Mode Opératoire

Certificats ADCS des services émission, déploiement et renouvellement

Code : MO-SEC-005
Version : 1.0
Date : 14 septembre 2026
Auteur : Cédric LEGRAND
Classification : USAGE INTERNE — Équipe BTS SIO

Historique des révisions

Version	Date	Modifications
1.0	14/09/2026	Création initiale — formalisation du chantier du 14/09/2026 : émission et déploiement des certificats « BTS SIO Root CA » sur Traefik (*wildcard*) et six hôtes ou accès directs.

1 Objet

Ce mode opératoire décrit l'émission, le déploiement et le renouvellement des certificats TLS des services web de l'infrastructure BTS SIO, signés par l'autorité de certification d'entreprise **« BTS SIO Root CA »** (AD CS, sur DC1). L'objectif est d'**éliminer les alertes de certificat** dans les navigateurs : la racine étant publiée dans l'Active Directory, tout poste joint au domaine lui accorde sa confiance nativement.

Le document couvre deux situations :

— l'**émission initiale** d'un certificat pour un service (nouveau service, nouvel hôte) ;

— le **renouvellement annuel** des certificats existants (validité d'un an, expiration le 14/09/2027 pour les certificats émis lors du chantier initial).

La vue d'ensemble de l'architecture (chaîne de confiance, décisions, état des lieux, supervision) relève du **MO-SEC-006**, document compagnon du présent MO.

2 Champ d'application

Public concerné	Administrateurs de l'infrastructure BTS SIO
Autorité	« BTS SIO Root CA » sur `SRV2022.bts.sio` (DC1, `10.0.112.2`), RSA 4096, expire le 07/09/2036
Gabarit ADCS	`BTSSIOWebServer1an` (« BTS SIO Web Server 1 an »), validité 1 an, clé min. 2048, exportable, EKU *Server Authentication*
Outils	OpenSSL sur l'hôte cible, `certreq`/`certutil` sur DC1, `curl` pour les vérifications
Durée	15–30 min par certificat (émission + déploiement + vérification)
Présentiel requis	Non — tout se fait à distance (SSH, WinRM, interfaces web)

Le périmètre couvert est le suivant :

- **Traefik** (CT 200, `docker-srv`) : un *wildcard* `*.docker.bts.sio` sert les onze FQDN du reverse proxy et l'accès de secours `https://10.0.112.20` ;
- **Portainer en accès direct** `:9443` : réutilise le même *wildcard* (même IP) ;
- **cinq hôtes ou accès directs**, un certificat chacun :
  - Proxmox (`https://10.0.112.200:8006`) ;
  - OPNsense (`https://10.0.112.1`) ;
  - Vaultwarden (`https://10.0.112.10`) ;
  - Wazuh (`https://wazuh.bts.sio`) ;
  - contrôleur UniFi en direct (`https://10.0.112.228:8443`).

Accès de secours inchangés

Les accès de secours HTTP par `IP:port` (LAN) et les accès VPN restent inchangés : ce MO ne modifie aucun flux pare-feu ni aucune règle de routage.

3 Prérequis

Prérequis

- **Le gabarit `BTSSIOWebServer1an` est publié sur la CA et les réglages de la CA sont en place (§ 4) — sinon, exécuter le script versionné `files/adcs-template.ps1`**

- **Un accès administrateur à DC1 (session locale, RDP ou WinRM) avec le compte `BTS\Administrateur`**
- **OpenSSL disponible sur l'hôte cible (cas général) ; pour OPNsense, le CSR est préparé sur le poste d'administration (§ 6.4)**
- **Le certificat racine `BTS-SIO-Root-CA.crt` disponible localement (export ADCS, également en pièce jointe de l'item CA dans Vaultwarden)**
- **Identifiants d'administration des services concernés, lus dans Vaultwarden le moment venu**

Postes hors domaine

Sur un poste non joint au domaine, la racine doit être importée une fois : sous Linux, copier `BTS-SIO-Root-CA.crt` dans `/usr/local/share/ca-certificates/` puis `sudo update-ca-certificates` ; sous Windows, importer dans le magasin « Autorités de certification racines de confiance ». Firefox dispose de son propre magasin : y importer la racine séparément (Paramètres → Vie privée et sécurité → Certificats → Afficher les certificats → Autorités → Importer, cocher « identifier des sites web »).

4 Infrastructure ADCS (rappel)

4.1 Autorité de certification

Nom	« BTS SIO Root CA » (autorité entreprise racine, auto-signée)
Hôte	DC1 — `SRV2022.bts.sio` (`10.0.112.2`), Windows Server 2022
Config string	`SRV2022.bts.sio\BTS SIO Root CA` (espaces comprises, à reprendre à l'identique)
Clé	RSA 4096
Expiration	07/09/2036
Distribution	Publiée dans l'AD → confiance native des postes du domaine (aucune GPO requise)

4.2 Gabarit dédié `BTSSIOWebServer1an`

Le gabarit `BTSSIOWebServer1an` (displayName « BTS SIO Web Server 1 an ») est une **duplication du gabarit intégré « Web Server »**, avec les adaptations suivantes :

— **validité 1 an** (`pKIExpirationPeriod` = −315576000000000) au lieu de 2 ans — la limite acceptée par les navigateurs est de 398 jours ;

— clé privée **exportable** autorisée (nécessaire à l'import OPNsense et au keystore UniFi) ;

— taille de clé minimale 2048 ; EKU *Server Authentication* ; le sujet est fourni par le demandeur (`ENROLLEE_SUPPLIES_SUBJECT`) ;

— **ACE « Enroll » explicite** pour `BTS\Administrateur` sur ce gabarit uniquement — elle autorise les soumissions `certreq` scriptées sans déléguer l'inscription sur les autres gabarits, et se retire proprement si besoin.

4.3 Réglages de la CA

Réglage	Valeur et rôle
`policy\EditFlags`	`0x15014e` — inclut `EDITF_ATTRIBUTESUBJECTALTNAME2` (`0x40000`), ajouté le 14/09/2026 (valeur avant : `0x11014e`) : la CA accepte les SAN fournis dans le CSR
`CA\RequestExtensionList`	`2.5.29.37` — l'extension EKU (*extendedKeyUsage*) est reprise du CSR ; exigence d'OPNsense, qui refuse un certificat sans EKU *serverAuth* explicite

4.4 Mise en place initiale (script versionné)

L'intégralité de cette préparation (flag SAN, création du gabarit, ACE Enroll, publication, `RequestExtensionList`, redémarrage de `certsvc` et contrôles) est automatisée par le script **`files/adcs-template.ps1`**, versionné dans le dossier du présent MO et reproduit en annexe B. Il est idempotent et s'exécute sur DC1 dans une session PowerShell élevée :

```
powershell.exe -ExecutionPolicy Bypass -File adcs-template.ps1
```

Vérification

Après exécution du script (ou pour auditer l'état existant), trois contrôles doivent être concluants :

```
certutil -config "SRV2022.bts.sio\BTS SIO Root CA" -CATemplates
   -> la liste publiee contient BTSSIOWebServer1an
certutil -config "SRV2022.bts.sio\BTS SIO Root CA" -getreg
    policy\EditFlags
   -> la valeur inclut 0x40000 (constate : 0x15014e)
certutil -config "SRV2022.bts.sio\BTS SIO Root CA" -getreg
    CA\RequestExtensionList
   -> 2.5.29.37
```

5 Émettre un certificat

5.1 Préparer la configuration OpenSSL

Chaque cible dispose d'une configuration OpenSSL versionnée dans `files/` (reproduites en annexe C). Pour une cible nouvelle, copier le modèle le plus proche et adapter `CN` et `[ alt_names ]`.

Fichier (`files/`)	CN	SANs
`openssl-wildcard.cnf`	`docker.bts.sio`	DNS `docker.bts.sio`, `*.docker.bts.sio` ; IP `10.0.112.20`
`openssl-proxmox.cnf`	`10.0.112.200`	IP `10.0.112.200`
`openssl-opnsense.cnf`	`10.0.112.1`	IP `10.0.112.1`
`openssl-vaultwarden.cnf`	`10.0.112.10`	IP `10.0.112.10`
`openssl-wazuh.cnf`	`wazuh.bts.sio`	DNS `wazuh.bts.sio` ; IP `10.0.112.103`
`openssl-unifi.cnf`	`10.0.112.228`	DNS `unifi.docker.bts.sio` ; IP `10.0.112.228`

EKU `serverAuth` obligatoire

La section `[ req_ext ]` doit contenir `extendedKeyUsage = serverAuth` (grâce à `RequestExtensionList`, l'EKU du CSR est reprise dans le certificat émis). Les configurations `openssl-wildcard.cnf` et `openssl-proxmox.cnf` en sont dépourvues — elles reflètent l'émission du 14/09/2026, acceptée par Traefik et Proxmox : l'ajouter lors de leur renouvellement (retour d'expérience OPNsense, § 9).

1 Générer la clé privée et le CSR sur l'hôte cible

La clé privée est générée sur l'hôte qui l'utilisera et n'en sort jamais. Exemple pour le *wildcard* sur `docker-srv` (CT 200) :

```
mkdir -p /opt/docker/traefik/certs && chmod 700 /opt/docker/traefik/certs
cd /opt/docker/traefik/certs
openssl genrsa -out wildcard.key 4096
chmod 600 wildcard.key
openssl req -new -key wildcard.key -out wildcard.csr \
    -config openssl-wildcard.cnf
```

Adapter les chemins selon la cible (§ 6). Exception documentée : pour OPNsense, la clé est générée sur le poste d'administration puis importée via l'interface web, car l'appliance n'offre pas d'OpenSSL opérationnel pour préparer un CSR ; le fichier de clé est détruit du poste après import (§ 6.4).

2 Contrôler le CSR avant soumission

```
openssl req -in wildcard.csr -noout -subject -text \
  | grep -A1 -e 'Subject:' -e 'Subject Alternative Name'
```

Le sujet et les SAN (DNS et `IP Address`) doivent correspondre exactement à la cible. Tout écart se corrige dans le `.cnf` avant la

soumission.

3 Soumettre le CSR à la CA

Déposer le CSR sur DC1 (par exemple dans `C:\adcs\req.csr`), puis soumettre :

```
certreq -submit -config "SRV2022.bts.sio\BTS SIO Root CA" -attrib
    "CertificateTemplate:BTSSIOWebServer1an" req.csr cert.cer
```

Le `Request ID` attribué figure dans la sortie et dans le journal de la CA (`certutil -view`) : le noter dans les consignations du chantier.

Soumission automatisée via WinRM : options impératives

En exécution non interactive (WinRM/`pypsrp`), `certreq` peut geler sur une boîte modale. Utiliser systématiquement les options `-q -f` et purger au préalable `cert.cer` et `cert.rsp` (boîte de confirmation de remplacement sinon). En cas de gel malgré tout : identifier la demande dans `certutil -view`, récupérer le certificat avec `certreq -retrieve <ID>` — jamais de re-soumission (chaque re-soumission crée une entrée supplémentaire dans le journal : cf. les Request ID 7 et 8 inertes du 14/09/2026).

4 Valider le certificat émis

Rapatrier `cert.cer` sur l'hôte cible, puis :

```
# Si le fichier est en DER (pas de "-----BEGIN CERTIFICATE-----") :
openssl x509 -inform der -in cert.cer -out cert.crt

openssl verify -CAfile BTS-SIO-Root-CA.crt cert.crt
openssl x509 -in cert.crt -noout -dates -ext subjectAltName
```

Attendu : `OK` ; durée de validité ≤ 398 jours ; SANs conformes au

CSR. Conserver le certificat au format PEM, en `0600`.

6 Déployer le certificat (par cible)

Sauvegarde avant chaque remplacement

Avant de remplacer un certificat en place, copier le fichier d'origine en `<fichier>.bak-<date>` (convention du chantier initial : `.bak-https-20260914`). Le retour arrière consiste à restaurer la copie et à redémarrer le service concerné.

6.1 Traefik — *wildcard* en `defaultCertificate`

5 **Installer les fichiers et déclarer le certificat par défaut**

Fichiers : `/opt/docker/traefik/certs/wildcard.crt` et `wildcard.key` (0600) sur `docker-srv`.

1. **`docker-compose.yml` : ajouter le volume `./certs:/etc/traefik/certs:ro` au service Traefik ;**
2. **`dynamic.yml` : déclarer le certificat par défaut :**

```
tls:
  stores:
    default:
      defaultCertificate:
        certFile: /etc/traefik/certs/wildcard.crt
        keyFile: /etc/traefik/certs/wildcard.key
```

3. **`traefik.yml` : retirer le bloc `defaultGeneratedCert` (ancien *wildcard* auto-signé) ;**
4. **appliquer : `docker compose config -quiet && docker compose up -d` — un redémarrage du conteneur, coupure des**

FQDN de quelques secondes.

6.2 Portainer en accès direct (:9443)

Portainer réutilise le *wildcard* (même IP `10.0.112.20`, SAN IP inclus) : aucune émission supplémentaire.

6 Monter les certificats et redémarrer

Dans `/opt/docker/portainer/docker-compose.yml` : ajouter `command: -sslcert /certs/wildcard.crt -sslkey /certs/wildcard.key` et le volume `/opt/docker/traefik/certs:/certs:ro`, puis `docker compose up -d`. Pas de rechargement à chaud : le conteneur doit être recréé/redémarré pour prendre en compte un certificat renouvelé.

Flags dépréciés en Portainer 2.41

`-sslcert`/`-sslkey` sont dépréciés depuis la version 2.41 : migrer vers `-tlscert`/`-tlskey` lors d'une prochaine mise à jour de l'image.

6.3 Proxmox VE (10.0.112.200:8006)

7 Remplacer le certificat de pveproxy

Sur l'hôte PVE (clé et CSR préparés dans `/root/adcs/`) :

```
install -m 644 proxmox.crt /etc/pve/local/pveproxy-ssl.pem
install -m 600 proxmox.key /etc/pve/local/pveproxy-ssl.key
systemctl restart pveproxy
```

Le certificat seul suffit (la racine est connue des clients).

6.4 OPNsense (https://10.0.112.1)

8 **Importer puis sélectionner le certificat**

L'import se fait exclusivement par l'interface web :

1. **System → Trust → Certificates → + : coller le certificat PEM et la clé privée PEM, Save ;**
2. **System → Settings → Administration → *SSL Certificate* : sélectionner le nouveau certificat, Save (l'interface web recharge aussitôt).**

Détruire ensuite le fichier de clé privée du poste d'administration (exception de génération, § 5).

EKU exigée

OPNsense refuse un certificat sans EKU *serverAuth* : les deux premières émissions du 14/09/2026 (Request ID 7 et 8, CSR sans `extendedKeyUsage`) ont été refusées et n'ont jamais été déployées ; la troisième (Request ID 9), émise après ajout de l'EKU au `.cnf`, est en service. Vérifier la ligne `extendedKeyUsage = serverAuth` avant toute soumission pour cette cible.

6.5 Vaultwarden (https://10.0.112.10, CT 127)

9 **Remplacer le couple cert/clé et redémarrer**

La terminaison TLS est assurée par le conteneur (`ROCKET_TLS`) avec les fichiers de `/opt/vaultwarden/ssl/` :

```
cd /opt/vaultwarden/ssl
# cert.pem <- certificat ADCS signe ; key.pem generee sur place (0600)
chmod 600 key.pem
cd /opt/vaultwarden && docker compose restart vaultwarden
```

Vérifier ensuite un accès de bout en bout (`bw status` ou connexion web) : le coffre est la source des secrets de l'infrastructure.

6.6 Wazuh dashboard (`https://wazuh.bts.sio`, CT 103)

10 Remplacer les certificats du dashboard

Fichiers de `/etc/wazuh-dashboard/certs/` : déposer le certificat signé en `wazuh-dashboard.pem` et la clé en `wazuh-dashboard-key.pem`, puis :

```
chmod 400 /etc/wazuh-dashboard/certs/wazuh-dashboard*.pem
chown wazuh-dashboard:wazuh-dashboard \
    /etc/wazuh-dashboard/certs/wazuh-dashboard*.pem
systemctl restart wazuh-dashboard
systemctl is-active wazuh-dashboard
```

Ne pas toucher à la PKI interne Wazuh

Le `root-ca.pem` présent dans le même dossier est la CA interne de l'indexer Wazuh : il n'a rien à voir avec la chaîne servie aux navigateurs et ne doit pas être modifié.

6.7 Contrôleur UniFi en accès direct (`10.0.112.228:8443`)

11 Importer dans le keystore Java

Le certificat vit dans le keystore Java `/opt/docker/unifi/config/data/keystore` (mot de passe `aircontrolenterprise`, alias `unifi`) :

```
cd /opt/docker/unifi/adcs
```

```
openssl pkcs12 -export -out unifi.p12 -inkey unifi.key -in unifi.crt \
    -name unifi -password pass:aircontrolenterprise
docker cp unifi.p12 unifi:/tmp/
docker exec unifi keytool -importkeystore \
    -srckeystore /tmp/unifi.p12 -srcstoretype PKCS12 \
    -srcstorepass aircontrolenterprise \
    -destkeystore /config/data/keystore \
    -deststorepass aircontrolenterprise -destkeypass
   aircontrolenterprise \
    -alias unifi -noprompt
chown 1000:1000 /opt/docker/unifi/config/data/keystore
docker restart unifi
```

Le `chown 1000:1000` après coup est indispensable (le conteneur tourne sous l'UID 1000). Si `keytool` est absent de l'image, exécuter celui d'une image `eclipse-temurin` jetable montant le dossier de données. En cas d'interface cassée après import : restaurer le keystore sauvegardé et redémarrer le conteneur.

7 Renouvellement annuel

Les certificats émis le 14/09/2026 expirent le **14/09/2027**. Le renouvellement suit exactement la procédure d'émission initiale, avec les mêmes gabarit et configurations.

12 **Anticiper l'échéance**

La supervision d'expiration par Uptime Kuma est en place depuis le 14/09/2026 : notification d'expiration activée (J-30) sur les moniteurs HTTPS (identifiants régénérés et consignés au coffre, cf. MO-SEC-006). Un jalon calendrier à mi-août 2027 reste recommandé en garde-fou pour le renouvellement du lot.

13 Réémettre chaque certificat

Pour chaque cible du tableau du § 5.1 :

1. **ajouter** `extendedKeyUsage = serverAuth` **à** `openssl-wildcard.cnf` **et** `openssl-proxmox.cnf` **(les quatre autres l'ont déjà) ;**
2. **régénérer une clé nouvelle et le CSR (§ 5), soumettre, valider ; noter les nouveaux** `Request ID` **;**
3. **redéployer selon le § 6 en redémarrant chaque service concerné (Traefik, Portainer,** `pveproxy`**, Vaultwarden,** `wazuh-dashboard`**, conteneur UniFi ; OPNsense recharge à la sélection) ;**
4. **purger les fichiers de l'ancien certificat une fois la vérification du § 8 passée.**

Renouvellement groupé

Tous les certificats du chantier partagent la même date d'expiration : traiter le lot en une seule session (2–3 h) plutôt qu'au fil de l'eau, pour ne pas multiplier les fenêtres de redémarrage.

8 Vérification

14 Contrôler la chaîne présentée

```
openssl s_client -connect homer.docker.bts.sio:443 \
    -servername homer.docker.bts.sio </dev/null 2>/dev/null \
  | openssl x509 -noout -issuer -subject -ext subjectAltName
```

Attendu : émetteur `CN = BTS SIO Root CA`**, sujet et SANs conformes.**

15 Exécuter la batterie d'acceptation

La batterie `check_https.sh` (reproduite en annexe A) teste les onze FQDN Traefik et `https://10.0.112.20` avec `curl -cacert` (jamais `-k`) : douze lignes `OK` et code de sortie 0 sont attendus. Compléter par un `curl -cacert` sur chaque hôte hors Traefik : `https://10.0.112.200:8006`, `https://10.0.112.1`, `https://10.0.112.10`, `https://wazuh.bts.sio`, `https://10.0.112.20:9443`, `https://10.0.112.228:8443`.

Vérification

Avant de clôturer une émission ou un renouvellement :

- ☐ `openssl verify -CAfile` **retourne** `OK`
- ☐ **Batterie 12/12** `OK` **(cibles Traefik) et** `curl -cacert` **OK sur chaque hôte direct**
- ☐ **Navigateur d'un poste du domaine sans alerte sur au moins deux services**
- ☐ **Accès de secours HTTP par** `IP:port` **(LAN) toujours fonctionnels**
- ☐ `Request ID` **noté ; clés privées en** `0600` **(ou** `0400` **pour Wazuh), jamais hors de leur hôte**

9 Dépannage

Symptôme	Solution
`certreq` : « template not found » / gabarit refusé	Vérifier la publication : `certutil -CATemplates` avec la config string habituelle (§ 4.4). Republier avec `-SetCATemplates +BTSSIOWebServer1an`. Contrôler l'ACE Enroll de `BTS\Administrateur` sur le gabarit.
SAN absents du certificat émis	Le flag `EDITF_ATTRIBUTESUBJECTALTNAME2` (`0x40000`) n'est pas actif : `certutil -getreg policy\EditFlags` doit inclure `0x40000` (constaté : `0x15014e`). Sinon : `-setreg policy\EditFlags +262144` puis `Restart-Service certsvc`.
EKU absente, OPNsense refuse le certificat	Ajouter `extendedKeyUsage = serverAuth` dans `[ req_ext ]` du `.cnf`, vérifier `RequestExtensionList = 2.5.29.37`, ré-émettre (cf. Request ID 7 et 8, inertes).
`certreq` gèle en session WinRM	Boîte modale bloquante : relancer avec `-q -f` après purge de `cert.cer` et `cert.rsp`. Si la demande est partie : `certutil -view` puis `certreq -retrieve <ID>` — ne jamais re-soumettre.
`cert.cer` illisible / erreur PEM	Le fichier est en DER : `openssl x509 -inform der -in cert.cer -out cert.crt`.
Traefik redémarre en boucle après modification	Restaurer les `.bak` des trois fichiers (`docker-compose.yml`, `traefik.yml`, `dynamic.yml`), `docker compose up -d`, analyser `docker logs traefik`.
Wazuh dashboard ne repart pas	Droits attendus : `400`, propriétaire `wazuh-dashboard:wazuh-dashboard` sur les deux `.pem`. Sinon restaurer le `.bak` et redémarrer.
UniFi : UI cassée après import keystore	Restaurer `keystore.bak-<date>` dans `/opt/docker/unifi/config/data/`, `chown 1000:1000`, `docker restart unifi`.
Firefox signale encore le certificat	Firefox ignore le magasin système : importer la racine dans son magasin propre (§ 3).

10 Voir aussi

— **MO-SEC-006** — Architecture PKI interne et HTTPS des services (chaîne de confiance, décisions, état des lieux au 14/09/2026, supervision)
— **MO-PLT-012** — Se connecter à Uptime Kuma (supervision des expirations, une fois le point ouvert levé)
— **MO-NET-008** — Architecture du Wi-Fi pédagogique (la même CA « BTS SIO Root CA » sert les certificats NPS/802.1X)
— **MO-AD-005** — Vérifier la santé de l'Active Directory avant intervention (réflexe avant toute modification ADCS)
— Spécification du chantier (dépôt) : `docs/superpowers/specs/2026-09-14-https-services-docker-design.md`

A Batterie de vérification `check_https.sh`

Batterie d'acceptation TLS : chaque URL doit répondre **sans `-k`**, validée par la racine ADCS. Sur le poste d'administration Kubuntu, la racine est installée dans le magasin système ; adapter `CA` sinon.

```
#!/usr/bin/env bash
set -uo pipefail
CA=/usr/local/share/ca-certificates/BTS-SIO-Root-CA.crt
fail=0
check() { # <url> <codes attendus, regex>
  local code
  code="$(curl -s -o /dev/null -w '%{http_code}' --cacert "${CA}" \
      --connect-timeout 6 --max-time 12 "$1")"
  if [[ "${code}" =~ $2 ]]; then printf 'OK   %-45s %s\n' "$1" "${code}"
  else printf 'FAIL %-45s %s\n' "$1" "${code}"; fail=1; fi
}
check https://homer.docker.bts.sio '^(200)$'
check https://portainer.docker.bts.sio '^(200)$'
check https://kuma.docker.bts.sio '^(200|302)$'
check https://netbox.docker.bts.sio '^(200|302)$'
check https://grafana.docker.bts.sio '^(200|302)$'
check https://prometheus.docker.bts.sio '^(200|401)$'
check https://auth.docker.bts.sio '^(200|302)$'
check https://unifi.docker.bts.sio '^(200|302)$'
check https://pihole.docker.bts.sio/admin/ '^(200|302)$'
check https://cadvisor.docker.bts.sio '^(200|401)$'
check https://traefik.docker.bts.sio '^(200|302)$'
check https://10.0.112.20 '^(200)$'
exit "${fail}"
```

B Script de préparation ADCS `adcs-template.ps1`

Versionné dans `files/adcs-template.ps1` ; à exécuter sur DC1 (§ 4.4).

```
# ============================================================================
# adcs-template.ps1 -- Preparation d'ADCS pour les certificats des services web
# ----------------------------------------------------------------------------
# Chantier "HTTPS des services" du 14/09/2026 (MO-SEC-005 / MO-SEC-006).
# A executer sur DC1 (SRV2022.bts.sio) dans une session PowerShell elevee,
# avec le compte BTS\Administrateur. Toutes les etapes sont idempotentes :
# le script peut etre relance sans effet indesirable.
#
# Actions :
#   1. Active EDITF_ATTRIBUTESUBJECTALTNAME2 (0x40000) sur la CA
#      (EditFlags : 0x11014e -> 0x15014e ; SAN acceptes depuis le CSR).
#   2. Cree le gabarit "BTSSIOWebServer1an" (duplique de "Web Server" :
#      validite 1 an, cle min 2048, cle exportable, EKU ServerAuth,
#      ENROLLEE_SUPPLIES_SUBJECT) s'il n'existe pas deja.
#   3. Ajoute l'ACE "Enroll" pour BTS\Administrateur sur ce gabarit
#      uniquement (soumissions certreq scriptees, action reversible).
#   4. Publie le gabarit sur la CA.
#   5. Fixe RequestExtensionList = 2.5.29.37 : l'EKU serverAuth est fournie
#      par le CSR (exigence OPNsense, cf. MO-SEC-006).
#   6. Redemarre le service certsvc et affiche l'etat final a controler.
#
# Commentaires volontairement en ASCII : powershell.exe 5.1 lit mal les
# fichiers UTF-8 sans BOM. Ce script ne contient aucun secret.
# ============================================================================

#Requires -RunAsAdministrator
[CmdletBinding()]
param()

$ErrorActionPreference = 'Stop'
Import-Module ActiveDirectory

$CaConfig     = 'SRV2022.bts.sio\BTS SIO Root CA'   # config string exacte
```

```
    (espaces comprises)
$TemplateName = 'BTSSIOWebServer1an'                # nom machine du gabarit
$DisplayName  = 'BTS SIO Web Server 1 an'           # displayName
$EnrollGuid   = [Guid]'0e10c968-78f3-11d2-90cc-00c04fd91ab1'  # extended-right
    Certificate-Enrollment
$EnrollSid    = (New-Object
    System.Security.Principal.NTAccount('BTS\Administrateur')).Translate(
                    [System.Security.Principal.SecurityIdentifier])

$configNC = (Get-ADRootDSE).configurationNamingContext
$ctPath   = "CN=Certificate Templates,CN=Public Key
    Services,CN=Services,$configNC"
$tplDn    = "CN=$TemplateName,$ctPath"

# --- 1. Flag SAN sur la CA ------------------------------------------------
Write-Host '== 1. Activation de EDITF_ATTRIBUTESUBJECTALTNAME2 (0x40000)'
& certutil -config $CaConfig -setreg policy\EditFlags +262144 | Out-Null

# --- 2. Gabarit duplique de "Web Server", validite 1 an -------------------
if (Get-ADObject -Filter { objectClass -eq 'pKICertificateTemplate' -and Name
    -eq $TemplateName } `
        -SearchBase $ctPath) {
    Write-Host "== 2. Gabarit $TemplateName deja present : creation ignoree"
} else {
    Write-Host "== 2. Creation du gabarit $TemplateName"
    $src = Get-ADObject -Identity "CN=WebServer,$ctPath" -Properties *
    # Si "WebServer" est introuvable, lister les gabarits et adapter :
    #   Get-ADObject -SearchBase $ctPath -Filter { objectClass -eq
    'pKICertificateTemplate' } |
    #       Select-Object Name, DisplayName
    $attrs = @{
        objectClass                     = 'pKICertificateTemplate'
        displayName                     = $DisplayName
        flags                           = $src.flags
        pKIDefaultKeySpec               = $src.pKIDefaultKeySpec
        pKIDefaultCSPs                  = $src.pKIDefaultCSPs
```

```
        pKICriticalExtensions           = $src.pKICriticalExtensions
        pKIMaxIssuingDepth              = $src.pKIMaxIssuingDepth
        pKIExtendedKeyUsage             = $src.pKIExtendedKeyUsage
    # Server Authentication
        pKIExpirationPeriod             =
    [BitConverter]::GetBytes([long]-315576000000000)  # 1 an
        pKIOverlapPeriod                =
    [BitConverter]::GetBytes([long]-36288000000000)   # 6 semaines
        'msPKI-Certificate-Name-Flag'   = $src.'msPKI-Certificate-Name-Flag'
    # ENROLLEE_SUPPLIES_SUBJECT
        'msPKI-Enrollment-Flag'         = $src.'msPKI-Enrollment-Flag'
        'msPKI-Private-Key-Flag'        = ($src.'msPKI-Private-Key-Flag' -bor
    0x10)         # cle exportable
        'msPKI-RA-Signature'            = 0
        'msPKI-Minimal-Key-Size'        = 2048
        'msPKI-Template-Schema-Version' = 2
        'msPKI-Template-Minor-Revision' = 2
        revision                        = 100
    }
    New-ADObject -Name $TemplateName -Path $ctPath -Type
    pKICertificateTemplate -OtherAttributes $attrs
}

# --- 3. ACE Enroll pour BTS\Administrateur (sur ce gabarit uniquement) ------
$acl = Get-Acl -Path "AD:$tplDn"
$alreadyGranted = $acl.Access | Where-Object {
    $_.AccessControlType -eq 'Allow' -and
    ($_.ActiveDirectoryRights -band
    [System.DirectoryServices.ActiveDirectoryRights]::ExtendedRight) -and
    $_.ObjectType -eq $EnrollGuid -and

    ($_.IdentityReference.Translate([System.Security.Principal.SecurityIdentifier])
    -eq $EnrollSid)
}
if ($alreadyGranted) {
    Write-Host '== 3. ACE Enroll pour BTS\Administrateur deja presente :
```

```
    ignoree'
} else {
    Write-Host '== 3. Ajout de l''ACE Enroll pour BTS\Administrateur'
    $ace = New-Object System.DirectoryServices.ActiveDirectoryAccessRule(
        $EnrollSid,
        [System.DirectoryServices.ActiveDirectoryRights]::ExtendedRight,
        [System.Security.AccessControl.AccessControlType]::Allow,
        $EnrollGuid)
    $acl.AddAccessRule($ace)
    Set-Acl -Path "AD:$tplDn" -AclObject $acl
}

# --- 4. Publication du gabarit sur la CA -----------------------------------
Write-Host '== 4. Publication du gabarit sur la CA'
& certutil -config $CaConfig -SetCATemplates "+$TemplateName" | Out-Null

# --- 5. EKU serverAuth fournie par le CSR ----------------------------------
Write-Host '== 5. RequestExtensionList = 2.5.29.37 (EKU prise au CSR)'
& certutil -config $CaConfig -setreg CA\RequestExtensionList '2.5.29.37' |
    Out-Null

# --- 6. Redemarrage et controles -------------------------------------------
Write-Host '== 6. Redemarrage de certsvc puis controles'
Restart-Service certsvc -Force

Write-Host ''
Write-Host '-- Controles attendus :'
Write-Host '   * EditFlags inclut 0x40000 (valeur constatee apres chantier :
    0x15014e)'
& certutil -config $CaConfig -getreg policy\EditFlags
Write-Host '   * RequestExtensionList = 2.5.29.37'
& certutil -config $CaConfig -getreg CA\RequestExtensionList
Write-Host "   * Le gabarit $TemplateName figure dans la liste publiee :"
& certutil -config $CaConfig -CATemplates
```

C Configurations OpenSSL versionnées

Fichiers du dossier `files/` (§ 5.1). Rappel : ajouter `extendedKeyUsage = serverAuth` dans `[ req_ext ]` de `openssl-wildcard.cnf` et `openssl-proxmox.cnf` lors de leur renouvellement.

`openssl-wildcard.cnf` (Traefik, Portainer)

```
[ req ]
default_bits       = 4096
prompt             = no
default_md         = sha256
distinguished_name = dn
req_extensions     = req_ext

[ dn ]
CN = docker.bts.sio
O  = BTS SIO - Lycee Jean Lurcat

[ req_ext ]
subjectAltName = @alt_names

[ alt_names ]
DNS.1 = docker.bts.sio
DNS.2 = *.docker.bts.sio
IP.1  = 10.0.112.20
```

`openssl-proxmox.cnf` (Proxmox)

```
[ req ]
default_bits       = 4096
prompt             = no
default_md         = sha256
distinguished_name = dn
req_extensions     = req_ext

[ dn ]
CN = 10.0.112.200
O  = BTS SIO - Lycee Jean Lurcat

[ req_ext ]
subjectAltName = @alt_names

[ alt_names ]
IP.1 = 10.0.112.200
```

`openssl-opnsense.cnf` (OPNsense)

```
[ req ]
default_bits       = 4096
prompt             = no
default_md         = sha256
distinguished_name = dn
req_extensions     = req_ext

[ dn ]
CN = 10.0.112.1
O  = BTS SIO - Lycee Jean Lurcat

[ req_ext ]
subjectAltName   = @alt_names
extendedKeyUsage = serverAuth

[ alt_names ]
IP.1 = 10.0.112.1
```

`openssl-vaultwarden.cnf` (Vaultwarden)

```
[ req ]
default_bits       = 4096
prompt             = no
default_md         = sha256
distinguished_name = dn
req_extensions     = req_ext

[ dn ]
CN = 10.0.112.10
O  = BTS SIO - Lycee Jean Lurcat

[ req_ext ]
subjectAltName   = @alt_names
extendedKeyUsage = serverAuth

[ alt_names ]
IP.1 = 10.0.112.10
```

`openssl-wazuh.cnf` (Wazuh)

```
[ req ]
default_bits       = 4096
prompt             = no
default_md         = sha256
distinguished_name = dn
req_extensions     = req_ext

[ dn ]
CN = wazuh.bts.sio
O  = BTS SIO - Lycee Jean Lurcat

[ req_ext ]
subjectAltName   = @alt_names
extendedKeyUsage = serverAuth

[ alt_names ]
DNS.1 = wazuh.bts.sio
IP.1  = 10.0.112.103
```

`openssl-unifi.cnf` (UniFi)

```
[ req ]
default_bits       = 4096
prompt             = no
default_md         = sha256
distinguished_name = dn
req_extensions     = req_ext

[ dn ]
CN = 10.0.112.228
O  = BTS SIO - Lycee Jean Lurcat

[ req_ext ]
subjectAltName   = @alt_names
extendedKeyUsage = serverAuth

[ alt_names ]
DNS.1 = unifi.docker.bts.sio
IP.1  = 10.0.112.228
```