)   : 30  (renouveler 30j avant expiration)

# Apres creation, cliquer "Issue / Renew"
# Le plugin ACME va :
#  1. Contacter Let's Encrypt
#  2. Creer un enregistrement TXT DNS _acme-challenge.botum.ca via Cloudflare API
#  3. Let's Encrypt valide le TXT record
#  4. Certificat + cle privee telecharges et stockes dans OPNsense
#  5. Enregistrement TXT supprime automatiquement

# Verification :
# System > Trust > Certificates
# -> wildcard-botum-ca doit apparaitre avec expiration ~90j
```

Ajouter egalement un certificat pour le domaine racine si necessaire :

```
# Certificat additionnel pour botum.ca (sans wildcard)
Common Name : botum.ca
# Ajouter dans Alt Names : www.botum.ca
# Meme challenge type Cloudflare
```

5. Distribuer le certificat : OPNsense, HAProxy, Nginx, Docker

5.1 OPNsense Web UI (admin panel)

```
# System > Settings > Administration
SSL Certificate : wildcard-botum-ca
-> L'interface admin OPNsense passe en HTTPS valide
-> Accessible via https://fw.botum.ca/ sans avertissement navigateur
```

5.2 HAProxy (Billet 5 - reverse proxy)

```
# Services > HAProxy > Settings
# Dans chaque Frontend HTTPS :
SSL Certificates : wildcard-botum-ca
SSL offloading   : activer
# HAProxy gere le TLS termination pour tous les services derriere

# Exemple frontend Grafana :
# Name : grafana-https
# Bind : 0.0.0.0:443 (VLAN Servers)
# SSL  : wildcard-botum-ca
# Backend : grafana-backend (192.168.20.x:3000)
```

5.3 Nginx (proxy inverse interne)

```
# Sur le serveur Nginx, via SFTP/SCP depuis OPNsense :
# OPNsense stocke les certificats dans :
# /var/etc/acme-client/home/[nom-cert]/

# Automatisation via action ACME post-renouvellement :
# Services > ACME Client > Automations > Add
Name    : deploy-to-nginx
Run Command : /usr/local/sbin/deploy_cert_nginx.sh

# /usr/local/sbin/deploy_cert_nginx.sh
#!/bin/sh
CERT_SRC="/var/etc/acme-client/home/wildcard-botum-ca"
NGINX_HOST="192.168.20.5"
scp $CERT_SRC/fullchain.cer admin@$NGINX_HOST:/etc/nginx/ssl/botum.ca.crt
scp $CERT_SRC/*.key         admin@$NGINX_HOST:/etc/nginx/ssl/botum.ca.key
ssh admin@$NGINX_HOST "nginx -t && nginx -s reload"
echo "Nginx cert deployed: $(date)" >> /var/log/acme_deploy.log
```

5.4 Services Docker

```
# Monter le certificat dans les containers via volume Docker
# Sur le host Docker (us-srv-dck-01) :

# docker-compose.yml - exemple Grafana avec TLS
services:
  grafana:
    image: grafana/grafana:latest
    volumes:
      - /opt/certs/botum.ca.crt:/etc/grafana/ssl/cert.pem:ro
      - /opt/certs/botum.ca.key:/etc/grafana/ssl/key.pem:ro
    environment:
      GF_SERVER_PROTOCOL: https
      GF_SERVER_CERT_FILE: /etc/grafana/ssl/cert.pem
      GF_SERVER_CERT_KEY: /etc/grafana/ssl/key.pem

# Script de synchronisation des certificats (cron sur Docker host) :
# /opt/scripts/sync_certs.sh
#!/bin/bash
SRC="admin@192.168.10.1:/var/etc/acme-client/home/wildcard-botum-ca"
scp -i /root/.ssh/id_ed25519 $SRC/fullchain.cer /opt/certs/botum.ca.crt
scp -i /root/.ssh/id_ed25519 $SRC/*.key         /opt/certs/botum.ca.key
chmod 640 /opt/certs/botum.ca.key
docker compose -f /opt/stacks/monitoring/docker-compose.yml restart grafana wazuh
echo "Certs synced: $(date)" >> /var/log/cert_sync.log

# Crontab (renouvellement auto J+1 du renouvellement ACME) :
# 0 3 * * * /opt/scripts/sync_certs.sh >> /var/log/cert_sync.log 2>&1
```

6. Renouvellement automatique (cron ACME natif)

Le plugin os-acme-client integre un **cron natif** qui verifie et renouvelle les certificats automatiquement. Let's Encrypt emet des certificats valides **90 jours** ; le renouvellement se declenche 30 jours avant l'expiration.

```
# Activer le cron ACME :
# Services > ACME Client > Settings
# [x] Enable Cron Job
# Run interval : Daily (recommande)
# Scheduled at : 03:30 (nuit pour limiter impact)

# Ce que fait le cron automatiquement :
# 1. Verifie tous les certificats configures
# 2. Si expiration < 30 jours -> renouvellement automatique
# 3. Effectue le challenge DNS-01 (Cloudflare/OVH/Gandi)
# 4. Telecharge le nouveau certificat
# 5. Declenche les Automations post-renouvellement
# 6. Recharge OPNsense avec le nouveau certificat

# Verification des logs ACME :
# Services > ACME Client > Log File
# ou SSH OPNsense :
tail -100 /var/log/acme.log | grep -E "SUCCESS|ERROR|Renew"

# Test manuel du renouvellement (sans vraiment renouveler) :
# Services > ACME Client > Certificates > [cert] > Simulate
# Ou forcer un renouvellement immediatement :
# [bouton] Issue / Renew
```

7. Alerting sur expiration (integration monitoring Billet 9)

Meme avec renouvellement automatique, une alerte proactive sur les expirations est indispensable — le renouvellement peut echouer (API DNS indisponible, rate limit Let's Encrypt, etc.).

7.1 Script de monitoring des certificats

```
#!/bin/bash
# /opt/scripts/check_cert_expiry.sh
# Verifie l'expiration et alerte via Telegram

TELEGRAM_TOKEN="[votre-token]"
TELEGRAM_CHAT="-1001234567890"
WARN_DAYS=14
CRIT_DAYS=7

alert_telegram() {
    curl -s "https://api.telegram.org/bot${TELEGRAM_TOKEN}/sendMessage"          -d "chat_id=${TELEGRAM_CHAT}"          -
}

# Verifier les certs sur OPNsense
for cert in grafana.botum.ca wazuh.botum.ca adguard.botum.ca fw.botum.ca; do
    EXPIRY=$(echo | openssl s_client -connect ${cert}:443 -servername ${cert} 2>/dev/null              | openssl x509 -no
    if [ -z "$EXPIRY" ]; then
        alert_telegram "CERT ALERT: Impossible de verifier ${cert}"
        continue
    fi
    EXPIRY_EPOCH=$(date -d "$EXPIRY" +%s 2>/dev/null || date -j -f "%b %d %H:%M:%S %Y %Z" "$EXPIRY" +%s)
    NOW_EPOCH=$(date +%s)
    DAYS_LEFT=$(( (EXPIRY_EPOCH - NOW_EPOCH) / 86400 ))

    if [ $DAYS_LEFT -le $CRIT_DAYS ]; then
        alert_telegram "CRITIQUE: Cert ${cert} expire dans ${DAYS_LEFT} jours !"
    elif [ $DAYS_LEFT -le $WARN_DAYS ]; then
        alert_telegram "WARNING: Cert ${cert} expire dans ${DAYS_LEFT} jours"
    fi
    echo "${cert}: ${DAYS_LEFT} jours restants"
done

# Crontab (tous les jours a 08h00) :
# 0 8 * * * /opt/scripts/check_cert_expiry.sh
```

7.2 Integration Prometheus / Grafana (Billet 9)

```
# Exporter les metriques de certificats via ssl_exporter
# docker-compose.yml
services:
  ssl-exporter:
    image: ribbybibby/ssl-exporter:latest
    ports:
      - "9219:9219"
    volumes:
      - ./ssl_targets.yml:/ssl_targets.yml:ro

# ssl_targets.yml
- targets:
    - grafana.botum.ca:443
    - wazuh.botum.ca:443
    - adguard.botum.ca:443
    - fw.botum.ca:443

# prometheus.yml - ajouter scrape config
scrape_configs:
  - job_name: ssl_certs
    metrics_path: /probe
    relabel_configs:
      - source_labels: [__address__]
        target_label: __param_target
    static_configs:
      - targets:
          - grafana.botum.ca:443
          - wazuh.botum.ca:443

# Alerte Prometheus (rules/certs.yml)
groups:
  - name: cert-expiry
    rules:
      - alert: CertExpiryWarning
        expr: ssl_cert_not_after - time() < 86400 * 14
        labels: { severity: warning }
        annotations:
          summary: "Cert {{ $labels.target }} expire bientot"
      - alert: CertExpiryCritical
        expr: ssl_cert_not_after - time() < 86400 * 7
        labels: { severity: critical }
```

8. Cas d'usage : AdGuard DoH, Grafana, Wazuh Dashboard

Les certificats Let's Encrypt ameliorent directement trois services cles de notre stack.

8.1 AdGuard Home DNS-over-HTTPS (Billet 8)

AdGuard Home supporte nativement DoH avec TLS. Avec le certificat wildcard, activer HTTPS sur adguard.botum.ca :

```
# AdGuard Home > Settings > Encryption Settings
Enable encryption : Yes
Server name       : adguard.botum.ca
HTTPS port        : 443
DNS-over-HTTPS    : https://adguard.botum.ca/dns-query
TLS certificate   : /etc/adguard/ssl/botum.ca.crt  (synced from OPNsense)
Private key       : /etc/adguard/ssl/botum.ca.key

# Sur les clients (navigateurs, iOS, Android) :
# DNS-over-HTTPS provider : https://adguard.botum.ca/dns-query
# Chiffrement bout-en-bout sans avertissement certificat
```

8.2 Grafana (Billet 9)

```
# grafana.ini (ou variables d'env Docker) :
[server]
protocol         = https
cert_file        = /etc/grafana/ssl/cert.pem
cert_key         = /etc/grafana/ssl/key.pem
domain           = grafana.botum.ca
root_url         = https://grafana.botum.ca/
enforce_domain   = true

# Resultat : https://grafana.botum.ca -> cadenas vert
# HSTS automatique -> impossible de downgrader en HTTP
# Les API calls Prometheus -> Grafana chiffres
```

8.3 Wazuh Dashboard (Billet 11)

```
# /etc/wazuh-dashboard/opensearch_dashboards.yml
server.host: 0.0.0.0
server.port: 5601
server.ssl.enabled: true
server.ssl.certificate: /etc/wazuh-dashboard/certs/wazuh-dashboard.pem
server.ssl.key: /etc/wazuh-dashboard/certs/wazuh-dashboard-key.pem

# Remplacer les certs auto-signes Wazuh par le wildcard botum.ca :
# Copier botum.ca.crt -> wazuh-dashboard.pem
# Copier botum.ca.key -> wazuh-dashboard-key.pem
# Redemarrer : systemctl restart wazuh-dashboard
# Resultat : https://wazuh.botum.ca:5601 -> certificat valide
```

9. Conclusion : un stack TLS unifie et automatise

Avec le billet 14, **tous les services du Stack OPNsense BOTUM sont securises par un certificat TLS valide**, renouvel automatiquement, sans aucune intervention manuelle :

- **os-acme-client** — Plugin OPNsense natif, integration Let's Encrypt
- **DNS-01 challenge** — Validation sans port 80, support Cloudflare/OVH/Gandi
- **Wildcard *.botum.ca** — Un certificat pour tous les sous-domaines
- **HAProxy + Nginx** — TLS termination centralise
- **Docker services** — Certificats synchronises automatiquement
- **Cron ACME** — Renouvellement automatique 30j avant expiration
- **Alerting Telegram + Prometheus** — Surveillance proactive des expirations

Prochain billet (15/15) : Netflow + ntopng sur OPNsense — analyse de trafic reseau en temps reel, visualisation des flux par application, detection d'anomalies. Le dernier billet de la serie !

Article complet : blog.botum.ca/opnsense-lets-encrypt-acme-certificats

Site web : www.botum.ca • contact@botum.ca