Mode Opératoire

Mise à jour du firmware OPNsense via l'API REST

Code :	MO-NET-002
Version :	1.0
Date :	9 avril 2026
Auteur :	Cédric LEGRAND
Classification :	USAGE INTERNE — Équipe BTS SIO

Historique des révisions

Version	Date	Modifications
1.0	09/04/2026	Création initiale (procédure validée le 07/04/2026)

1 Objet

Ce mode opératoire décrit la procédure complète de mise à jour du firmware OPNsense via l'API REST. Il couvre la vérification des mises à jour disponibles, la sauvegarde de la configuration, l'exécution de la mise à jour, le redémarrage du pare-feu et les contrôles post-upgrade.

La procédure a été élaborée et validée le 7 avril 2026, lors du passage de la version 26.1.2 à la version 26.1.5 sur le pare-feu OPNsense de l'infrastructure BTS SIO.

L'accès SSH au pare-feu étant indisponible (mot de passe root inconnu), l'intégralité de la procédure repose sur l'API REST exposée par l'interface web d'OPNsense.

2 Champ d'application

Public concerné	Administrateurs de l'infrastructure BTS SIO
Système	Pare-feu OPNsense 26.1.x (`10.0.112.1`)
Protocole	HTTP (HTTPS non activé — tâche T40 reportée)
Authentification	Session PHP (pas de clés API configurées)
Durée estimée	20–30 minutes (dont 5–10 min de redémarrage)

3 Prérequis

Prérequis

- Accès réseau au pare-feu : `http://10.0.112.1` (câble RJ45 sur site ou tunnel VPN WireGuard)
- Compte administrateur OPNsense (identifiants dans le gestionnaire de mots de passe de l'équipe)
- `curl` ou `Python requests` pour les appels API
- Espace disque suffisant sur le pare-feu pour le téléchargement des paquets
- Créneau de maintenance prévu (interruption de service pendant le redémarrage)

Interruption de service

La mise à jour nécessite un redémarrage du pare-feu. Pendant cette phase (5 à 10 minutes), **tout le trafic réseau est interrompu** : accès Internet, DNS, VPN WireGuard, Suricata IDS. Planifier l'opération en dehors des heures de cours.

4 Procédure

4.1 S'authentifier sur l'API OPNsense

L'API OPNsense utilise un système de sessions PHP avec jeton CSRF. L'authentification se fait en deux temps : récupération du jeton, puis envoi des identifiants.

Étape 1 — Récupérer le jeton CSRF initial

Envoyer une requête GET sur la page d'accueil pour obtenir le cookie de session et le jeton CSRF :

```
# Recuperer le cookie de session et le CSRF token
curl -v -c cookies.txt http://10.0.112.1/ 2>&1 \
  | grep -i "X-CSRFToken"
```

Le jeton CSRF est retourné dans l'en-tête `X-CSRFToken` de la réponse. Le conserver pour l'étape suivante.

Étape 2 — S'authentifier avec les identifiants

Soumettre le formulaire de connexion avec le jeton CSRF récupéré :

```
curl -b cookies.txt -c cookies.txt \
  -X POST http://10.0.112.1/ \
  -d "usernamefld=admin" \
  -d "passwordfld=MOT_DE_PASSE" \
  -d "login=1" \
  -H "X-CSRFToken: JETON_CSRF"
```

Après authentification, recharger une page protégée pour récupérer un nouveau jeton CSRF valide pour la session :

```
curl -b cookies.txt -c cookies.txt \
  http://10.0.112.1/ui/core/firmware \
  -D - 2>/dev/null | grep "X-CSRFToken"
```

Alternative Python

Pour une automatisation plus robuste, utiliser la bibliothèque `requests` de Python avec un objet `Session` qui gère automatiquement les cookies et les redirections. Un script d'exemple est disponible dans le wiki de l'équipe.

4.2 Vérifier les mises à jour disponibles

Étape 3 — Interroger l'API firmware/status

Une fois authentifié, interroger l'endpoint de statut du firmware :

```
curl -b cookies.txt \
  http://10.0.112.1/api/core/firmware/status
```

La réponse JSON contient les informations clés :

```
{
  "status": "update",
  "product_version": "26.1.2",
  "product_latest": "26.1.5",
  "updates": 74,
  "download_size": "..."
}
```

Interprétation des champs principaux :

Champ	Signification
`status`	`"update"` = mises à jour disponibles, `"none"` = système à jour
`product_version`	Version actuellement installée
`product_latest`	Dernière version disponible
`updates`	Nombre de paquets à mettre à jour

Note

Si `status` vaut `"none"`, le système est déjà à jour. Aucune action supplémentaire n'est nécessaire. Il peut être nécessaire de forcer une vérification préalable via `POST /api/core/firmware/check` pour actualiser le cache des mises à jour.

4.3 Sauvegarder la configuration

⚠ Sauvegarde obligatoire

Ne jamais lancer une mise à jour sans avoir sauvegardé la configuration courante. En cas de problème, la restauration de ce fichier permet de revenir à l'état antérieur.

Étape 4 — Télécharger la configuration XML

La sauvegarde peut être réalisée via l'API ou l'interface web.

Via l'API :

```
curl -b cookies.txt \
  http://10.0.112.1/api/core/backup/download \
  -o opnsense_config_$(date +%Y-%m-%d_%H%M).xml
```

Via l'interface web :

Naviguer vers **System** → **Configuration** → **Backups**, puis cliquer sur Download configuration.

Étape 5 — Valider la sauvegarde

Vérifier que le fichier XML téléchargé est complet et valide :

```
# Verifier la taille (doit etre > 50 Ko)
ls -lh opnsense_config_*.xml

# Verifier la structure XML
head -5 opnsense_config_*.xml
# Doit commencer par : <?xml version="1.0"?>
```

Stocker le fichier dans un emplacement sécurisé (le dépôt de l'équipe contient un dossier `audit/infrastructure/configurations/` prévu à cet effet).

4.4 Exécuter la mise à jour

Étape 6 — Lancer la mise à jour du firmware

Déclencher le téléchargement et l'installation des paquets :

```
curl -b cookies.txt \
  -X POST http://10.0.112.1/api/core/firmware/update \
  -H "X-CSRFToken: JETON_CSRF"
```

La requête retourne immédiatement un accusé de réception. Le processus de mise à jour s'exécute en arrière-plan sur le pare-feu.

Étape 7 — Surveiller la progression

Interroger régulièrement l'endpoint de progression :

```
# Polling toutes les 30 secondes
while true; do
  STATUS=$(curl -s -b cookies.txt \
    http://10.0.112.1/api/core/firmware/upgradestatus)
  echo "$(date +%H:%M:%S) - $STATUS"
  sleep 30
done
```

La réponse indique l'état courant : téléchargement des paquets, installation, préparation du redémarrage. Lorsque le processus est terminé, le champ `status` passe à `"done"` et un redémarrage est généralement requis.

⚠ Attention

Ne pas interrompre le processus de mise à jour une fois lancé. Une interruption pendant l'installation des paquets peut laisser le système dans un état incohérent.

4.5 Redémarrer le pare-feu

Étape 8 — Déclencher le redémarrage

Une fois la mise à jour terminée, lancer le redémarrage via l'API :

```
curl -b cookies.txt \
  -X POST http://10.0.112.1/api/core/firmware/reboot \
  -H "X-CSRFToken: JETON_CSRF"
```

Le pare-feu se redémarre. La connexion API est perdue immédiatement.

Étape 9 — Attendre le retour du pare-feu

Surveiller le retour du pare-feu par ping :

```
# Attendre que le pare-feu reponde
echo "Attente du reboot..."
while ! ping -c 1 -W 2 10.0.112.1 > /dev/null 2>&1; do
  echo "$(date +%H:%M:%S) - en attente..."
  sleep 10
done
echo "Pare-feu accessible !"
```

Le redémarrage prend généralement entre 5 et 10 minutes. Compter davantage si le serveur héberge également un hyperviseur (chargement du BIOS, des services, etc.).

4.6 Vérifier la mise à jour

Étape 10 — Contrôler la version installée

Se réauthentifier (la session a été perdue lors du redémarrage), puis interroger le statut :

```
# Apres re-authentification
curl -b cookies.txt \
  http://10.0.112.1/api/core/firmware/status
```

Vérifier que :

— `product_version` correspond à la version attendue (ex. `"26.1.5"`)

— `status` vaut `"none"` (plus de mises à jour en attente)

Étape 11 — Contrôler les services critiques

Valider le bon fonctionnement des services après la mise à jour :

```
# Suricata IDS
curl -s -b cookies.txt \
  http://10.0.112.1/api/ids/service/status

# DNS (Unbound)
curl -s -b cookies.txt \
  http://10.0.112.1/api/unbound/service/status

# Sante systeme
curl -s -b cookies.txt \
  http://10.0.112.1/api/core/system/status
```

Tester également la résolution DNS et la connectivité depuis un poste du réseau :

```
# Depuis un poste BTS SIO
ping -c 3 8.8.8.8
nslookup google.fr 10.0.112.1
```

4.7 Intervention BIOS (si nécessaire)

⚠ Retour d'expérience — 7 avril 2026

Lors de la mise à jour du 7 avril, le serveur Dell hébergeant OPNsense s'est bloqué après le redémarrage sur l'écran **« F1 to continue, F2 to enter Setup »**. Ce comportement est lié à une alerte BIOS (disque ou configuration matérielle) qui requiert une validation manuelle au clavier.
Le contrôleur BMC/iDRAC du serveur n'étant pas configuré (adresse IP à `0.0.0.0`), un accès **console physique** a été nécessaire pour appuyer sur F1 et poursuivre le démarrage.

Si le pare-feu ne répond pas au ping après 15 minutes :

1. Se rendre physiquement devant le serveur (salle serveur du lycée)
2. Brancher un écran et un clavier sur le serveur Dell
3. Si l'écran affiche « F1 to continue », appuyer sur `F1`
4. Attendre la fin du démarrage d'OPNsense (1–2 minutes supplémentaires)
5. Vérifier le retour du ping depuis un poste du réseau

Prévenir le blocage BIOS

Pour éviter ce problème lors des prochaines mises à jour, configurer l'iDRAC du serveur Dell avec une adresse IP accessible. Cela permettra d'accéder à la console distante (KVM) sans déplacement physique. La configuration de l'iDRAC peut se faire via le BIOS : **F2** → **iDRAC Settings** → **Network**.

5 Vérification

Vérification

Après la mise à jour, vérifier systématiquement les points suivants :

- ☐ La version OPNsense correspond à la version cible (`product_version`)
- ☐ Le statut firmware indique `"none"` (aucune mise à jour en attente)
- ☐ Le service Suricata IDS est actif et génère des alertes
- ☐ Le DNS Unbound résout correctement les noms (`nslookup`)
- ☐ Le tunnel VPN WireGuard est établi et fonctionnel
- ☐ La connectivité Internet est opérationnelle (`ping 8.8.8.8`)
- ☐ La sauvegarde de configuration pré-mise à jour est archivée
- ☐ Les règles de pare-feu sont inchangées (spot check sur quelques règles)

6 Dépannage

Problème	Solution
L'API retourne une erreur 403 ou une page de login	La session a expiré ou le jeton CSRF est invalide. Se réauthentifier entièrement : récupérer un nouveau cookie et un nouveau jeton CSRF, puis relancer la requête.
`firmware/status` retourne `"error"`	Le serveur de mises à jour n'est pas joignable. Vérifier la connectivité Internet du pare-feu et la résolution DNS. Tenter `POST /api/core/firmware/check` pour forcer une nouvelle vérification.
Le pare-feu ne répond plus après le redémarrage	Attendre 15 minutes. Si toujours inaccessible, accéder à la console physique (cf. section 4.7). Le serveur peut être bloqué sur un écran BIOS « F1 to continue ».
VPN WireGuard non fonctionnel après la mise à jour	Vérifier l'état de l'interface WireGuard dans **VPN** → **WireGuard** → **General**. S'assurer que le service est démarré. Si le tunnel est en échec, redémarrer le service. La configuration est conservée après mise à jour, mais le service peut nécessiter un redémarrage manuel.
Suricata IDS inactif après mise à jour	Naviguer vers **Services** → **Intrusion Detection** → **Administration**. Vérifier que *Enabled* est coché et relancer le service via le bouton ▶. La mise à jour peut désactiver certains rulesets : contrôler l'onglet **Rules**.
DNS non fonctionnel	Vérifier le service Unbound via **Services** → **Unbound DNS** → **General**. S'assurer que le service est actif. En dernier recours, tester la résolution directe : `dig @10.0.112.1 google.fr`.