Mode Opératoire

Comptes nominatifs ProxMox, rétrogradation du pool *icad* et 2FA TOTP

Code :	MO-PLT-021
Version :	1.0
Date :	16 avril 2026
Auteur :	Cédric LEGRAND
Classification :	USAGE INTERNE — Équipe BTS SIO

Historique des révisions

Version	Date	Modifications
1.0	16/04/2026	Création initiale — T14 (nominatifs), T16 (pool/icad), 2FA TOTP

1 Objet

Ce mode opératoire décrit la mise en conformité des comptes d'administration ProxMox VE de l'infrastructure BTS SIO avec les recommandations ANSSI PA-022 v3.0 sur l'administration sécurisée des systèmes d'information. Il couvre trois chantiers complémentaires :

1. **Création de comptes nominatifs** (T14) pour les administrateurs permanents (`clegrand@pve`, `manu@pve`), en remplacement de l'usage partagé de `root@pam` ;
2. **Rétrogradation du périmètre `/pool/icad`** (T16) de `Administrator` à `PVEAdmin` et harmonisation sur le groupe (modèle aligné sur les pools *ksav* et *publicom*) pour les trois comptes étudiants rattachés ;
3. **Activation d'un second facteur d'authentification (TOTP)** via l'interface en ligne de commande `pvesh`, compatible avec toute application conforme à la RFC 6238 (Bitwarden, Aegis, FreeOTP, Google Authenticator, Microsoft Authenticator, Proton Authenticator).

Ces actions répondent au constat **F-VIRT-019** de l'audit authentifié du 15 mars 2026 (comptes étudiants avec droits excessifs sur le pool *icad*) ainsi qu'à la recommandation R-12 (« déployer une authentification forte sur les interfaces d'administration »).

2 Champ d'application

Application	ProxMox VE 8.4.14 (kernel 6.8.12-17-pve)
Hébergement	Serveur physique (`10.0.112.200`), 2× Xeon E5-2620 v2, 32 Go RAM
Accès	`https://10.0.112.200:8006` ou SSH `root@10.0.112.200`
Comptes concernés	2 nouveaux (`clegrand@pve`, `manu@pve`) + 3 existants (`dbalmigere@pve`, `lbonet@pve`, `mboyer@pve`)
Durée estimée	20–30 minutes (15 min pour les ACL, 5 min par TOTP utilisateur)
Référentiel	ANSSI PA-022 v3, ANSSI « Les Essentiels » Virtualisation (2024), Proxmox VE *User Management*

3 Prérequis

Prérequis

— Accès SSH `root` sur le nœud ProxMox : `ssh root@10.0.112.200` (clé ed25519 déployée)
— Identifiants `root@pam` sauvegardés dans Vaultwarden (collection *Virtualisation*)
— Deux mots de passe de 24 caractères aléatoires prégénérés pour `clegrand@pve` et `manu@pve` (générateur interne Vaultwarden)
— Binaire `qrencode` installé localement pour générer les QR codes TOTP en mode texte (`apt install qrencode` si absent)
— Application TOTP installée sur le téléphone de l'utilisateur enrôlé (Bitwarden, Aegis, FreeOTP, Google/Microsoft/Proton Authenticator, etc.)
— Validation préalable du tuteur établissement (Manu MARTINEZ) pour la rétrogradation du pool *icad* (acquise le 30/03/2026 puis confirmée le 16/04/2026 par Paul R.)

Contexte historique du pool *icad*

Le groupe et le pool *icad* hébergeaient initialement un projet étudiant (CT 100 « icad », aujourd'hui arrêté). Trois comptes étudiants y gardent une habilitation `Administrator` *posée individuellement* sur le chemin `/pool/icad`, ce qui dévie du modèle appliqué aux pools *ksav* et *publicom* (ACL au niveau du groupe avec le rôle `PVEAdmin`). L'harmonisation retire sept privilèges sensibles (dont `Permissions.Modify` et `Sys.Modify`) et unifie la gouvernance des trois périmètres pédagogiques.

4 Procédure

4.1 Création des comptes nominatifs (T14)

Les comptes `clegrand@pve` et `manu@pve` sont créés dans le realm natif *pve* (authentification interne ProxMox). Le compte partagé `root@pam` est conservé pour les opérations de bris de glace uniquement.

Étape 1 — Se connecter à l'hôte ProxMox

```
ssh root@10.0.112.200
```

La clé ed25519 de l'administrateur principal est déjà déployée dans `/root/.ssh/authorized_keys` depuis le 26/03/2026.

Étape 2 — Créer les deux comptes nominatifs et attribuer le rôle `PVEAdmin`

```
# Cr\'{e}ation des comptes nominatifs (realm pve)
pveum user add clegrand@pve --comment "Cedric LEGRAND - admin nominatif"
pveum user add manu@pve     --comment "Emmanuel MARTINEZ -
    co-administrateur"

# D\'{e}finition des mots de passe (interactif)
pveum passwd clegrand@pve
pveum passwd manu@pve

# Attribution du r\^{o}le PVEAdmin sur la racine (propagation
    h\'{e}rit\'{e}e)
pveum acl modify / --users clegrand@pve --roles PVEAdmin --propagate 1
pveum acl modify / --users manu@pve     --roles PVEAdmin --propagate 1
```

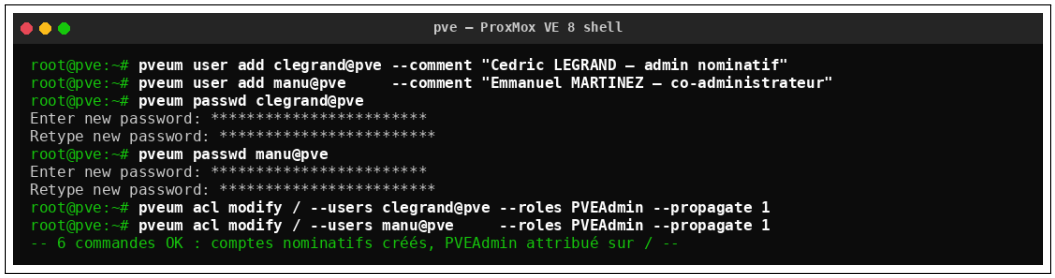

Figure 1 – Création des deux comptes nominatifs et attribution du rôle PVEAdmin sur `/`.

Étape 3 — Vérifier la liste des comptes actifs

```
pveum user list
```

Figure 2 – Liste des comptes ProxMox après création des deux comptes nominatifs : 11 comptes *pve* (2 administrateurs, 9 étudiants en trois groupes) et 1 compte *pam* réservé au bris de glace.

Cloisonnement `PVEAdmin` vs `Administrator`

Le rôle `PVEAdmin` couvre tout l'exploitation courante (gestion des VM/CT, snapshots, sauvegardes, migration, console) mais exclut explicitement quatre privilèges « système » : `Sys.Modify`, `Permissions.Modify`, `Realm.Allocate` et `Mapping.Modify`. Ces droits restent cantonnés au compte `root@pam`, conformément au principe ANSSI PA-022 de séparation par domaine technique.

4.2 Rétrogradation et harmonisation du pool *icad* (T16)

Avant l'intervention, les trois étudiants du groupe *icad* possèdent chacun une ACL `Administrator` posée *individuellement* sur `/pool/icad`, contrairement aux pools *ksav* (`/vms/101`) et *publicom* (`/vms/102`) qui utilisent une ACL posée au niveau du **groupe** avec le rôle `PVEAdmin`.

Étape 4 — État initial de la matrice d'habilitations

```
pveum acl list
```

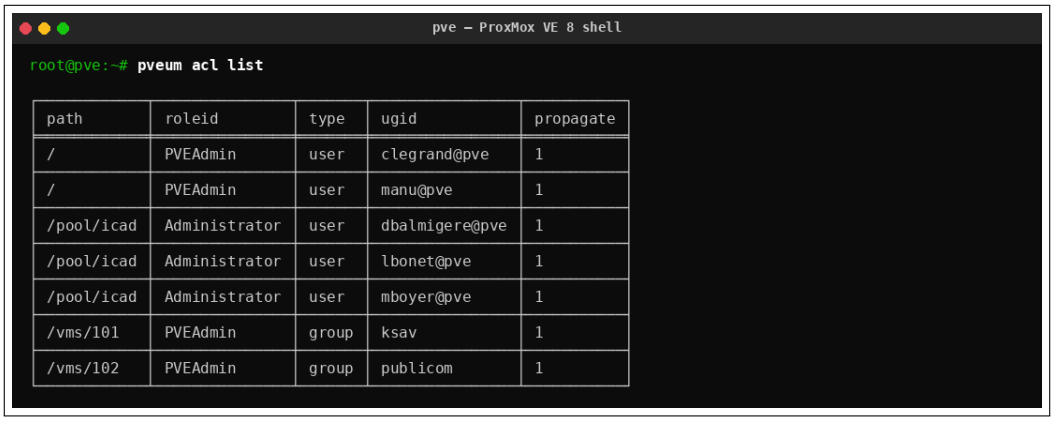

Figure 3 – État initial : trois ACL utilisateurs `Administrator` sur `/pool/icad` (ligne 3 à 5), incohérentes avec le modèle groupé utilisé pour *ksav* et *publicom*.

Étape 5 — Privilèges effectifs d'un étudiant du groupe *icad*

```
pveum user permissions dbalmigere@pve --path /pool/icad
```

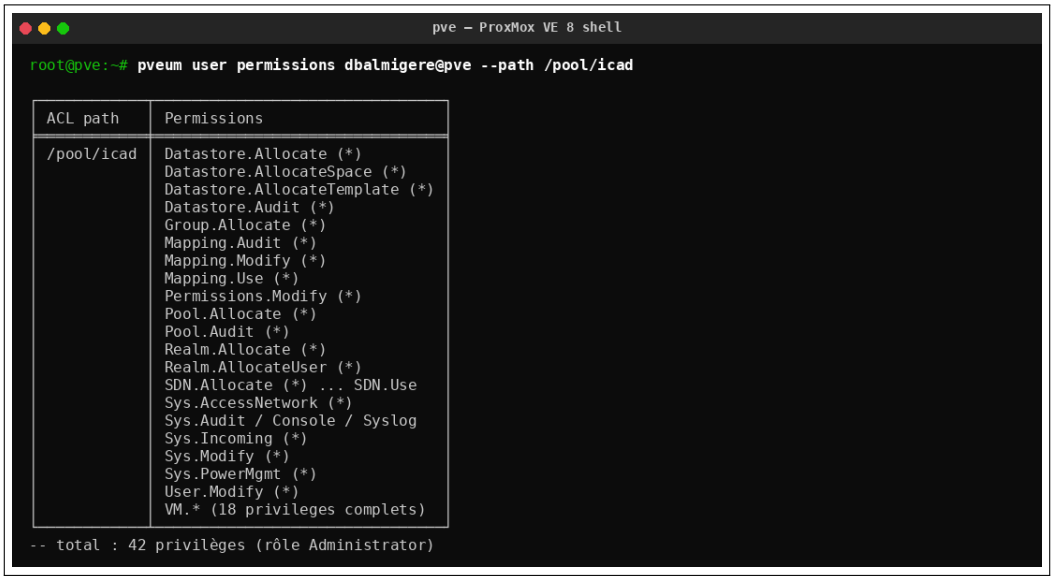

Figure 4 – Avant T16 : **42 privilèges** effectifs pour `dbalmigere@pve`, dont sept droits sensibles `Permissions.Modify`, `Sys.Modify`, `Realm.Allocate`, `Mapping.Modify`, `Sys.AccessNetwork`, `Sys.Incoming`, `Sys.PowerMgmt`.

⚠ Ordre des commandes critique

Toujours ajouter l'ACL du groupe avant de retirer les ACL utilisateurs individuelles. Un retrait d'ACL prématuré laisserait temporairement les étudiants sans aucun accès au pool, ce qui peut interrompre un TP en cours.

Étape 6 — Attribuer le rôle `PVEAdmin` au groupe *icad* sur le pool

```
# 1. Ajout de l'ACL au niveau du groupe (nouvelle habilitation)
pveum acl modify /pool/icad --groups icad --roles PVEAdmin --propagate 1
```

Étape 7 — Retirer les ACL utilisateurs `Administrator` devenues redondantes

```
# 2-4. Suppression des trois ACL utilisateurs Administrator
pveum acl delete /pool/icad --users dbalmigere@pve --roles Administrator
pveum acl delete /pool/icad --users lbonet@pve     --roles Administrator
pveum acl delete /pool/icad --users mboyer@pve     --roles Administrator
```

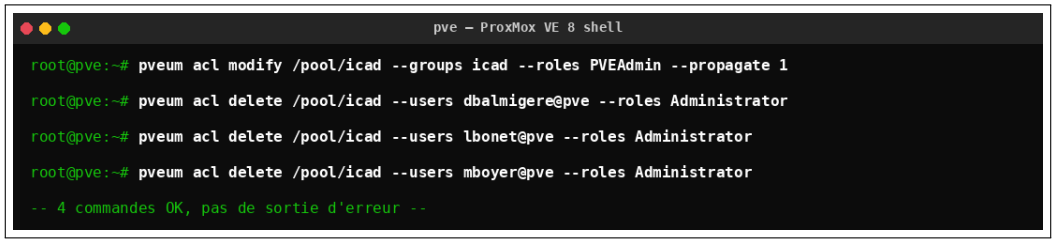

Figure 5 – Séquence complète T16 : une commande d'ajout au niveau du groupe, puis trois commandes de retrait des ACL utilisateurs devenues redondantes.

Étape 8 — Vérifier l'état final de la matrice d'habilitations

```
pveum acl list
```

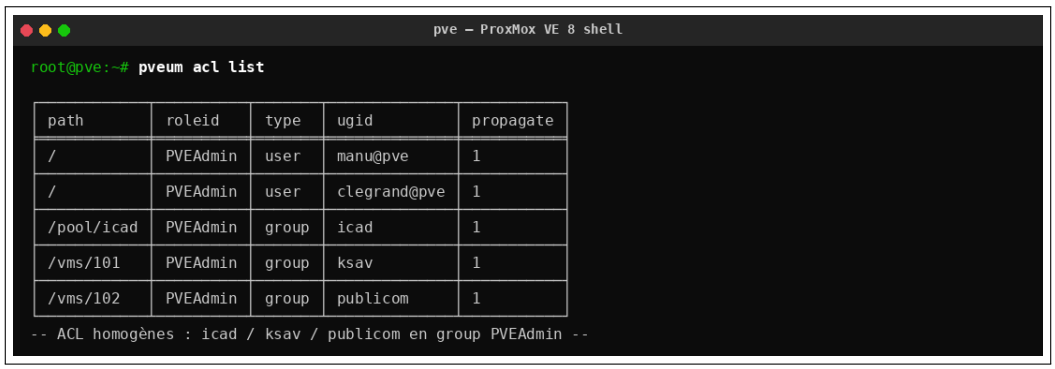

Figure 6 – Après T16 : la matrice est homogène — les trois pools étudiants (*icad*, *ksav*, *publicom*) utilisent désormais une ACL `PVEAdmin` posée au niveau du groupe.

Étape 9 — Constater la réduction des privilèges effectifs

```
pveum user permissions dbalmigere@pve --path /pool/icad
```

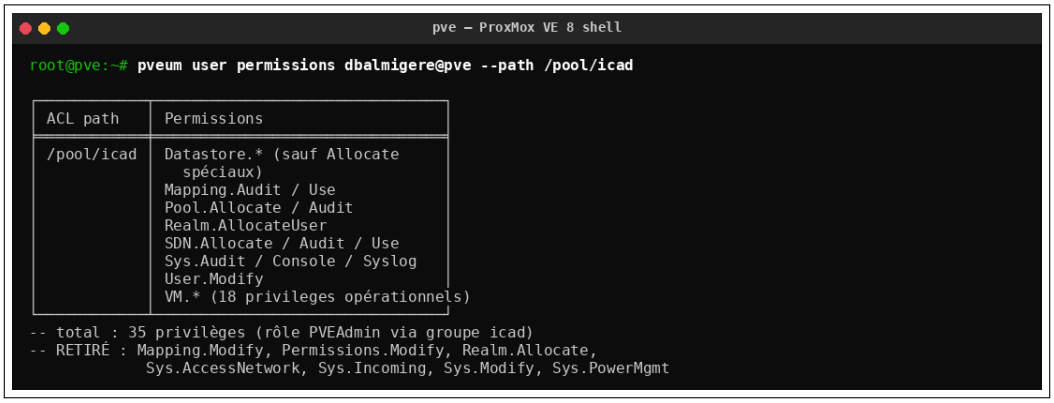

Figure 7 – Après T16 : **35 privilèges** effectifs (-7 par rapport au rôle Administrator), suffisants pour toutes les opérations d'exploitation dans le pool sans permettre l'escalade de privilèges.

4.3 Activation du 2FA TOTP via `pvesh`

L'activation du TOTP est une opération **individuelle** que chaque administrateur effectue pour son propre compte. La procédure décrite ci-dessous est agnostique à l'application cliente choisie (toute application conforme à la RFC 6238 est compatible).

Étape 10 — Générer un secret TOTP aléatoire et l'URI `otpauth`

```
# Secret de 30 octets al\'{e}atoires encod\'{e}s en base32 (format RFC
    4648)
SECRET=$(openssl rand -base32 30 | tr -d '=' | tr '+/' 'AB')

# Construction de l'URI otpauth selon la RFC 6238 (SHA1, 6 digits,
    p\'{e}riode 30 s)
URI="otpauth://totp/PVE-BTS:clegrand@pve?secret=${SECRET}\
&issuer=PVE-BTS&algorithm=SHA1&digits=6&period=30"
```

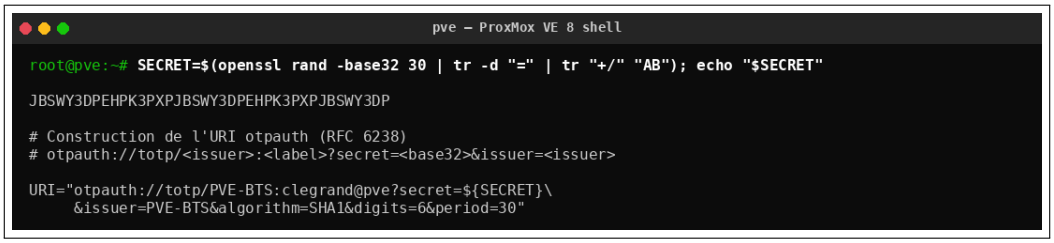

Figure 8 – Génération du secret partagé TOTP et construction de l'URI `otpauth`. L'URI encode à la fois le secret, l'algorithme de hachage, la longueur et la période du code.

Étape 11 — Afficher le QR code en mode terminal

```
qrencode -t UTF8 "$URI"
```

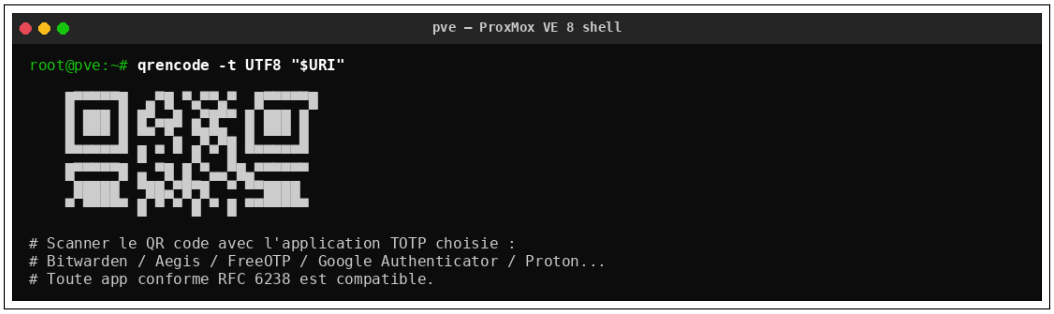

Figure 9 – Le QR code généré dans le terminal est scanné avec l'application TOTP choisie par l'utilisateur. Le secret ne transite jamais par le réseau ni par un tiers : il reste sur le terminal de l'administrateur et dans l'application.

Archiver le secret dans le gestionnaire de mots de passe

Il est recommandé de stocker également le secret TOTP dans Vaultwarden (champ *TOTP* natif de l'élément `clegrand@pve`) pour permettre la reconstitution du second facteur en cas de perte du téléphone, sans devoir ré-enrôler. Cette duplication reste conforme à l'ANSSI tant que le coffre est protégé par une phrase-passe forte et un 2FA distinct.

Étape 12 — Enregistrer le second facteur côté ProxMox

L'administrateur doit fournir **un code OTP valide** (lu dans l'application) pour prouver qu'il a bien enrôlé le secret côté client :

```
pvesh create /access/tfa/clegrand@pve \
    --type totp \
    --totp "$URI" \
    --value <CODE_OTP_6_CHIFFRES> \
    --description "TOTP principal clegrand - 16/04/2026" \
    --password "$MDP_CLEGRAND"
```

Figure 10 – Enregistrement du facteur TOTP : l'URI complet (avec secret) et le code OTP de vérification sont transmis en une seule requête à l'API. Le mot de passe du compte est requis pour prévenir toute altération par un accès concurrent.

Étape 13 — Vérifier l'enrôlement

```
pveum user tfa list clegrand@pve
```

Figure 11 – Liste des facteurs TFA enregistrés pour `clegrand@pve`. La commande retourne l'identifiant du facteur, son type et sa description — jamais le secret lui-même.

4.4 Exemple d'une connexion après activation du 2FA

Après enrôlement, toute connexion (interface web ou API) de `clegrand@pve` déclenche un échange en **deux étapes** : l'authentification primaire renvoie un *ticket partiel* marqué `NeedTFA=1`, qui ne donne aucun accès aux ressources tant que le second facteur n'est pas validé. Il faut alors réinjecter ce ticket partiel accompagné du code TOTP courant pour obtenir un ticket d'accès complet.

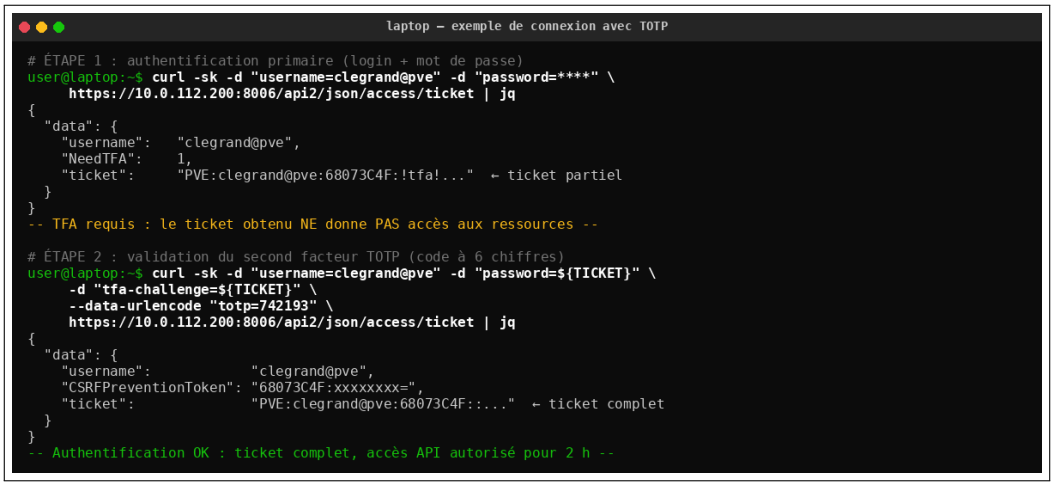

Figure 12 – Déroulé d'une connexion avec TOTP côté API : phase 1 renvoie un ticket partiel marqué `NeedTFA=1`, phase 2 présente le code OTP et reçoit un ticket complet valide 2 heures.

Interface web équivalente

Sur `https://10.0.112.200:8006`, le même échange est réalisé de façon transparente : un premier écran demande identifiant et mot de passe, puis un second écran « Second factor required » demande le code à 6 chiffres généré par l'application. L'option « Remember me » ne concerne que le premier facteur — le code TOTP est demandé à chaque nouvelle session.

5 Vérifications finales

Vérification

Contrôle	Commande / attendu
Comptes nominatifs présents	`pveum user list | grep -E 'clegrand|manu'` — 2 lignes, `enable=1`
Rôle `PVEAdmin` attribué	`pveum acl list | grep '/.*PVEAdmin'` — 2 lignes
Pool *icad* harmonisé	`pveum acl list | grep /pool/icad` — 1 seule ligne `group icad PVEAdmin`
Aucune ACL utilisateur résiduelle	`pveum acl list | grep -c Administrator` — 0
Privilèges étudiant réduits	`pveum user permissions dbalmigere@pve` — absence de `Permissions.Modify` et `Sys.Modify`
TOTP enrôlé pour un administrateur	`pveum user tfa list clegrand@pve` — 1 entrée de type `totp`
Connexion interactive OK	Login web + code OTP accepté en moins de 30 s

6 Rollback

Chaque chantier est réversible indépendamment. En cas d'anomalie constatée pendant les vérifications, appliquer uniquement le rollback du bloc concerné.

⚠ Accès console nécessaire en cas de perte TOTP

Si le TOTP est activé sur le compte de bris de glace `root@pam` **et** que le téléphone associé est perdu, la seule remise en service passe par un accès console physique ou IPMI au nœud. Éditer `/etc/pve/priv/tfa.cfg` et retirer la section correspondante puis redémarrer `pveproxy`. **Garder systématiquement un jeu de codes de récupération** (`-type recovery`) hors ligne.

Rollback T14 — retrait des comptes nominatifs

```
pveum user delete clegrand@pve
pveum user delete manu@pve
```

Rollback T16 — restauration des ACL utilisateurs Administrator

```
# Restaurer les ACL utilisateurs Administrator
pveum acl modify /pool/icad --users dbalmigere@pve --roles Administrator
    --propagate 1
pveum acl modify /pool/icad --users lbonet@pve     --roles Administrator
    --propagate 1
pveum acl modify /pool/icad --users mboyer@pve     --roles Administrator
    --propagate 1

# Retirer l'ACL groupe PVEAdmin
pveum acl delete /pool/icad --groups icad --roles PVEAdmin
```

Rollback TOTP — révoquer un facteur enrôlé

```
# Lister les facteurs pour r\'{e}cup\'{e}rer l'identifiant
pveum user tfa list clegrand@pve

# Supprimer un facteur pr\'{e}cis
pveum user tfa delete clegrand@pve --id <IDENTIFIANT_TFA>
```

7 Voir aussi

— **MO-PLT-018** — *Administration des conteneurs LXC sur ProxMox VE* (s'applique après migration sur un compte nominatif)
— **MO-PLT-016** — *Sauvegardes ProxMox* (les API tokens à privilèges réduits supplantent les comptes interactifs pour l'automatisation)
— **MO-AD-008** — *Sauvegardes Active Directory* (même logique de moindre privilège appliquée à l'AD)
— **MO-PLT-009** — *Connexion à Vaultwarden* (stockage des mots de passe nominatifs et des secrets TOTP)
— Référence externe : ANSSI PA-022 v3.0 (2021), « Les Essentiels » Virtualisation (2024), Proxmox VE *User Management*.