Mode Opératoire

Vérification post-hardening du pare-feu OPNsense

Code :	MO-NET-001
Version :	1.0
Date :	13 avril 2026
Auteur :	Cédric LEGRAND
Classification :	USAGE INTERNE — Équipe BTS SIO

Historique des révisions

Version	Date	Modifications
1.0	13/04/2026	Création initiale

1 Objet

Ce mode opératoire définit la procédure de vérification à appliquer après chaque intervention de sécurisation sur le pare-feu OPNsense de l'infrastructure BTS SIO. Il fournit une liste de contrôles couvrant les principaux axes de durcissement identifiés lors de l'audit de sécurité : règles de filtrage, détection d'intrusion, accès d'administration, résolveur DNS et proxy.

Chaque contrôle est accompagné de la commande API ou du chemin d'accès dans l'interface web, du résultat attendu et de la conduite à tenir en cas d'anomalie. La procédure peut être exécutée sur site (câble RJ45) comme à distance via le tunnel VPN WireGuard.

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 à activer — tâche T40)
Authentification	Session PHP avec jeton CSRF
Durée estimée	15–20 minutes

Constat d'audit	Tâche	Domaine de vérification
F-SEC-001	T33	Suppression de la règle Open_Bar
F-SEC-002	T33	Filtrage des interfaces optionnelles
F-SEC-003, F-SEC-004	—	Rulesets et interfaces Suricata IDS
F-SEC-005	T33	Capture du trafic par le proxy Squid
F-SEC-006, F-SEC-007	T32	SSH WAN désactivé, verrouillage actif
F-SEC-009	T34	Comptes nominatifs et privilèges
F-SEC-010, F-SEC-011	T45	DNSSEC, journalisation DNS
—	T40	HTTPS sur l'interface d'administration

3 Prérequis

Prérequis

— Accès réseau au pare-feu : `http://10.0.112.1` (RJ45 sur site ou VPN WireGuard)
— Identifiants administrateur OPNsense (gestionnaire de mots de passe de l'équipe)
— `Python 3` avec `requests` ou `curl` pour les appels API
— Connaissance des constats d'audit à vérifier (références F-SEC-xxx ci-dessus)

Authentification API

L'API OPNsense utilise une session PHP avec jeton CSRF. Se référer au mode opératoire **MO-NET-002** (section 4.1) pour la procédure d'authentification complète. Tous les appels API ci-dessous supposent une session authentifiée.

4 Procédure de vérification

4.1 Règles de filtrage — Open_Bar et interfaces

Étape 1 — Vérifier la suppression de la règle Open_Bar (F-SEC-001)

L'alias `Open_Bar` (valeur `10.0.0.0/16`) permettait à l'ensemble du réseau interne de contourner le proxy et les mécanismes de filtrage. Contrôler sa suppression :

Via l'interface web :

Naviguer vers **Firewall** → **Aliases**. Vérifier qu'aucun alias nommé `Open_Bar` n'apparaît dans la liste.

Naviguer vers **Firewall** → **Rules** → **LAN**. Vérifier qu'aucune règle n'utilise cet alias comme source ou destination.

Via l'API :

```
# Rechercher l'alias Open_Bar
curl -s -b cookies.txt \
  "http://10.0.112.1/api/firewall/alias/searchItem" \
  -X POST -H "X-CSRFToken: $CSRF" \
  -d '{"searchPhrase":"Open_Bar"}'
```

Résultat attendu : le champ `rows` de la réponse doit être vide (`rowCount: 0`). Si l'alias est encore présent, supprimer la règle associée, puis l'alias, et appliquer les modifications.

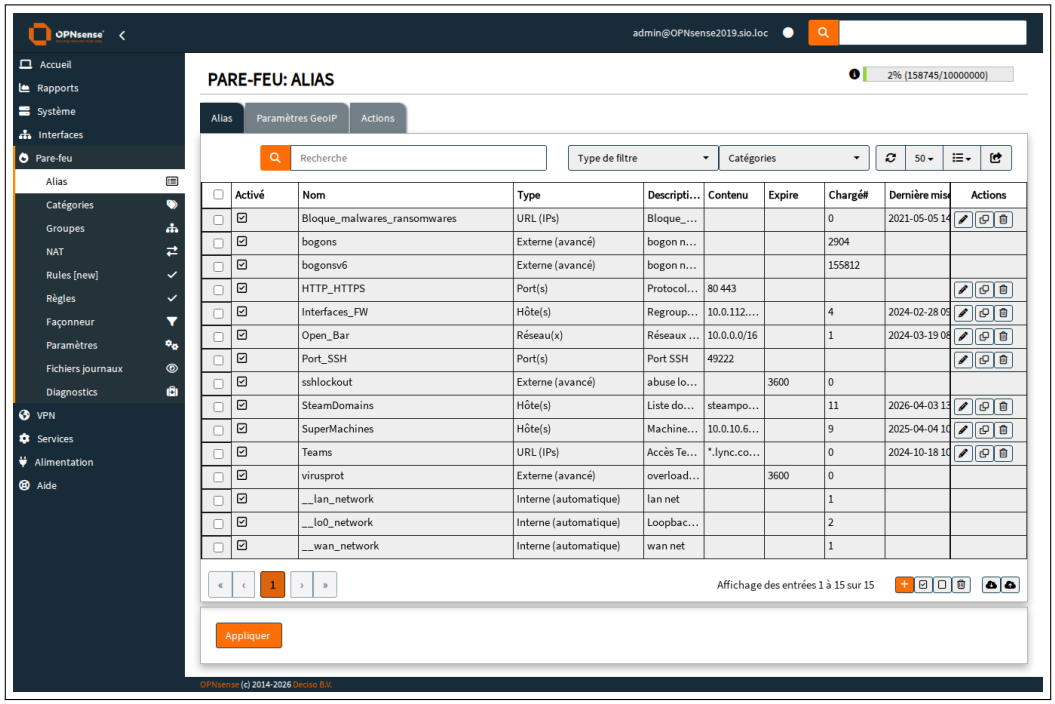

Figure 1 – Page **Pare-feu** → **Alias** : l'alias `Open_Bar` est encore visible (constat F-SEC-001 non corrigé).

Étape 2 — Vérifier le filtrage des interfaces optionnelles (F-SEC-002)

Les interfaces `opt1` et `opt2` avaient une règle PASS `any` → `any` sans aucun filtrage. Contrôler leur état :

Via l'interface web :

Naviguer vers **Interfaces** → **Overview**. Pour chaque interface optionnelle :

— Si l'interface n'est pas utilisée : elle doit être **désactivée**

— Si l'interface est active : naviguer vers **Firewall** → **Rules** → [**interface**] et vérifier qu'il n'existe aucune règle PASS `any` → `any`

Résultat attendu : les interfaces inutilisées sont désactivées ; les interfaces actives ont des règles restrictives (source et destination explicites).

4.2 Détection d'intrusion — Suricata IDS

Étape 3 — Vérifier les rulesets activés (F-SEC-003)

L'audit a révélé que zéro des 64 rulesets étaient activés malgré un service Suricata en fonctionnement. Contrôler l'activation :

```
# Lister les rulesets et compter les actifs
curl -s -b cookies.txt \
  "http://10.0.112.1/api/ids/settings/listRulesets" \
  | python3 -c "
import sys, json
data = json.load(sys.stdin)
enabled = [k for k,v in data.items()
           if isinstance(v, dict) and v.get('enabled')=='1']
print(f'Rulesets actifs : {len(enabled)}/64')
for r in sorted(enabled)[:10]:
    print(f'  - {r}')
if len(enabled) > 10:
    print(f'  ... et {len(enabled)-10} autres')
"
```

Résultat attendu : au minimum **32 rulesets** activés, incluant :

— Les rulesets ET Open (emerging-xxx)

— abuse.ch Feodo Tracker, SSL Fingerprint Blacklist

— ET Open Drop, Compromised, Botcc

Étape 4 — Vérifier les interfaces surveillées (F-SEC-004)

Initialement, seule l'interface WAN était surveillée. Contrôler que le LAN est également inclus :

Via l'interface web :

Naviguer vers **Services** → **Intrusion Detection** → **Administration**. Dans le champ *Interfaces*, vérifier que **WAN** et **LAN** sont sélectionnés.

Via l'API :

```
curl -s -b cookies.txt \
  "http://10.0.112.1/api/ids/settings/get" \
  | python3 -c "
import sys, json
data = json.load(sys.stdin)
ifaces = data.get('ids',{}).get('general',{}).get('interfaces','')
print(f'Interfaces surveillees : {ifaces}')
"
```

Résultat attendu : les interfaces WAN et LAN apparaissent toutes les deux dans la configuration.

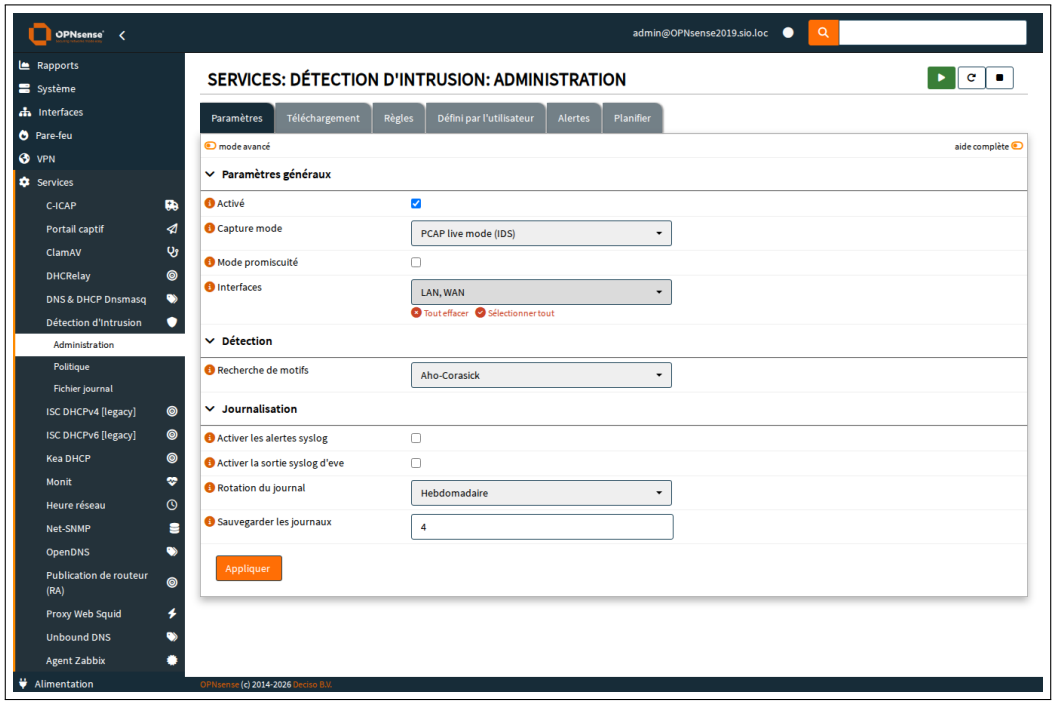

Figure 2 – Page **Services** → **Détection d'intrusion** → **Administration** : paramètres généraux et interfaces surveillées.

Étape 5 — Vérifier la génération d'alertes

Un service Suricata actif avec des rulesets chargés doit produire des alertes dans le journal `eve.json`. Contrôler :

```
# Interroger les dernieres alertes
curl -s -b cookies.txt \
  -X POST "http://10.0.112.1/api/ids/service/queryAlerts" \
  -H "X-CSRFToken: $CSRF" \
  -d '{"current":1,"rowCount":5}' \
  | python3 -c "
import sys, json
data = json.load(sys.stdin)
rows = data.get('rows', [])
print(f'Alertes recentes : {data.get(\"total\",0)} total')
for r in rows:
    ts = r.get('timestamp','')[:19]
    sig = r.get('alert', {}).get('signature', 'N/A')
    src = r.get('src_ip','?')
    print(f'  [{ts}] {sig} (src: {src})')
"
```

Résultat attendu : le nombre total d'alertes est supérieur à zéro. Des évènements récents (moins de 24 heures) doivent apparaître si le réseau est actif.

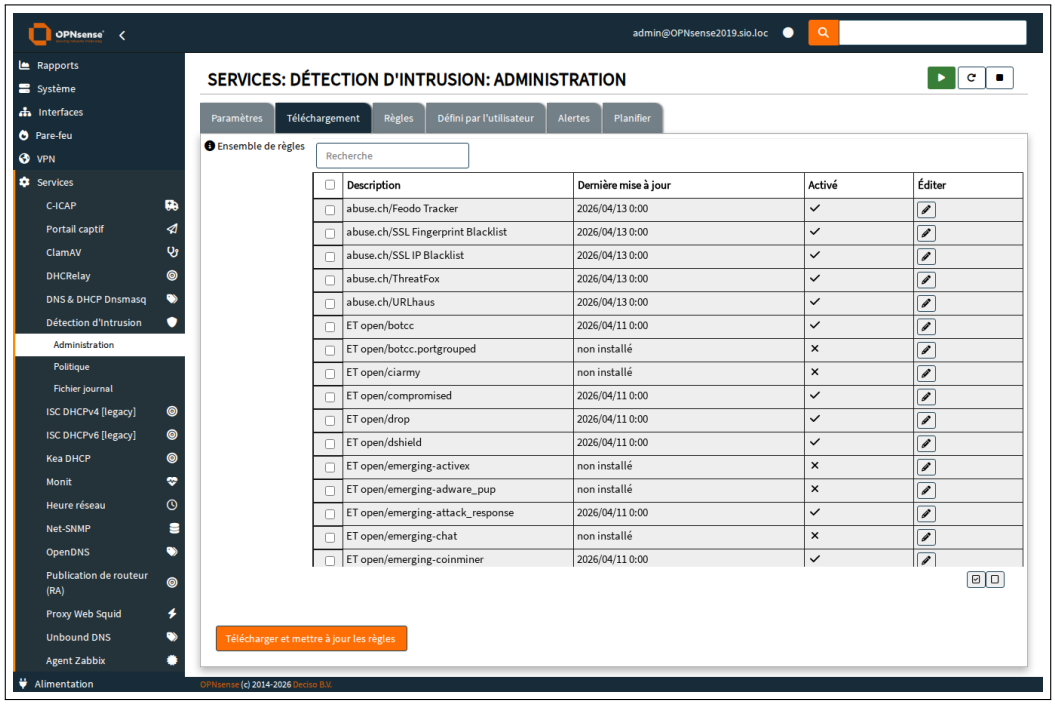

Figure 3 – Onglet **Téléchargement** : liste des rulesets avec leur état d'activation et leur date de dernière mise à jour.

Mode IDS versus IPS

La configuration actuelle fonctionne en mode **IDS** (détection seule, mode `pcap`). Le basculement en mode **IPS** (prévention, mode `netmap`) est documenté dans un mode opératoire séparé (MO-SEC-003) et ne doit être activé qu'en période contrôlée.

4.3 Accès d'administration — SSH et comptes

Étape 6 — Vérifier la désactivation du SSH sur WAN (F-SEC-006)

Le service SSH était exposé sur le port 49222 en WAN avec une règle PASS depuis toute source. Contrôler :

Via l'interface web :

1. Naviguer vers **System** → **Settings** → **Administration**
2. Vérifier que *Listen Interfaces* est restreint au **LAN uniquement** (pas WAN)
3. Naviguer vers **Firewall** → **Rules** → **WAN**
4. Vérifier qu'aucune règle PASS vers le port 49222 n'est présente

Résultat attendu : le service SSH n'écoute que sur le LAN et aucune règle WAN ne permet d'y accéder. L'accès distant passe exclusivement par le tunnel VPN WireGuard.

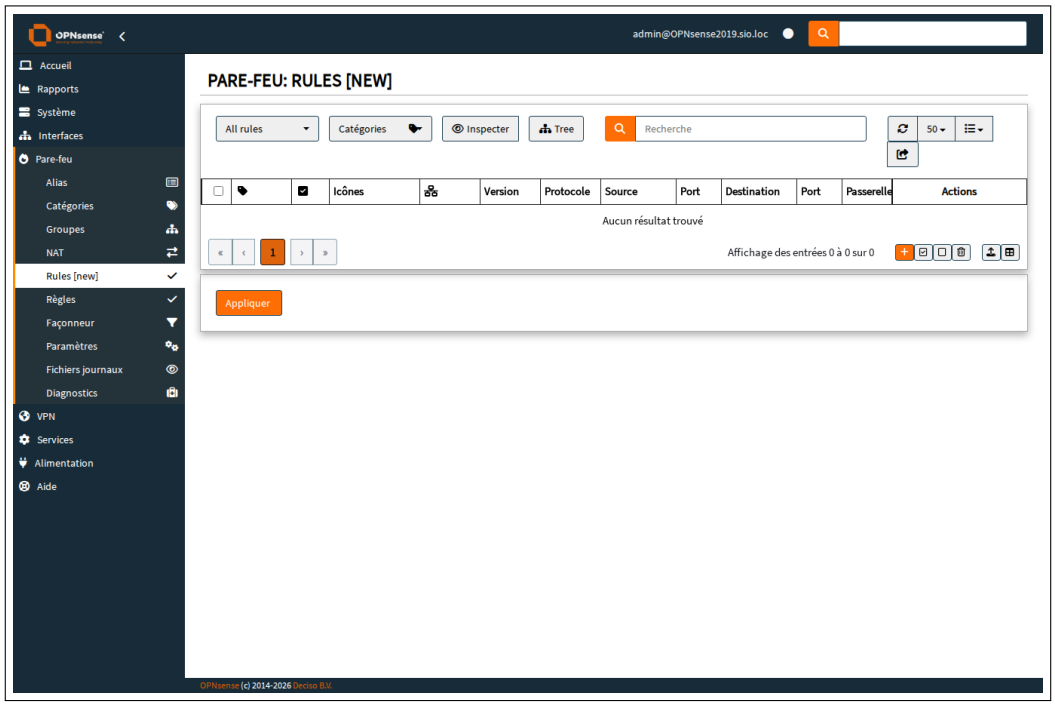

Figure 4 – Page **Pare-feu** → **Règles** → **WAN** : vérifier l'absence de règle PASS vers le port 49222.

Étape 7 — Vérifier le verrouillage anti brute-force SSH (F-SEC-007)

Le mécanisme `sshlockout` était configuré mais l'alias associé était vide (0 entrées). Contrôler :

Via l'interface web :

Naviguer vers **Firewall** → **Aliases**. Vérifier que l'alias `sshlockout` est de type **External (Advanced)** et que le seuil de verrouillage est configuré dans les paramètres SSH (**System** → **Settings** → **Administration** → *Login Protection*).

Résultat attendu : le seuil est défini (3 à 5 tentatives), la période de blocage est d'au moins 15 minutes.

Étape 8 — Vérifier les comptes et privilèges (F-SEC-009)

L'audit a identifié trois comptes (`root`, `admin`, `Manuel`) tous avec les privilèges `page-all`. Contrôler la séparation des rôles :

Via l'interface web :

Naviguer vers **System** → **Access** → **Users**. Vérifier que :

— Des **comptes nominatifs** existent pour chaque administrateur

— Le compte `root` est restreint à la **console physique uniquement**

— Le mot de passe partagé (`Colombo66`) a été remplacé par des mots de passe individuels stockés dans Vaultwarden

— Chaque compte dispose de privilèges adaptés à son rôle (pas de `page-all` systématique)

Résultat attendu : au minimum deux comptes nominatifs, mot de passe partagé abandonné, `root` non utilisable via l'interface web.

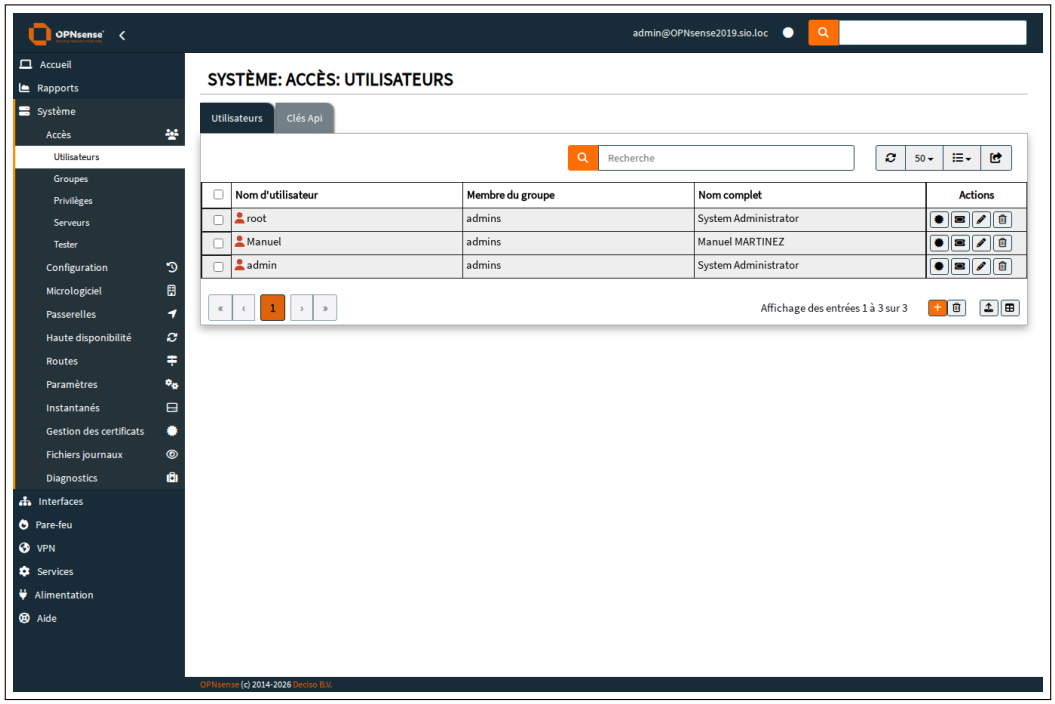

Figure 5 – Page **Système** → **Accès** → **Utilisateurs** : trois comptes (`admin`, `Manuel`, `root`), tous membres du groupe `admins` (constat F-SEC-009).

4.4 Résolveur DNS — Unbound

Étape 9 — Vérifier la configuration DNSSEC et la journalisation (F-SEC-010, F-SEC-011)

Le DNS Unbound avait la journalisation désactivée (`logqueries=0`, `logreplies=0`), pas de DNSSEC, et transférait les requêtes en clair vers la box Orange. Contrôler :

```
curl -s -b cookies.txt \
  "http://10.0.112.1/api/unbound/settings/get" \
  | python3 -c "
import sys, json
data = json.load(sys.stdin)
g = data.get('unbound',{}).get('general',{})
print(f'DNSSEC          : {g.get(\"dnssec\",\"?\")}'
      f'  (attendu: 1)')
print(f'Log queries     : {g.get(\"logqueries\",\"?\")}'
      f'  (attendu: 1)')
print(f'Log replies     : {g.get(\"logreplies\",\"?\")}'
      f'  (attendu: 1)')
print(f'Enabled         : {g.get(\"enabled\",\"?\")}'
      f'  (attendu: 1)')
# Verifier les forwarders
fwd = data.get('unbound',{}).get('dots',{})
print(f'DNS over TLS    : {len(fwd)} forwarder(s)')
"
```

Via l'interface web :

Naviguer vers **Services** → **Unbound DNS** → **General** :

— *DNSSEC* doit être **coché**

— *Log Queries* et *Log Replies* doivent être **cochés**

Naviguer vers l'onglet **DNS over TLS** :

— Au moins un résolveur DoT de confiance doit être configuré (ex. `1.1.1.1:853` Cloudflare, `9.9.9.9:853` Quad9)

— Le forwarder vers la box Orange (`192.168.1.1`) en clair doit être supprimé ou placé en dernière position

Résultat attendu : DNSSEC actif, journalisation complète, résolution chiffrée via DoT.

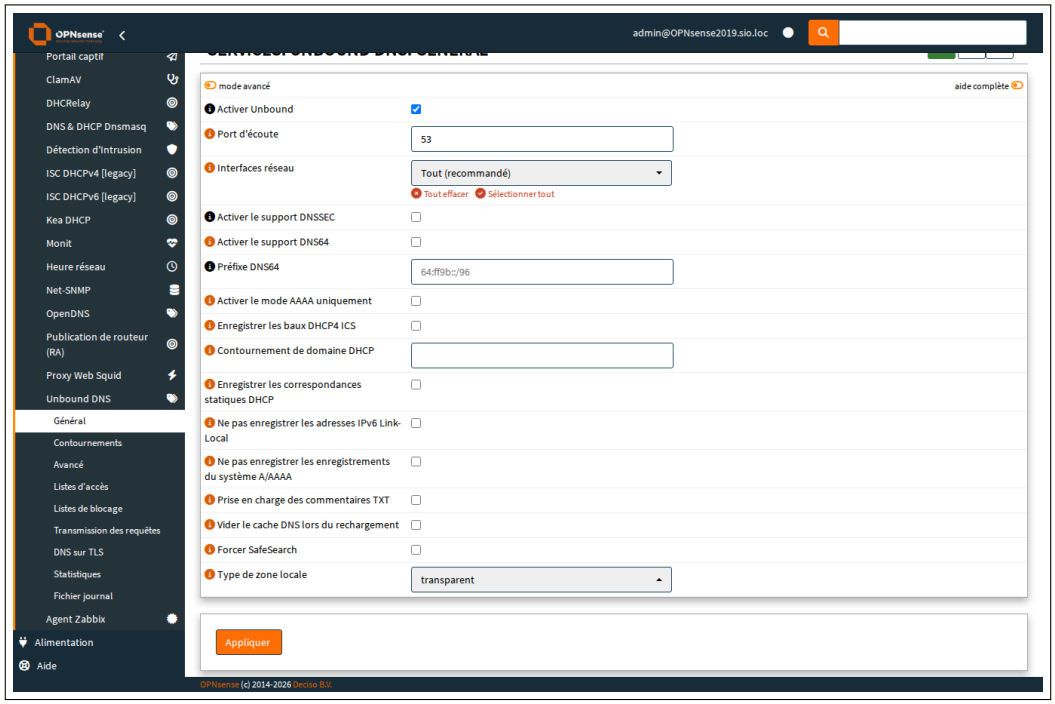

Figure 6 – Page **Services** → **Unbound DNS** → **General** : configuration du résolveur DNS (DNSSEC, journalisation, DoT).

Étape 10 — Tester la résolution DNS

Depuis un poste du réseau BTS SIO, vérifier le fonctionnement du résolveur :

```
# Resolution standard
nslookup google.fr 10.0.112.1

# Verification DNSSEC (doit retourner le flag AD)
dig +dnssec google.fr @10.0.112.1

# Resolution du domaine local
nslookup dc1.bts.sio 10.0.112.1
```

Résultat attendu : la résolution fonctionne, le flag `ad` (Authenticated Data) apparaît dans la réponse `dig` si DNSSEC est actif.

4.5 Proxy Squid — Capture du trafic

Étape 11 — Vérifier que le proxy reçoit du trafic (F-SEC-005)

Le proxy Squid était intégralement configuré (SSL Bump, liste Toulouse, ClamAV) mais ne recevait aucune requête en raison de la règle Open_Bar. Après suppression de cette règle, contrôler :

Via l'interface web :

Naviguer vers **Services** → **Web Proxy** → **Real Time Logs**. Des requêtes HTTP/HTTPS provenant de postes internes doivent apparaître.

Si aucune requête n'apparaît :

1. Vérifier les règles NAT de redirection : **Firewall** → **NAT** → **Port Forward**
2. Confirmer que les règles de redirection HTTP (80) et HTTPS (443) pointent bien vers le port du proxy (3128 par défaut)
3. Vérifier que la source des règles NAT ne référence plus l'alias supprimé

Résultat attendu : des requêtes apparaissent dans les logs temps réel du proxy lorsqu'un utilisateur navigue sur Internet depuis un poste interne.

4.6 HTTPS sur l'interface d'administration

Étape 12 — Vérifier l'activation du HTTPS (T40)

L'interface d'administration est actuellement en HTTP simple. Après activation du HTTPS :

Via l'interface web :

Naviguer vers **System** → **Settings** → **Administration** :

— *Protocol* doit être réglé sur **HTTPS**

— *SSL Certificate* doit référencer un certificat valide

— *HTTP Redirect* doit être **coché** (redirection automatique HTTP → HTTPS)

Test de connectivité :

```
# Le HTTP doit rediriger vers HTTPS
curl -v http://10.0.112.1/ 2>&1 | grep "Location:"
# Attendu : Location: https://10.0.112.1/

# Le HTTPS doit repondre
curl -sk https://10.0.112.1/api/core/system/status
```

Résultat attendu : l'accès en HTTP est redirigé vers HTTPS, l'API répond correctement en HTTPS.

⚠ Accès console requis pour T40

L'activation du HTTPS nécessite un accès à la **console physique** du serveur, car le mot de passe root SSH est inconnu. Si la bascule vers HTTPS provoque une perte d'accès à l'interface web, la console permettra de revenir en HTTP. Ne pas tenter cette opération à distance sans accès de secours.

5 Vérification — Grille récapitulative

Vérification

Après chaque campagne de hardening, cocher systématiquement les points suivants :

Filtrage réseau

☐ Alias Open_Bar supprimé (`Firewall` → `Aliases`)
☐ Règle associée supprimée des règles LAN
☐ Interfaces opt1/opt2 désactivées ou filtrées

Détection d'intrusion

☐ Suricata actif avec ≥ 32 rulesets chargés
☐ Interfaces WAN **et** LAN surveillées
☐ Alertes récentes présentes dans `eve.json`

Accès d'administration

☐ SSH WAN désactivé (port 49222 fermé)
☐ Verrouillage `sshlockout` configuré (seuil 3–5 tentatives)
☐ Comptes nominatifs créés, mot de passe partagé remplacé
☐ `root` restreint à la console physique
☐ HTTPS activé sur l'interface d'administration (si T40 réalisée)

DNS

☐ DNSSEC activé
☐ Journalisation des requêtes et réponses activée
☐ Résolution DNS over TLS configurée
☐ Résolution fonctionnelle depuis un poste interne

Proxy

☐ Le proxy Squid reçoit du trafic (logs temps réel non vides)
☐ Les règles NAT de redirection sont correctement configurées

6 Dépannage

Problème	Solution
L'API retourne 403 ou redirige vers la page de login	La session a expiré. Se réauthentifier complètement : nouveau cookie, nouveau jeton CSRF (cf. MO-NET-002, § 4.1).
Suricata affiche 0 rulesets malgré l'activation	Les règles n'ont pas été téléchargées. Exécuter `POST /api/ids/service/updateRules` puis `POST /api/ids/service/reconfigure`. Le téléchargement prend environ 2 minutes.
Aucune alerte Suricata malgré les rulesets actifs	Vérifier que l'interface LAN est bien sélectionnée dans la configuration IDS. Si le problème persiste, redémarrer le service : `POST /api/ids/service/restart`.
Le proxy ne reçoit aucune requête	Vérifier les règles NAT de port forward : elles doivent rediriger les ports 80 et 443 du LAN vers le port 3128 du pare-feu. Si les règles référencent un alias supprimé, les recréer manuellement.
Perte de connectivité Internet après suppression d'Open_Bar	La suppression de la règle de contournement redirige tout le trafic vers le proxy. Vérifier que le proxy est démarré et que les règles NAT sont en place. En urgence, créer une règle temporaire PASS sur LAN pour rétablir la connectivité.
DNS non fonctionnel après activation DNSSEC	Certains domaines sans enregistrements DNSSEC peuvent échouer en mode strict. Vérifier avec `dig +dnssec domaine.fr @10.0.112.1`. Si le problème est généralisé, désactiver temporairement DNSSEC et vérifier la chaîne de résolution.
Interface web inaccessible après activation HTTPS	Si le certificat est invalide ou la configuration incorrecte, accéder à la console physique du serveur. Utiliser l'option 12 du menu console (*Restore web GUI access defaults*) pour rétablir l'accès en HTTP.