BENCHMARK DE DURCISSEMENT pfSense 2026

🐡 AYI NEDJIMI CONSULTANTS (ANC)

Version : 3.3 — Mai 2026 **Applicabilité :** pfSense CE 2.7.x / 2.8.x — pfSense Plus 24.x **Classification :** CONFIDENTIEL **Auteur :** AYI NEDJIMI Consultants — https://ayinedjimi-consultants.fr **Source :** CIS pfSense Firewall Benchmark v1.1.0 (30-06-2023) + SOCFortress 2025 + Emerging Threats

---

Conventions et niveaux de criticité

NIVEAU	SIGNIFICATION
🔴 **CRITIQUE**	Exploitable sans authentification, patch immédiat
🟠 **ÉLEVÉ**	Risque élevé d'exploitation, action sous 72h
🟡 **MOYEN**	Réduction significative de la surface d'attaque
🟢 **L1**	Baseline CIS — recommandé pour tous
🔵 **L2**	Défense en profondeur — environnements sensibles
⚪ **INFO**	Bonne pratique / observabilité

Format de chaque contrôle : > **CIS Ref** | **MITRE** | **Niveau** > - Description du risque > - Impact potentiel > - Navigation interface / CLI > - CLI de vérification > - Remédiation > - Valeur par défaut > - Critère de conformité

Table des matières

1. Domaine 1 — Configuration initiale et mise à jour du système
2. Domaine 2 — Gestion des utilisateurs et accès administrateur
3. Domaine 3 — Politique de mots de passe et authentification
4. Domaine 4 — Règles de pare-feu et politiques réseau
5. Domaine 5 — Services réseau (DNS, DHCP, NTP, SNMP)
6. Domaine 6 — VPN (OpenVPN, IPsec, WireGuard)
7. Domaine 7 — Paquets et extensions (pfBlockerNG, Snort/Suricata)
8. Domaine 8 — Interface web et accès SSH
9. Domaine 9 — Haute disponibilité et sauvegardes (CARP)

10. Domaine 10 — Journalisation et supervision

- Réponse à incident
- Références
- ANNEXE — Checklists de vérification rapide

Top 10 Quick Wins — 80% du risque en priorité

Ces 10 actions couvrent la majorité des vecteurs d'attaque documentés sur pfSense. Commencer ici avant tout autre contrôle.

#	ACTION	DOMAINE	IMPACT	EFFORT
1	Mettre à jour pfSense vers la dernière version stable	D1	🔴 CRITIQUE	Faible
2	Activer HTTPS uniquement pour la WebGUI, désactiver HTTP	D8	🔴 CRITIQUE	Faible
3	Changer le mot de passe admin par défaut (pfsense)	D3	🔴 CRITIQUE	Faible
4	Désactiver l'accès console sans mot de passe	D2	🔴 CRITIQUE	Faible
5	Supprimer toutes les règles ANY source / ANY destination	D4	🟠 ÉLEVÉ	Moyen
6	Bloquer les réseaux bogons et RFC 1918 sur l'interface WAN	D4	🔴 CRITIQUE	Faible
7	Activer Login Protection (seuil ≤30, délai ≥300s)	D3	🟠 ÉLEVÉ	Faible
8	Activer AutoConfigBackup et tester la restauration	D9	🟠 ÉLEVÉ	Faible
9	Configurer un syslog distant vers SIEM	D10	🟡 MOYEN	Moyen
10		D5	🟡 MOYEN	Faible

#	ACTION	DOMAINE	IMPACT	EFFORT
	Activer DNSSEC + DNS-over-TLS sur le DNS Resolver			

Contrôles additionnels recommandés après les Quick Wins : *Bloquer le DNS-over-HTTPS (DoH) pour forcer la résolution via pfSense (D5 — Contrôle 5.7), automatiser les certificats TLS via ACME/Let's Encrypt (D8 — Contrôle 8.7), et implémenter la segmentation VLAN (D9 — Contrôle 9.5).*

Note — Choisir l'édition adaptée au contexte : *-* ***pfSense CE (Community Edition)*** *: gratuit, open source BSD, support communautaire. Convient aux environnements non critiques. -* ***pfSense Plus*** *: support entreprise Netgate, paquets signés Netgate (protection supply chain), ZFS avec chiffrement, CHARON IKEv2 étendu, patches sécurité prioritaires.* ***Recommandé pour les environnements de production et entreprise.*** *Voir Contrôle 1.7 pour l'analyse comparative. -* ***OPNsense*** *(fork de pfSense) : alternative basée sur HardenedBSD avec options de durcissement supplémentaires et cycle de publication plus rapide.*

Ce benchmark s'applique principalement à pfSense CE 2.7.x/2.8.x et pfSense Plus 24.x.

Domaine 1 — Configuration initiale et mise à jour du système

Objectif : *Établir une base saine dès l'installation de pfSense : vérification de l'intégrité du média d'installation, version à jour, identité réseau configurée, services inutiles désactivés, bannières légales en place.*

Contrôle 1.1 — Mettre à jour pfSense vers la dernière version stable

CIS Ref : *General Setting Policy* | ***MITRE :*** *T1190* | ***Niveau :*** 🔴 *CRITIQUE*

Description du risque

pfSense repose sur FreeBSD et embarque de nombreux composants open source (OpenSSL, PHP, strongSwan, OpenVPN). Des vulnérabilités critiques sont régulièrement découvertes dans ces composants. Les versions obsolètes de pfSense exposent l'organisation à des exploitations distantes sans authentification. Des CVE actifs ont ciblé les interfaces web de pfSense (injection PHP, contournement d'authentification) ainsi que les démons VPN sous-jacents.

CVEs critiques récents à corriger en priorité

CVE	COMPOSANT AFFECTÉ	VERSIONS VULNÉRABLES	DESCRIPTION	REMÉDIATION
CVE-2025-13086	OpenVPN / Widget WebGUI	pfSense CE < 2.8.0, pfSense Plus < 24.x	Injection de commandes OpenVPN Management Interface via le widget WebGUI. Un attaquant authentifié avec des droits OpenVPN Admin peut injecter des commandes arbitraires. **CISA KEV référencé.**	Mettre à jour vers pfSense CE 2.8.0+ ou pfSense Plus 24.x+. Restreindre immédiatement l'accès au widget OpenVPN aux seuls administrateurs nécessaires.
CVE-2024 (SSHGuard bypass)	SSHGuard / WebGUI	pfSense Plus ≤ 22.05.1, CE ≤ 2.6.0	Contournement de la protection contre les tentatives d'authentification via des requêtes web craftées permettant de	Mettre à jour le firmware pfSense. Compléter avec des règles de pare-feu restrictives sur les interfaces d'administration.

CVE	COMPOSANT AFFECTÉ	VERSIONS VULNÉRABLES	DESCRIPTION	REMÉDIATION
			contourner SSHGuard.	

> ***Action immédiate si CVE-2025-13086 applicable :*** *Restreindre l'accès au widget OpenVPN dans* `System > User Manager` *— retirer les droits OpenVPN Admin des comptes non strictement nécessaires, puis planifier la mise à jour sous 30 jours.*

Impact potentiel

- Compromission complète du firewall sans authentification préalable
- Injection de commandes VPN via interface WebGUI (CVE-2025-13086)
- Pivot vers le réseau interne depuis Internet
- Modification silencieuse des règles de pare-feu par un attaquant
- Exfiltration de la configuration complète (identifiants, certificats, clés privées)

Navigation

```
System > Update > System Update
→ Cliquer sur "Check for Updates"
→ Sélectionner la branche stable (Stable)
→ Cliquer sur "Confirm Upgrade"
→ Valider le redémarrage post-mise à jour
```

CLI de vérification

```
cat /etc/version
# Ou depuis la console pfSense (option 12 — PHP Shell) :
system_identify_specific_platform()
```

Remédiation

1. Sauvegarder la configuration avant toute mise à jour : `Diagnostics > Backup & Restore > Download configuration as XML`
2. Noter la version actuelle et consulter les release notes Netgate
3. Planifier la mise à jour en fenêtre de maintenance
4. Appliquer la mise à jour depuis `System > Update`
5. Vérifier le bon démarrage des interfaces et services post-redémarrage
6. Conserver les deux dernières configurations sauvegardées en dehors du firewall

Valeur par défaut : Version installée à la livraison ou au déploiement initial — pas de mise à jour automatique activée par défaut.

Critère de conformité : pfSense à jour selon le canal stable Netgate au moment de l'audit. Délai de patch critique ≤ 30 jours selon la politique de gestion des vulnérabilités de l'organisation.

Contrôle 1.2 — Définir un hostname et un domaine explicites

> ***CIS Ref :*** *1.4 Ensure Hostname is set* | ***MITRE :*** *T1590* | ***Niveau :*** 🟢 *L1*

Description du risque

Le nom d'hôte par défaut de pfSense est `pfSense` et le domaine par défaut est `home.arpa`. Un équipement avec ces valeurs par défaut est immédiatement identifiable lors d'une reconnaissance réseau. De plus, un hostname générique complique la corrélation des logs dans un SIEM et l'identification d'actifs dans un inventaire.

Impact potentiel

- Reconnaissance facilitée de l'équipement réseau (fingerprinting)
- Corrélation des logs impossible ou ambiguë dans un SIEM
- Déploiement incorrect des certificats PKI (SAN/CN erronés)
- Non-conformité aux politiques de nommage d'actifs

Navigation

```
System > General Setup
→ Hostname : saisir le nom d'hôte unique (ex: fw-prd-01)
→ Domain : saisir le domaine DNS interne (ex: corp.exemple.fr)
→ Cliquer sur Save
```

CLI de vérification

```
hostname
# Doit retourner le hostname configuré, pas "pfSense"
cat /etc/hosts | grep -v "^#"
```

Remédiation

1. Aller dans `System > General Setup`
2. Renseigner le champ `Hostname` avec un identifiant unique et descriptif (ex: `fw-paris-01`)
3. Renseigner le champ `Domain` avec le domaine DNS de l'organisation
4. Cliquer sur `Save`
5. Mettre à jour les entrées DNS correspondantes dans le résolveur interne

Valeur par défaut : Hostname `pfSense`, domaine `home.arpa`.

Critère de conformité : Hostname différent de `pfSense` et `localhost`. Domaine correspondant au domaine DNS officiel de l'organisation. Vérification via `hostname` en CLI.

Contrôle 1.3 — Désactiver IPv6 si non utilisé ou sécuriser le dual-stack

CIS Ref : *1.6 Ensure IPv6 is disabled if not used* | ***MITRE :*** *T1562.004, T1190* | ***Niveau :*** 🟢 *L1*

Description du risque

Si IPv6 n'est pas activement utilisé et géré dans l'organisation, le laisser actif augmente inutilement la surface d'attaque. Les règles de pare-feu IPv4 ne protègent pas contre le trafic IPv6. Des attaques de type "IPv6 Bypass" permettent de contourner les politiques de sécurité IPv4 en tunnelant du trafic via des mécanismes de transition IPv6 (6to4, Teredo, ISATAP).

Si IPv6 est activement utilisé (dual-stack) : Des règles de pare-feu équivalentes doivent exister pour IPv4 et IPv6. L'absence de parité de politique (dual-stack policy parity) crée des voies non protégées exploitables par un attaquant. Par défaut, pfSense crée uniquement des règles IPv4 — les règles IPv6 doivent être créées explicitement.

Impact potentiel

- Contournement des règles de pare-feu IPv4 via des tunnels IPv6 (6to4, Teredo, ISATAP)
- Trafic réseau non supervisé via IPv6 natif si les règles IPv4 ne couvrent pas IPv6
- Surface d'attaque élargie sans bénéfice opérationnel si IPv6 non utilisé
- Absence de visibilité dans les logs si les sondes ne capturent pas l'IPv6
- Attaques de type Rogue Router Advertisement (RA spoofing) sur les segments LAN
- NDP Spoofing (équivalent IPv6 de l'ARP poisoning IPv4) non protégé

Navigation

```
# Si IPv6 n'est PAS utilisé — désactiver complètement :
System > Advanced > Networking
→ Section "IPv6 Options"
→ Décocher "Allow IPv6"
→ Cliquer sur Save

# Si IPv6 EST utilisé (dual-stack) — sécuriser :
Firewall > Rules
→ Pour chaque interface où des règles IPv4 existent :
→   Vérifier qu'une règle IPv6 équivalente est présente
→   Protocol : TCP6 / UDP6 / IPv6-ICMP (selon le flux)
→   Créer les règles IPv6 manquantes

# Bloquer les ICMPv6 hors link-local non nécessaires :
Firewall > Rules > WAN
→ Bloquer ICMPv6 type 133-136 (RS/RA/NS/NA) entrant depuis WAN
→ Autoriser uniquement les types ICMPv6 nécessaires :
→   Type 128/129 (Echo Request/Reply) si ping IPv6 requis
→   Type 135/136 (Neighbor Solicitation/Advertisement) sur les interfaces locales uniquement

# Bloquer les mécanismes de transition IPv6 non utilisés :
Firewall > Rules > LAN
→ Bloquer : Protocol IPv6, type 6to4 (RFC 3056 : 2002::/16)
→ Bloquer : Protocol UDP, destination port 3544 (Teredo)
→ Bloquer : Protocol 41 (IPv6-in-IPv4 encapsulation) depuis/vers Internet
```

CLI de vérification

```
# Vérifier si IPv6 est actif :
grep -i "ipv6" /etc/rc.conf
sysctl net.inet6.ip6.forwarding
# La valeur doit être 0 si IPv6 est désactivé

# Si IPv6 actif, vérifier la parité des règles :
pfctl -sr | grep "inet6\|proto tcp6\|proto udp6"
# Comparer avec les règles IPv4 pour s'assurer de la parité

# Vérifier les adresses IPv6 sur les interfaces :
ifconfig | grep inet6
# Si IPv6 désactivé : aucune adresse globale (2000::/3) ne doit apparaître

# Vérifier les tunnels de transition non autorisés :
pfctl -sr | grep -E "proto 41|6to4|teredo|3544"
# Les règles de blocage doivent être présentes
```

Remédiation

Cas 1 — IPv6 non utilisé : 1. Vérifier au préalable qu'aucun service opérationnel ne dépend d'IPv6 2. Aller dans `System > Advanced > Networking` 3. Décocher la case `Allow IPv6` 4. Sauvegarder et vérifier que les interfaces n'ont plus d'adresses IPv6 via `Interfaces > Overview` 5. Ajouter des règles de pare-feu bloquant les protocoles de transition (proto 41, UDP 3544 Teredo)

Cas 2 — IPv6 actif en dual-stack : 1. Auditer toutes les règles IPv4 existantes et créer les équivalents IPv6 2. Bloquer les types ICMPv6 dangereux (RS/RA depuis WAN) 3. Bloquer les mécanismes de transition IPv6 non nécessaires 4. Activer le logging sur toutes les règles IPv6 (même principe que les règles IPv4) 5. S'assurer que le DNS Resolver répond correctement aux requêtes AAAA

Valeur par défaut : IPv6 activé par défaut. Aucune règle IPv6 créée automatiquement à part les règles de blocage anti-spoofing de base.

Critère de conformité : Si IPv6 non requis : case `Allow IPv6` décochée, `ifconfig | grep inet6` ne retourne aucune adresse globale. Si IPv6 utilisé : chaque règle IPv4 a un équivalent IPv6 documenté. Mécanismes de transition bloqués. ICMPv6 RA/RS bloqués depuis le WAN.

Contrôle 1.4 — Activer la vérification DNS Rebind

> ***CIS Ref :*** *1.7 Ensure 'DNS Rebind Check' is configured* | ***MITRE :*** *T1190* | ***Niveau :*** 🟢 *L1*

Description du risque

Les attaques de DNS Rebinding permettent à un site web malveillant de faire croire au navigateur de la victime qu'il communique avec un domaine externe alors qu'il accède en réalité à l'interface d'administration de pfSense. Sans protection contre le DNS Rebinding, l'attaquant peut exécuter des actions administratives sur pfSense depuis le navigateur d'un utilisateur interne.

Impact potentiel

- Accès non autorisé à l'interface d'administration pfSense
- Modification de règles de pare-feu ou de la configuration réseau
- Exfiltration de la configuration complète via le navigateur d'un utilisateur interne
- Escalade vers une compromission totale du réseau

Navigation

```
System > Advanced > Admin Access
→ Section "Login Protection" / "DNS Rebind Check"
→ Cocher "DNS Rebind Check" (Enabled)
→ Dans le champ "Alternate Hostnames", ajouter les noms DNS légitimes de l'interface
→ Cliquer sur Save
```

CLI de vérification

```
grep -i "rebind" /cf/conf/config.xml
# La balise <nodnsrebindcheck> ne doit PAS être présente (ou doit être vide)
```

Remédiation

1. Aller dans `System > Advanced > Admin Access`
2. S'assurer que `DNS Rebind Check` est coché

3. Ajouter dans `Alternate Hostnames` tous les noms DNS légitimes utilisés pour accéder à l'interface (ex: `pfsense.corp.exemple.fr`)
4. Sauvegarder

Valeur par défaut : DNS Rebind Check activé par défaut depuis pfSense 2.5.x.

Critère de conformité : La case `DNS Rebind Check` est cochée ET les noms d'hôtes légitimes sont déclarés dans `Alternate Hostnames`.

Contrôle 1.5 — Configurer la bannière SSH et le MOTD

> ***CIS Ref :*** *1.1 Ensure SSH warning banner is configured / 1.3 MOTD* | ***MITRE :*** *T1078* | ***Niveau :*** 🟢 *L1*

Description du risque

L'absence de bannière d'avertissement avant connexion SSH peut compromettre la capacité juridique à poursuivre un intrus. Dans certaines juridictions, un système sans bannière est considéré comme "ouvert" et l'accès non autorisé peut être difficile à caractériser légalement. La bannière informe également les administrateurs légitimes du cadre d'utilisation du système.

Impact potentiel

- Impossibilité de poursuivre pénalement un intrus en l'absence de bannière légale
- Accès accidentel non signalé par absence d'avertissement
- Non-conformité aux exigences légales et réglementaires (RGPD, LPM, ISO 27001)

Navigation

```
System > Advanced > Admin Access
→ Section "Secure Shell"
→ Activer SSH si nécessaire, configurer le port
→ Puis via CLI pour la bannière :
```

CLI de vérification

```
grep "^Banner" /etc/ssh/sshd_config
cat /etc/issue.net
cat /etc/motd
```

Remédiation

1. Se connecter en SSH ou via la console pfSense
2. Configurer la bannière SSH dans `/etc/ssh/sshd_config` :

   ```
   Banner /etc/issue.net
   ```

3. Créer le fichier `/etc/issue.net` avec un contenu légal approprié :

```
vi /etc/issue.net
```

Contenu recommandé :

```
AVERTISSEMENT — Systeme informatique prive
Acces reserve aux personnes autorisees.
Toute connexion non autorisee est strictement interdite
et fera l'objet de poursuites judiciaires.
Les activites sur ce systeme sont enregistrees et surveillees.
```

4. Configurer le MOTD post-connexion :

```
vi /etc/motd
```

5. Redémarrer le service SSH : `pfSsh.php playback svc restart sshd`

Valeur par défaut : Aucune bannière configurée. Le MOTD par défaut affiche les informations de version FreeBSD.

Critère de conformité : Le fichier `/etc/issue.net` existe et contient un avertissement légal. La directive `Banner /etc/issue.net` est présente dans `/etc/ssh/sshd_config`. Vérification via `grep "^Banner" /etc/ssh/sshd_config`.

Contrôle 1.6 — Vérifier l'intégrité du média d'installation

CIS Ref : *Best Practice* | ***MITRE :*** *T1195.001* | ***Niveau :*** 🔴 *CRITIQUE*

Description du risque

Un média d'installation pfSense dont l'intégrité n'a pas été vérifiée peut avoir été altéré (supply chain attack), soit lors du téléchargement (DNS hijacking, MITM), soit directement sur le dépôt de distribution si ce dernier est compromis. L'installation d'un firmware altéré peut introduire des backdoors persistantes au niveau le plus profond du système, indétectables après coup.

En 2024, des acteurs malveillants ont distribué des ISO pfSense modifiées via des miroirs non officiels. La seule protection est la vérification systématique des checksums SHA256 et des signatures GPG avant tout déploiement.

Impact potentiel

- Backdoor persistante intégrée dans le firmware installé
- Compromission totale et silencieuse dès l'installation initiale
- Règles de pare-feu apparemment correctes mais contournées au niveau noyau
- Impossibilité de détecter la compromission post-installation
- Chaîne de confiance rompue pour toutes les opérations de sécurité ultérieures

Navigation

```
Avant installation (poste de téléchargement) :
→ Télécharger uniquement depuis https://www.netgate.com/pfsense-plus-software/try-pfsense
→   ou https://www.pfsense.org/download/ (CE uniquement)
→ Télécharger le fichier SHA256 correspondant
→ Vérifier la signature GPG avec la clé officielle Netgate

Après installation :
→ Diagnostics > Command Prompt
→ sha256 /bin/pfsense-upgrade (vérification binaire de mise à jour)
```

CLI de vérification

```
# Sur le poste de téléchargement (Linux/macOS) :
sha256sum pfSense-CE-2.7.x-RELEASE-amd64.iso
# Comparer avec le checksum officiel Netgate

# Vérification GPG :
gpg --verify pfSense-CE-2.7.x-RELEASE-amd64.iso.sha256.asc pfSense-CE-2.7.x-RELEASE-
amd64.iso.sha256

# Sur pfSense déployé, vérifier la version installée :
cat /etc/version
cat /etc/version.buildtime
pkg info pfSense
```

Remédiation

1. Télécharger pfSense exclusivement depuis `netgate.com` ou `pfsense.org` (pas de miroirs tiers)
2. Télécharger le fichier de checksum SHA256 associé à l'ISO
3. Vérifier le checksum AVANT de créer le média bootable :

   ```
   sha256sum -c pfSense-CE-*.iso.sha256
   # Doit retourner : pfSense-CE-*.iso: OK
   ```

4. Importer la clé GPG Netgate et vérifier la signature du fichier checksum
5. Documenter le checksum vérifié dans le registre des équipements
6. Pour pfSense Plus Business Edition : utiliser le canal officiel Netgate avec signature de paquet vérifiée automatiquement

Valeur par défaut : Aucune vérification automatique d'intégrité à l'installation — la responsabilité incombe à l'administrateur.

Critère de conformité : Checksum SHA256 vérifié et documenté pour chaque déploiement. Source de téléchargement exclusivement officielle (netgate.com). Processus documenté dans la procédure d'installation.

Contrôle 1.7 — Choisir l'édition pfSense adaptée à l'environnement

CIS Ref : *Best Practice* | ***MITRE :*** *T1195.001* | ***Niveau :*** 🟡 *MOYEN*

Description du risque

Le choix entre pfSense CE (Community Edition) et pfSense Plus a des implications de sécurité directes. pfSense Plus offre des garanties de chaîne d'approvisionnement et des fonctionnalités de sécurité absentes de la version CE, notamment le contrôle d'intégrité des paquets via la signature numérique Netgate et un support réactif pour les avis de sécurité critiques.

Avantages sécuritaires spécifiques de pfSense Plus vs CE :

FONCTIONNALITÉ	PFSENSE CE	PFSENSE PLUS
Paquets signés	Non (paquets non signés)	Oui — seuls les paquets signés Netgate installables (protection supply chain)
Support sécurité	Communautaire uniquement	Support officiel Netgate — avis de sécurité proactifs
Système de fichiers	UFS (pas de chiffrement natif)	ZFS avec support chiffrement natif (GELI)
IPsec IKEv2	strongSwan standard	CHARON IKEv2/strongSwan étendu (plus de fonctionnalités)
HA chiffrée	pfsync non chiffré	Synchronisation HA avec chiffrement possible
Cycle de mises à jour	Plus lent	Plus rapide, patches de sécurité prioritaires
Version recommandée	Environnements non critiques	**Production, environnements sensibles, OIV/OSE**

Impact potentiel

- Avec CE : installation possible de paquets non signés et potentiellement compromis (supply chain attack)
- Sans support officiel : délai de détection et correction des CVE plus long
- Sans ZFS : impossible de chiffrer le stockage local de la configuration (clés VPN, certificats)
- CHARON IKEv2 plus limité en CE pour les fonctionnalités PKI avancées

Navigation

```
# Vérifier l'édition installée :
Diagnostics > Command Prompt
→ Saisir : cat /etc/version.buildtime
→ Ou : cat /etc/platform

# Pour pfSense Plus : vérifier la validité de la licence
System > Netgate Device ID
→ Vérifier que le Device ID est enregistré sur le portail Netgate

# Vérifier l'intégrité de la signature des paquets (pfSense Plus) :
pkg -d info pfSense 2>&1 | grep "signature"
```

CLI de vérification

```
# Identifier l'édition :
cat /etc/platform
# pfSense CE : retourne "pfSense"
# pfSense Plus : retourne "pfSense-plus"

# Vérifier la version exacte :
cat /etc/version
# Comparer avec la dernière version stable sur netgate.com

# Sur pfSense Plus — vérifier la signature des paquets installés :
pkg check --checksum 2>&1 | grep -c "FAILED"
# Doit retourner 0
```

Remédiation

1. Pour les environnements de production et sensibles : migrer vers pfSense Plus 24.x
2. Obtenir une licence pfSense Plus (gratuite pour un usage non commercial via le portail Netgate)
3. Documenter le choix d'édition dans la politique de sécurité avec justification
4. Si CE conservé : renforcer les contrôles de vérification d'intégrité des paquets avant installation
5. Activer la vérification automatique des mises à jour de sécurité disponibles

Valeur par défaut : Choix d'édition à la discrétion de l'administrateur. pfSense CE ne vérifie pas les signatures de paquets.

Critère de conformité : Pour les environnements de production : pfSense Plus déployé. Pour les environnements non critiques : pfSense CE documenté avec justification et mesures compensatoires. Version à jour dans les 30 jours suivant un patch de sécurité critique.

Contrôle 1.8 — Désactiver l'affichage des erreurs PHP

CIS Ref : *Best Practice* | ***MITRE :*** *T1082* | ***Niveau :*** 🟡 *MOYEN*

Description du risque

L'affichage des erreurs PHP dans l'interface d'administration de pfSense révèle des informations sensibles sur la structure interne de l'application : chemins de fichiers, noms de variables, stack traces, versions de composants. Ces informations facilitent considérablement le travail d'un attaquant cherchant à exploiter des vulnérabilités dans l'interface web.

Impact potentiel

- Divulgation de chemins internes du système de fichiers FreeBSD
- Révélation des versions exactes de PHP et des composants
- Traces d'erreur exploitables pour identifier des vecteurs d'injection
- Information gathering facilitant des attaques ciblées sur la WebGUI

Navigation

```
System > Advanced > Admin Access
→ Section "webConfigurator"
→ Décocher "Display PHP Errors" (si disponible selon la version)
→ Cliquer sur Save

Ou via CLI :
→ Diagnostics > Edit File
→ /etc/php.ini ou /usr/local/lib/php.ini
→ Vérifier : display_errors = Off
```

CLI de vérification

```
grep "display_errors" /usr/local/lib/php.ini /usr/local/etc/php.ini 2>/dev/null
# Doit retourner : display_errors = Off
php -r "echo ini_get('display_errors');"
# Doit retourner : 0 (ou vide)
```

Remédiation

1. Aller dans `System > Advanced > Admin Access`
2. Si l'option `Display PHP Errors` est présente, la décocher
3. Vérifier la configuration PHP directement :

   ```
   grep "display_errors" /usr/local/lib/php.ini
   ```

4. Si `display_errors = On`, modifier la valeur à `Off`
5. Redémarrer le service web : `pfSsh.php playback svc restart nginx`
6. Tester qu'aucune erreur PHP n'est visible dans la WebGUI lors d'une navigation normale

Valeur par défaut : Variable selon la version pfSense — les versions récentes désactivent l'affichage par défaut.

Critère de conformité : `display_errors = Off` dans la configuration PHP. Aucun message d'erreur PHP visible dans la WebGUI lors d'une session de navigation standard.

Domaine 2 — Gestion des utilisateurs et accès administrateur

Objectif : *Restreindre l'accès administratif à pfSense aux seules personnes autorisées, en supprimant les comptes par défaut, en activant la protection contre le brute-force et en imposant un timeout de session.*

Contrôle 2.1 — Supprimer ou renommer le compte admin par défaut

CIS Ref : *2.4 Ensure all default accounts are disabled or use strong passwords* | ***MITRE :*** *T1078.001* | ***Niveau :*** 🔴 *CRITIQUE*

Description du risque

Le compte `admin` par défaut de pfSense est universellement connu des attaquants. Son nom d'utilisateur prévisible réduit de moitié le travail d'une attaque par force brute ou par credential stuffing. Les scanners automatisés tentent systématiquement les identifiants `admin/pfsense` sur tout équipement pfSense détecté sur Internet.

Impact potentiel

- Compromission immédiate en cas de mot de passe faible ou par défaut
- Accès administratif complet à l'interface de gestion
- Modification de toutes les politiques de sécurité réseau
- Déploiement de backdoors persistantes dans la configuration

Navigation

```
System > User Manager
→ Cliquer sur le compte "admin"
→ Modifier le nom d'utilisateur (Username) par un identifiant non prévisible
→ Ou : créer un nouveau compte administrateur, lui attribuer le groupe "admins"
→   puis désactiver le compte "admin" d'origine
→ Cliquer sur Save
```

CLI de vérification

```
# Vérifier les comptes locaux dans la configuration
grep -A3 "<user>" /cf/conf/config.xml | grep "<name>"
# Ou via PHP Shell :
# pfSsh.php playback listusers
```

Remédiation

1. Aller dans `System > User Manager`
2. Créer un nouveau compte avec un identifiant unique et non prévisible

3. Lui attribuer les mêmes privilèges que `admin` (groupe `admins`)
4. Configurer un mot de passe fort (≥16 caractères, complexe)
5. Se reconnecter avec le nouveau compte
6. Désactiver (et non supprimer) le compte `admin` d'origine pour éviter tout impact sur des scripts
7. Documenter le nouveau compte administrateur dans le coffre-fort de mots de passe

Valeur par défaut : Compte `admin` avec mot de passe `pfsense` à l'installation initiale.

Critère de conformité : Aucun compte actif avec l'identifiant `admin` et le mot de passe `pfsense`. Le compte administrateur principal a un identifiant non prévisible. Vérification via `System > User Manager`.

Contrôle 2.2 — Activer Login Protection (protection contre le brute-force)

> ***CIS Ref :*** *3.2 Login Protection Threshold ≤30 / 3.3 Allow after ≥300s* | ***MITRE :*** *T1110* | ***Niveau :*** 🟠 *ÉLEVÉ*

Description du risque

Sans protection contre le brute-force, un attaquant peut tester des millions de combinaisons mot de passe sur l'interface web ou SSH de pfSense. La fonctionnalité Login Protection de pfSense utilise `sshguard` pour bloquer automatiquement les adresses IP qui échouent plusieurs fois consécutivement.

Impact potentiel

- Compromission des comptes administrateurs par force brute
- Déni de service sur l'interface de gestion par saturation de tentatives
- Exploitation de mots de passe faibles ou réutilisés

Navigation

```
System > Advanced > Admin Access
→ Section "Login Protection"
→ Cocher "Enable"
→ Threshold : saisir 30 (ou moins)
→ Blocktime : saisir 300 (ou plus, en secondes)
→ Detection time : saisir 3600
→ Pass List : ajouter les IPs des administrateurs de confiance (jamais bloquées)
    Ex: 192.168.99.10, 10.0.1.0/24 (réseau management)
→ Block List : ajouter les IPs malveillantes connues (bloquées immédiatement)
    Ex: adresses issues du Threat Intelligence ou des logs d'attaque récents
→ Cliquer sur Save
```

> ***Important :*** *SSHGuard protège simultanément la WebGUI, SSH et les autres services d'authentification. Les adresses dans la Pass List ne seront* ***jamais*** *bloquées — les ajouter avec précaution. Les adresses de la Block List sont bloquées immédiatement dès leur première tentative.*
>
> ***Considération CVE :*** *Un contournement SSHGuard (CVE-2024) affecte pfSense Plus ≤ 22.05.1 et CE ≤ 2.6.0 via des requêtes web craftées. La mise à jour vers les versions récentes est impérative pour garantir l'efficacité de Login Protection.*

CLI de vérification

```
# Vérifier que sshguard est actif
ps aux | grep sshguard

# Voir les IPs actuellement bloquées par SSHGuard
pfctl -t sshlockout -T show

# Compter le nombre d'IPs bloquées
pfctl -t sshlockout -T show | wc -l

# Vérifier la configuration Login Protection
grep -i "sshguard\|loginprotect" /cf/conf/config.xml

# Surveiller les blocages en temps réel
clog /var/log/auth.log | tail -50 | grep -i "blocked\|sshguard"
```

Remédiation

1. Aller dans `System > Advanced > Admin Access`
2. Dans la section `Login Protection`, cocher `Enable`
3. Définir le seuil (`Threshold`) à `30` ou moins (nombre d'échecs avant blocage)
4. Définir `Blocktime` à `300` secondes minimum
5. Définir `Detection time` à `3600` secondes (fenêtre d'observation)
6. Ajouter les IPs des postes d'administration dans la `Pass List` pour éviter tout auto-blocage accidentel
7. Ajouter les IPs malveillantes connues dans la `Block List` pour un blocage immédiat
8. Sauvegarder
9. Vérifier la liste de blocage active : `pfctl -t sshlockout -T show`

Valeur par défaut : Login Protection désactivée par défaut. Pass List et Block List vides.

Critère de conformité : Login Protection activée, seuil ≤ 30 tentatives, délai de blocage ≥ 300 secondes. IPs de management dans la Pass List. `pfctl -t sshlockout -T show` fonctionnel. Vérification dans `System > Advanced > Admin Access`.

Contrôle 2.3 — Protéger la console par mot de passe

> ***CIS Ref :*** *2.3 Ensure Console Menu is Password Protected* | ***MITRE :*** *T1078.004* | ***Niveau :*** 🟠 *ÉLEVÉ*

Description du risque

Par défaut, pfSense affiche un menu de console accessible sans authentification. Toute personne ayant un accès physique ou via IPMI/iDRAC/ILO peut accéder au menu console et effectuer des opérations sensibles : réinitialisation du mot de passe webGUI, modification des interfaces, activation de SSH. Cela contourne toutes les protections logicielles.

Impact potentiel

- Réinitialisation du mot de passe administrateur sans connaissance préalable des identifiants
- Accès root au système FreeBSD sous-jacent
- Modification des règles de pare-feu sans trace dans les logs applicatifs
- Compromission totale irréversible si l'accès physique n'est pas contrôlé

Navigation

```
System > Advanced > Admin Access
→ Section "Console Options"
→ Cocher "Password protect the console menu"
→ Cliquer sur Save
```

CLI de vérification

```
grep -i "console" /cf/conf/config.xml | grep -i "password\|protect"
# Vérifier la présence de <noconsolemenus/> ou équivalent
```

Remédiation

1. Aller dans `System > Advanced > Admin Access`
2. Dans la section `Console Options`, cocher `Password protect the console menu`
3. Sauvegarder
4. Tester en accédant à la console : le système doit demander l'identifiant et le mot de passe de l'administrateur pfSense
5. Mettre en place des contrôles d'accès physiques complémentaires (verrouillage du rack, BIOS protégé par mot de passe)

Valeur par défaut : Console accessible sans authentification.

Critère de conformité : La case `Password protect the console menu` est cochée. Vérification en accédant à la console physique ou série — une invite d'authentification doit apparaître.

Contrôle 2.4 — Configurer l'authentification LDAP ou RADIUS

CIS Ref : *2.2 Ensure LDAP or RADIUS server configured* | ***MITRE :*** *T1078* | ***Niveau :*** 🔵 *L2*

Description du risque

La gestion des comptes locaux sur chaque firewall est difficile à maintenir dans un parc multi-équipements et ne permet pas une révocation centralisée immédiate. L'intégration à un annuaire d'entreprise (Active Directory via LDAP/LDAPS, ou RADIUS) permet une gestion centralisée des identités, l'application de politiques uniformes et la révocation instantanée d'accès lors d'un départ ou d'un incident.

Impact potentiel

- Comptes orphelins actifs après départ d'un administrateur
- Absence de révocation centralisée et immédiate
- Politique de mots de passe non uniforme entre équipements
- Non-conformité aux exigences d'audit (RGPD, ISO 27001, PCI DSS)

Navigation

```
System > User Manager > Authentication Servers
→ Cliquer sur "+ Add"
→ Type : LDAP ou RADIUS
→ Configurer l'adresse du serveur, le port, la base DN, les identifiants de service
→ Pour LDAP : recommander LDAPS (port 636) avec validation du certificat
→ Pour RADIUS : recommander RADIUS avec attribut d'autorisation de groupe
→ Cliquer sur Save
→ Puis : System > User Manager > Settings
→ Sélectionner le serveur d'authentification configuré
→ Cliquer sur Save
```

CLI de vérification

```
grep -A10 "<authserver>" /cf/conf/config.xml | grep "<type>\|<host>"
```

Remédiation

1. Aller dans `System > User Manager > Authentication Servers`
2. Ajouter un serveur LDAP (recommandé : LDAPS sur port 636 avec certificat valide) ou RADIUS
3. Tester la connexion avec le bouton `Select` puis vérifier les groupes remontés
4. Aller dans `System > User Manager > Settings` et sélectionner le serveur configuré
5. Conserver au minimum un compte local d'urgence (break-glass) avec un mot de passe complexe stocké en coffre-fort

Valeur par défaut : Authentification locale uniquement.

Critère de conformité : Au moins un serveur LDAP ou RADIUS configuré et fonctionnel. L'authentification de production utilise le serveur centralisé. Un compte local d'urgence est documenté et son mot de passe conservé en coffre-fort hors-bande.

Contrôle 2.5 — Configurer un timeout de session ≤ 10 minutes

CIS Ref : *2.1 Ensure Sessions Timeout ≤ 10 Minutes* | ***MITRE :*** *T1078* | ***Niveau :*** 🟢 *L1*

Description du risque

Une session WebGUI non expirée laissée sur un poste de travail sans surveillance expose l'interface d'administration pfSense à toute personne qui accède au poste. Le hijacking de session (vol de cookie de session) est également facilité si les sessions ont une durée de vie longue ou illimitée.

Impact potentiel

- Accès non autorisé à l'interface d'administration via une session ouverte
- Hijacking de session par vol de cookie
- Modifications non autorisées de la configuration réseau
- Non-conformité aux politiques de sécurité (ISO 27001 A.9, PCI DSS 8.1.8)

Navigation

```
System > Advanced > Admin Access
→ Section "webConfigurator"
→ Session Timeout : saisir 10 (minutes)
→ Cliquer sur Save
```

CLI de vérification

```
grep -i "session\|timeout" /cf/conf/config.xml | grep -i "webgui\|timeout"
```

Remédiation

1. Aller dans `System > Advanced > Admin Access`
2. Dans le champ `Session Timeout`, saisir `10` (valeur en minutes)
3. Sauvegarder
4. Vérifier que les sessions expirent correctement après 10 minutes d'inactivité

Valeur par défaut : Timeout de session à 240 minutes (4 heures) par défaut.

Critère de conformité : Session Timeout ≤ 10 minutes. Vérification dans `System > Advanced > Admin Access`, champ `Session Timeout`.

Domaine 3 — Politique de mots de passe et authentification

> ***Objectif :*** *Imposer une politique de mots de passe robuste, désactiver les comptes locaux au profit de l'annuaire d'entreprise, et éliminer tous les mots de passe par défaut.*

Contrôle 3.1 — Désactiver le statut de compte local (Local Account Status)

> ***CIS Ref :*** *3.1 Ensure 'Local Account status' is set to 'Disabled'* | ***MITRE :*** *T1078* | ***Niveau :*** 🔵 *L2*

Description du risque

Lorsqu'une authentification centralisée (LDAP/RADIUS) est configurée, les comptes locaux créent une voie d'accès parallèle qui peut contourner les politiques centralisées de révocation et de politique de mots de passe. Un compte local avec un mot de passe ancien ou par défaut peut rester actif même après la migration vers l'annuaire d'entreprise.

Impact potentiel

- Comptes locaux orphelins contournant la politique centralisée
- Accès persistant d'anciens administrateurs via des comptes locaux non révoqués
- Absence de traçabilité dans le système IAM central

Navigation

```
System > User Manager
→ Pour chaque compte local non essentiel :
→   Cliquer sur le compte > Décocher "Disabled" (ou cocher pour désactiver)
→   Sauvegarder
→ Conserver uniquement le compte de secours (break-glass) local
```

CLI de vérification

```
grep -A5 "<user>" /cf/conf/config.xml | grep "<name>\|<disabled>"
```

Remédiation

1. Aller dans `System > User Manager`
2. Identifier tous les comptes locaux actifs
3. Désactiver tous les comptes qui ne sont pas des comptes de secours
4. Conserver un unique compte local `break-glass` avec un mot de passe de 32 caractères aléatoires stocké en coffre-fort physique
5. Documenter la procédure de réactivation du compte de secours

Valeur par défaut : Comptes locaux actifs — statut par défaut `Enabled`.

Critère de conformité : Seul le compte break-glass local est actif. Tous les autres comptes sont désactivés ou utilisent l'authentification LDAP/RADIUS. Vérification via `System > User Manager`.

Contrôle 3.2 — Changer le mot de passe admin par défaut

> ***CIS Ref :*** *3.4 Ensure default password of admin is changed* | ***MITRE :*** *T1078.001* | ***Niveau :*** 🔴 *CRITIQUE*

Description du risque

Le mot de passe par défaut de pfSense est `pfsense`. Il est documenté publiquement dans toutes les documentations officielles et non officielles. Des scripts d'exploitation automatique et des botnets testent systématiquement ce couple identifiant/mot de passe sur les interfaces pfSense exposées. C'est le vecteur d'attaque le plus simple et le plus fréquemment exploité.

Impact potentiel

- Compromission immédiate de l'interface d'administration
- Accès root au système FreeBSD sous-jacent
- Modification de toutes les règles de pare-feu
- Pivot vers le réseau interne sans détection

Navigation

```
System > User Manager
→ Cliquer sur le compte administrateur
→ Dans le champ "Password", saisir le nouveau mot de passe (≥16 car., complexe)
→ Confirmer dans "Confirm Password"
→ Cliquer sur Save
```

CLI de vérification

```
# Test de connexion avec le mot de passe par défaut (doit échouer)
# Via la CLI, vérifier que le hash du mot de passe n'est pas celui de "pfsense"
grep -A10 "<user>" /cf/conf/config.xml | grep "<bcrypt-hash>\|<password>"
```

Remédiation

1. Aller dans `System > User Manager`
2. Cliquer sur le compte administrateur
3. Définir un mot de passe d'au moins 16 caractères comportant majuscules, minuscules, chiffres et caractères spéciaux
4. Utiliser un gestionnaire de mots de passe pour générer et stocker le mot de passe
5. Documenter le mot de passe dans le coffre-fort de l'organisation (KeePass, Vault, etc.)
6. Ne jamais utiliser le même mot de passe sur plusieurs équipements

Valeur par défaut : `admin` / `pfsense`.

Critère de conformité : Le mot de passe du compte administrateur est différent de `pfsense` et de toute valeur triviale. Hash bcrypt différent de celui du mot de passe par défaut dans `config.xml`. Vérification via tentative de connexion avec `admin/pfsense` — doit échouer.

Contrôle 3.3 — Activer l'authentification à deux facteurs (2FA)

> ***CIS Ref :*** *Best Practice* | ***MITRE :*** *T1110* | ***Niveau :*** 🔵 *L2*

Description du risque

Même avec un mot de passe fort, un administrateur peut être victime de phishing, de keylogging ou de credential stuffing. L'authentification à deux facteurs (2FA) élimine le risque de compromission par simple vol ou devinette du mot de passe, en exigeant un second facteur (TOTP) que l'attaquant ne peut pas obtenir uniquement par compromission du mot de passe.

Impact potentiel

- Sans 2FA : un mot de passe compromis suffit pour un accès complet
- Risque de phishing ciblé des administrateurs réseau
- Credential stuffing depuis des fuites de bases de données publiques

Navigation

```
System > User Manager
→ Cliquer sur le compte administrateur
→ Section "OTP Seed" (One-Time Password)
→ Cliquer sur "Click to create a new TOTP key"
→ Scanner le QR code avec Google Authenticator, Authy ou FreeOTP
→ Cocher "Enabled" pour activer l'OTP
→ Sauvegarder
→ Tester la connexion avec le nouveau code TOTP
```

CLI de vérification

```
grep -A10 "<user>" /cf/conf/config.xml | grep "<otp_seed>"
# La présence d'un seed OTP indique que le 2FA est configuré
```

Remédiation

1. Installer une application TOTP sur le smartphone de l'administrateur (Google Authenticator, Authy, Microsoft Authenticator)
2. Aller dans `System > User Manager`, éditer le compte administrateur
3. Dans la section `OTP Seed`, générer une nouvelle clé TOTP
4. Scanner le QR code affiché avec l'application TOTP
5. Tester en saisissant un code TOTP valide dans le champ prévu
6. Activer et sauvegarder
7. Conserver un code de récupération d'urgence dans le coffre-fort physique

Valeur par défaut : 2FA désactivé par défaut.

Critère de conformité : Chaque compte administrateur avec accès à la WebGUI dispose d'un OTP seed configuré et activé. Vérification via `System > User Manager` — présence du champ `OTP Seed` renseigné.

Domaine 4 — Règles de pare-feu et politiques réseau

> ***Objectif :*** *Appliquer le principe du moindre privilège aux règles de pare-feu : source et destination explicites, services restreints, règle de refus implicite en dernière position, logging systématique. Bloquer les réseaux bogons, RFC 1918 et activer le filtrage de sortie (egress filtering).*

Contrôle 4.1 — Interdire les règles avec source “Any”

> ***CIS Ref :*** *4.1.2 Ensure no Allow Rule with Any in Source field* | ***MITRE :*** *T1190* | ***Niveau :*** 🟠 *ÉLEVÉ*

Description du risque

Une règle d’autorisation avec `Source: Any` permet à n’importe quelle adresse IP, y compris des adresses malveillantes d’Internet, d’accéder au service visé. Ce type de règle est souvent créé par commodité lors de tests puis oublié en production. Il constitue une porte ouverte directe vers des services internes.

Impact potentiel

- Accès non restreint à des services internes depuis n’importe quelle source
- Exploitation de services vulnérables exposés sans restriction
- Réduction à zéro de la protection périmétrique pour les services concernés
- Contournement de toute segmentation réseau

Navigation

```
Firewall > Rules
→ Sélectionner chaque interface (WAN, LAN, OPT1...)
→ Examiner chaque règle d'autorisation (action: Pass)
→ Identifier les règles avec "Source: *" ou "Source: any"
→ Modifier chaque règle pour spécifier une source explicite
→ Ou : désactiver/supprimer les règles non justifiées
```

CLI de vérification

```
pfctl -sr | grep -i "pass.*from any"
# Toute ligne retournée représente une règle non conforme
```

Remédiation

1. Aller dans `Firewall > Rules` pour chaque interface
2. Identifier toutes les règles `Pass` avec `Source: *` (any)
3. Pour chaque règle, déterminer la source légitime et la restreindre à l’adresse IP, au sous-réseau ou à l’alias correspondant
4. Supprimer les règles sans justification métier documentée

5. Mettre à jour la matrice de flux pour refléter les règles validées
6. Valider après modification que les flux légitimes fonctionnent toujours

Valeur par défaut : Règles `Allow Any` par défaut sur LAN pour éviter le lockout initial.

Critère de conformité : Aucune règle `Pass` avec `Source: any` sauf exception documentée et approuvée. Vérification via `pfctl -sr | grep "pass.*from any"` — résultat vide.

Contrôle 4.2 — Interdire les règles avec destination “Any”

> ***CIS Ref :*** *4.1.1 Ensure no Allow Rule with Any in Destination Field* | ***MITRE :*** *T1048* | ***Niveau :*** 🟠 *ÉLEVÉ*

Description du risque

Une règle d’autorisation avec `Destination: Any` permet à la source de joindre n’importe quelle adresse de destination, y compris des adresses de commande et contrôle (C2) sur Internet, ou des segments réseau internes normalement isolés. Ce type de règle facilite considérablement l’exfiltration de données et les mouvements latéraux.

Impact potentiel

- Exfiltration de données vers des serveurs C2 sans restriction
- Mouvements latéraux vers n’importe quel segment réseau
- Contournement de la segmentation réseau et des DMZ
- Communication directe avec des infrastructures malveillantes

Navigation

```
Firewall > Rules
→ Sélectionner chaque interface
→ Identifier les règles avec "Destination: *" ou "Destination: any"
→ Restreindre chaque destination à l'adresse, sous-réseau ou alias légitime
→ Sauvegarder et appliquer les modifications
```

CLI de vérification

```
pfctl -sr | grep -i "pass.*to any"
# Analyser chaque ligne retournée
```

Remédiation

1. Inventorier tous les flux réseau légitimes via la matrice de flux
2. Pour chaque règle `Pass` avec `Destination: any`, définir la destination précise
3. Utiliser les alias pfSense pour regrouper des destinations légitimes multiples
4. Documenter toute exception avec justification métier
5. Implémenter une règle DENY finale explicite en dernière position de chaque interface

Valeur par défaut : Règles `Allow Any` destination par défaut sur LAN.

Critère de conformité : Aucune règle `Pass` avec `Destination: any` sauf exception documentée. `pfctl -sr | grep "pass.*to any"` retourne uniquement des règles justifiées.

Contrôle 4.3 — Interdire les règles avec service “Any”

CIS Ref : *4.1.3 Ensure no Allow Rule with Any in Services field* | ***MITRE :*** *T1071* | ***Niveau :*** 🟠 *ÉLEVÉ*

Description du risque

Une règle autorisant tous les services (protocoles et ports) entre une source et une destination ouvre la voie à l’utilisation de protocoles non standards pour l’exfiltration ou le C2 (DNS over HTTPS tunnelé, ICMP tunnelé, ports élevés non surveillés). Les services légitimes n’ont besoin que d’un ensemble limité de ports.

Impact potentiel

- Tunneling de trafic malveillant via des protocoles non surveillés
- Exfiltration via des ports et protocoles non restreints
- Communication C2 masquée dans des flux apparemment légitimes
- Exploitation de services non intentionnellement exposés

Navigation

```
Firewall > Rules
→ Sélectionner chaque interface
→ Identifier les règles avec "Destination Port: *" ou "Protocol: any" sans restriction
→ Modifier pour spécifier le protocole (TCP/UDP) et le port exact
→ Utiliser des alias de ports pour les groupes de services liés
```

CLI de vérification

```
pfctl -sr | grep "pass" | grep -v "proto tcp\|proto udp\|proto icmp"
# Identifier les règles sans restriction de protocole
pfctl -sr | grep "pass.*port ="
# Vérifier que les ports sont explicites
```

Remédiation

1. Documenter la matrice de flux avec protocole et port précis pour chaque flux
2. Modifier chaque règle `Pass` pour spécifier le protocole (TCP, UDP, ICMP) et le port de destination
3. Créer des alias de ports pour faciliter la gestion des groupes (ex: alias `WEB` = TCP 80, 443)
4. Supprimer ou désactiver les règles sans restriction de service sans justification documentée

Valeur par défaut : Règles par défaut sans restriction de protocole/port sur LAN.

Critère de conformité : Toutes les règles `Pass` ont un protocole et un port de destination explicites. Aucune règle `Pass` avec `Protocol: any` et `Port: any` simultanément.

Contrôle 4.4 — Supprimer les règles inutilisées

CIS Ref : *4.1.4 Ensure there are no Unused Policies* | ***MITRE :*** *T1562.004* | ***Niveau :*** 🟢 *L1*

Description du risque

Les règles désactivées ou jamais utilisées encombrent la politique de sécurité et créent un risque de réactivation accidentelle ou malveillante. Elles complexifient les audits et peuvent masquer des règles permissives oubliées. Une politique de pare-feu propre est plus facile à maintenir et à auditer.

Impact potentiel

- Réactivation accidentelle de règles permissives lors d'une modification
- Complexité de la politique rendant les audits inefficaces
- Règles orphelines masquant des accès non intentionnels
- Non-conformité aux exigences de gestion des règles (PCI DSS 1.2)

Navigation

```
Firewall > Rules
→ Examiner chaque règle désactivée (icône grise)
→ Vérifier via Status > System Logs > Firewall si la règle a eu des hits
→ Supprimer les règles sans activité et sans justification documentée
→ Documenter les règles conservées désactivées avec justification
```

CLI de vérification

```
pfctl -sr | grep -c "pass\|block"
# Comparer avec le nombre de règles visibles dans la GUI
# Des écarts importants peuvent indiquer des règles cachées
pfctl -vsr
# Affiche les compteurs de hits par règle
```

Remédiation

1. Aller dans `Firewall > Rules` pour chaque interface
2. Identifier toutes les règles désactivées
3. Vérifier dans les logs si ces règles ont eu des hits récents
4. Supprimer les règles sans activité dans les 90 derniers jours sans justification documentée
5. Conserver un processus de revue annuelle des règles de pare-feu

Valeur par défaut : Les règles désactivées restent dans la configuration indéfiniment.

Critère de conformité : Aucune règle désactivée sans justification documentée. Revue des règles effectuée dans les 12 derniers mois avec documentation. `pfctl -vsr` montre des compteurs de hits > 0 pour les règles actives.

Contrôle 4.5 — Activer le logging sur toutes les règles

> ***CIS Ref :*** *4.1.5 Ensure Logging is Enable for All Firewall Rules* | ***MITRE :*** *T1562.006* | ***Niveau :*** 🟢 *L1*

Description du risque

Sans logging des règles de pare-feu, il est impossible de détecter des tentatives d'intrusion, des mouvements latéraux ou des comportements anormaux. Le logging est un prérequis fondamental pour la détection d'incidents, l'investigation forensique et la conformité réglementaire.

Impact potentiel

- Impossibilité de détecter des tentatives d'intrusion et de les corréler
- Investigation forensique impossible lors d'un incident de sécurité
- Non-conformité aux exigences légales de traçabilité (LPM, RGPD, PCI DSS)
- Délai de détection des incidents multiplié par absence de logs

Navigation

```
Firewall > Rules
→ Éditer chaque règle
→ Dans "Extra Options", cocher "Log"
→ Sauvegarder

Status > System Logs > Settings
→ Activer les options :
   - Log firewall default blocks
   - Log packets blocked by 'Block Bogon Networks' rules
   - Log packets blocked by 'Block Private Networks' rules
   - Web Server Log
   - Log Configuration Changes
→ Sauvegarder
```

CLI de vérification

```
pfctl -vsr | grep -i "log"
# Vérifier que les règles actives ont le flag "log"
tail -f /var/log/filter.log
# Observer le flux de logs en temps réel
```

Remédiation

1. Aller dans `Firewall > Rules`
2. Éditer chaque règle et cocher la case `Log` dans les options avancées
3. Aller dans `Status > System Logs > Settings`
4. Activer toutes les options de logging pertinentes
5. Configurer un serveur syslog distant (voir Domaine 10) pour la rétention long terme
6. Attention : le logging intensif peut impacter les performances sur du matériel limité — adapter la granularité selon les ressources disponibles

Valeur par défaut : Logging activé uniquement sur les règles de blocage par défaut. Les règles personnalisées n'ont pas le logging activé par défaut.

Critère de conformité : Toutes les règles `Pass` et `Block` ont la case `Log` cochée. `pfctl -vsr | grep log` retourne une ligne pour chaque règle active. Logs visibles dans `Status > System Logs > Firewall`.

Contrôle 4.6 — Configurer ICMP de façon sécurisée et bloquer les bogons/RFC1918 sur WAN

CIS Ref : *4.1.6 Ensure ICMP Request is securely configured* | ***MITRE :*** *T1018, T1557, T1498* | ***Niveau :*** 🟡 *MOYEN*

Description du risque

Deux vecteurs complémentaires doivent être adressés simultanément :

1. ICMP non restreint : Autoriser tous les types de messages ICMP expose à des attaques DoS (Ping Flood), de détournement de routage (ICMP Redirect), de collecte d'informations (ICMP Timestamp), et de tunneling pour l'exfiltration de données.

2. Bogons et RFC 1918 non bloqués sur WAN : Des paquets avec des adresses sources appartenant aux espaces privés (RFC 1918 : 10.0.0.0/8, 172.16.0.0/12, 192.168.0.0/16) ou aux espaces d'adressage non alloués (bogons) ne devraient jamais arriver depuis Internet. Leur présence indique soit une attaque de spoofing d'adresse IP, soit un équipement mal configuré. L'absence de blocage permet des attaques de type Adversary-in-the-Middle par ARP poisoning et des tentatives d'accès déguisées.

Impact potentiel

- Attaques DoS par Ping Flood et ICMP Redirect
- Collecte d'informations via ICMP Timestamp et Netmask Request
- Tunneling de données via ICMP (exfiltration discrète)
- Usurpation d'adresses IP privées depuis Internet (spoofing)
- Contournement de règles de pare-feu basées sur la source IP
- Attaques T1557 (Adversary-in-the-Middle) via ARP spoofing facilitées

Navigation

```
# Blocage ICMP sécurisé :
Firewall > Rules
→ Pour les règles ICMP existantes :
→   Protocol : ICMP
→   ICMP Type : "Echo Request" (type 8) et "Echo Reply" (type 0) uniquement
→   Bloquer : Type 5 (Redirect), Type 13 (Timestamp), Type 17 (Netmask Request)

# Blocage bogons et RFC 1918 sur WAN :
Interfaces > [WAN]
→ Cocher "Block private networks and loopback addresses"
→ Cocher "Block bogon networks"
→ Cliquer sur Save

Firewall > Rules > WAN
→ Vérifier la présence des règles automatiques :
→   "Block private networks" (RFC 1918 + loopback)
→   "Block bogon networks" (espaces non alloués)
```

CLI de vérification

```
pfctl -sr | grep icmp
# Vérifier que seuls les types ICMP autorisés sont présents

pfctl -sr | grep -E "block.*10\.|block.*172\.16\.|block.*192\.168\."
# Vérifier le blocage des espaces RFC 1918

grep -i "blockpriv\|blockbogon" /cf/conf/config.xml
# Vérifier la configuration dans config.xml
```

Remédiation

1. Aller dans `Interfaces > WAN`
2. Cocher `Block private networks and loopback addresses`
3. Cocher `Block bogon networks`
4. Sauvegarder et appliquer
5. Aller dans `Firewall > Rules` sur chaque interface
6. Remplacer les règles `ICMP: any` par des règles avec types ICMP spécifiques
7. Autoriser uniquement `Echo Request` (type 8) et `Echo Reply` (type 0) si le ping est requis
8. Créer des règles de blocage explicites pour les types ICMP dangereux
9. Activer le logging sur toutes les règles ICMP et de blocage WAN

Valeur par défaut : Blocage bogons et RFC 1918 configurable lors de l'assistant de configuration initiale — parfois non activé si ignoré. Règles ICMP permissives sur LAN.

Critère de conformité : Cases `Block private networks` et `Block bogon networks` cochées sur l'interface WAN. Aucune règle `Pass proto icmp` sans restriction de type ICMP. Types ICMP Redirect (5), Timestamp (13) et Netmask Request (17) bloqués.

Contrôle 4.7 — Activer l'anti-spoofing sur toutes les interfaces

CIS Ref : *Best Practice* | ***MITRE :*** *T1557, T1036* | ***Niveau :*** 🟠 *ÉLEVÉ*

Description du risque

Le spoofing d'adresse IP consiste à envoyer des paquets avec une adresse source falsifiée pour masquer l'origine réelle d'une attaque ou usurper l'identité d'un hôte de confiance. Sans protection anti-spoofing, un attaquant interne peut émettre du trafic au nom d'adresses IP d'autres segments réseau, contournant les règles de pare-feu basées sur la source. Cette technique est associée aux attaques T1557 (Adversary-in-the-Middle) et est un vecteur clé dans les attaques de réflexion DDoS (T1498).

Impact potentiel

- Contournement des ACL basées sur l'adresse IP source
- Amplification DDoS utilisant pfSense comme réflecteur involontaire
- Mouvements latéraux masqués par usurpation d'adresse IP d'un segment de confiance
- Attaques de poisoning ARP et MITM facilitées

Navigation

```
System > Advanced > Firewall & NAT
→ Section "Firewall Advanced"
→ Cocher "Disable Firewall Scrub" : NE PAS cocher (garder le scrubbing actif)

Firewall > Settings > Normalization
→ Vérifier que le scrubbing est actif sur toutes les interfaces

# Anti-spoofing via règles flottantes :
Firewall > Rules > Floating
→ Créer des règles anti-spoofing par interface :
→   Interface WAN : bloquer les paquets avec source = réseaux internes
→   Interface LAN : bloquer les paquets avec source hors du sous-réseau LAN assigné
→   Repeat pour chaque interface
```

CLI de vérification

```
# Vérifier le scrubbing pfSense (normalisation des paquets) :
pfctl -vsr | grep -i "scrub"

# Vérifier les règles scrub actives (syntaxe complète) :
pfctl -s rules | grep scrub

# Vérifier les règles anti-spoofing dans les règles flottantes :
pfctl -sr | grep "antispoof"

# Sur FreeBSD, vérifier uRPF si disponible :
sysctl net.inet.ip.check_interface
```

Remédiation

1. Activer le scrubbing sur toutes les interfaces dans `Firewall > Settings > Normalization`
2. Créer des règles flottantes anti-spoofing pour chaque interface :
    - Sur WAN : bloquer les sources correspondant aux espaces RFC 1918 (si pas déjà bloqués via contrôle 4.6)
    - Sur LAN : bloquer les sources hors du sous-réseau assigné à cette interface
    - Sur chaque OPT : même principe
3. Activer `Block private networks` et `Block bogon networks` sur le WAN (voir contrôle 4.6)
4. Tester avec des outils de spoofing contrôlés en environnement de lab

Valeur par défaut : Scrubbing activé par défaut. Règles anti-spoofing explicites non configurées par défaut.

Critère de conformité : Scrubbing actif sur toutes les interfaces. Règles flottantes anti-spoofing présentes et loggées. `pfctl -vsr | grep scrub` retourne au moins une règle par interface.

Contrôle 4.8 — Activer le filtrage de sortie (Egress Filtering)

> ***CIS Ref :*** *Best Practice* | ***MITRE :*** *T1048, T1071.001, T1041* | ***Niveau :*** 🟠 *ÉLEVÉ*

Description du risque

Le filtrage de sortie (egress filtering) consiste à contrôler le trafic quittant le réseau interne vers Internet. Sans egress filtering, un malware infectant un hôte interne peut librement communiquer avec ses serveurs C2 (Command and Control) sur n'importe quel port, exfiltrer des données via des protocoles alternatifs (DNS, ICMP, HTTP), et établir des reverse shells. L'egress filtering est l'un des contrôles les plus efficaces contre les attaques T1048 (Exfiltration Over Alternative Protocol) et T1071 (Application Layer Protocol C2).

Impact potentiel

- Communications C2 non bloquées depuis des hôtes compromis
- Exfiltration de données via protocoles non surveillés (DNS, ICMP, NTP)
- Reverse shells vers des infrastructures malveillantes non détectés
- Non-conformité (PCI DSS 1.2.1 : restriction du trafic sortant)

Navigation

```
Firewall > Rules > LAN (et chaque interface interne)
→ Supprimer les règles "Pass Any to Any" existantes
→ Créer des règles de sortie explicites pour les flux légitimes :
→   Pass TCP from LAN to any port 80, 443 (navigation web)
→   Pass UDP from LAN to DNS_Servers port 53 (résolution DNS — vers pfSense uniquement)
→   Pass TCP from SMTP_Servers to any port 25, 465, 587
→   Pass règles spécifiques pour les flux métier documentés
→ Ajouter en dernière position :
→   Block + Log : from LAN to any (règle de refus implicite explicite)

# Alias utiles à créer :
Firewall > Aliases
→ Créer un alias "RFC1918_Internal" avec tous les sous-réseaux internes
→ Créer un alias "DNS_Servers" avec les IPs des résolveurs autorisés
```

CLI de vérification

```
# Vérifier l'absence de règle "pass all" sortante non restreinte :
pfctl -sr | grep "pass.*from.*LAN.*to.*any" | grep -v "port"
# Toute ligne sans restriction de port est une règle non conforme

# Tester l'egress filtering en tentant une connexion sur un port non autorisé :
# Depuis un hôte interne : nc -zv 8.8.8.8 23 (doit être bloqué)
```

Remédiation

1. Inventorier tous les flux sortants légitimes (matrice de flux)
2. Créer des alias pour les destinations et groupes de services
3. Créer des règles explicites pour chaque flux légitime avec protocole et port précis
4. Ajouter une règle de blocage explicite et loggée en fin de chaîne
5. Tester pendant 24-48h en mode IDS (logging uniquement sans blocage) avant d'activer
6. Compléter avec pfBlockerNG (listes de réputation IP) et Suricata (inspection deep packet)

Valeur par défaut : LAN autorisé à accéder à Internet sans restriction par défaut.

Critère de conformité : Règle de blocage implicite explicite en dernière position sur chaque interface interne. Seuls les flux documentés et approuvés sont autorisés en sortie. `pfctl -sr | grep "pass.*from LAN"` retourne uniquement des règles avec protocole et port spécifiés.

Contrôle 4.9 — Activer le scrubbing / normalisation de trafic

CIS Ref : *Best Practice* | ***MITRE :*** *T1499.002, T1027* | ***Niveau :*** 🟠 *ÉLEVÉ*

Description du risque

Le scrubbing (normalisation de paquets) de pfSense, géré par `pf`, réassemble les fragments IP et normalise les en-têtes TCP/IP avant inspection. Sans scrubbing, les attaquants peuvent exploiter des techniques d'évasion IDS via des paquets fragmentés délibérément malformés, des séquences TCP chevauchantes ou des options IP non conformes. Ces techniques permettent de contourner les règles de pare-feu et les systèmes de détection d'intrusion.

Techniques d'évasion bloquées par le scrubbing : - Fragmentation IP pour contourner les ACL (les fragments partiels ne correspondent pas aux règles) - TCP session splicing via des paquets TCP hors séquence - Utilisation d'options IP non standard pour l'évasion d'inspection - Attaques de type TTL manipulation pour créer des vues divergentes entre IDS et cible

Impact potentiel

- Contournement des règles de pare-feu par fragmentation IP
- Évasion de Suricata/IDS via des paquets intentionnellement fragmentés
- Exploitation de services internes via des payloads fragmentés non reconstitués
- Attaques T1027 (Obfuscated Files or Information) via fragmentation réseau

Navigation

```
System > Advanced > Firewall & NAT
→ Section "Firewall Optimization Options"
→ Firewall Optimization Algorithm : sélectionner "Normal" ou "Conservative"
→ Disable Firewall Scrub : LAISSER DÉCOCHÉ (scrubbing activé)
→ Ne JAMAIS cocher "Disable Firewall Scrub" sur les interfaces exposées à Internet

System > Advanced > Firewall & NAT
→ Section "Miscellaneous"
→ Vérifier que "Disable Scrub" n'est pas activé

# Options avancées de scrubbing via règles pf personnalisées :
System > Advanced > Firewall & NAT > Custom Options
→ Ou via Diagnostics > Edit File > /etc/pf.conf.local
→ Ajouter des options de normalisation avancées :
→   scrub in on $wan all fragment reassemble random-id max-mss 1460 no-df
→   scrub out on $wan all random-id
```

> ***Avertissement :*** *Ne jamais désactiver le scrubbing sur les interfaces WAN ou DMZ exposées à Internet. La désactivation du scrubbing est uniquement justifiée dans de rares cas avec des équipements legacy incompatibles avec la normalisation IP, et uniquement sur des interfaces internes isolées.*

Options de normalisation avancées :

OPTION	DESCRIPTION	BÉNÉFICE SÉCURITÉ
`random-id`	Randomise le champ ID des en-têtes IP	Prévient le fingerprinting de l'OS et les corrélations entre paquets
`no-df`	Efface le bit "Don't Fragment" des paquets fragmentés	Permet la réassemblée des fragments même si DF est positionné
`max-mss 1460`	Limite le MSS TCP à 1460 octets	

OPTION	DESCRIPTION	BÉNÉFICE SÉCURITÉ
		Prévient la fragmentation excessive et les attaques de TCP MSS
`fragment reassemble`	Réassemble les fragments IP avant inspection	Bloque les évasions IDS par fragmentation
`min-ttl 15`	Impose un TTL minimum	Réduit l'efficacité du TTL manipulation pour les vues divergentes

CLI de vérification

```
# Vérifier les règles scrub actives
pfctl -s rules | grep scrub

# Vérifier l'absence de l'option "no-scrub" non justifiée :
pfctl -sr | grep "no scrub"
# Toute ligne retournée représente une exception — vérifier la justification

# Vérifier la configuration dans config.xml :
grep -i "scrub\|noscrub" /cf/conf/config.xml

# Vérifier les options avancées actives (random-id, max-mss) :
pfctl -sr | grep -E "random-id|max-mss|no-df|fragment"

# Statistiques de normalisation en temps réel :
pfctl -s info | grep -A5 "State Table\|Counters"

# Compteurs de fragments réassemblés (indicateur de tentatives d'évasion) :
pfctl -s info | grep -i "frag"
```

Remédiation

1. Aller dans `System > Advanced > Firewall & NAT`
2. S'assurer que `Disable Firewall Scrub` est **décoché** (scrubbing actif)
3. Sélectionner `Firewall Optimization Algorithm: Normal` (ou `Conservative` pour les connexions lentes/longues)
4. Pour les environnements nécessitant un durcissement avancé, ajouter les options via `Custom Options` :

   ```
   scrub in on $wan all fragment reassemble random-id max-mss 1460 no-df
   scrub out on $wan all random-id
   ```

5. Sauvegarder
6. Vérifier avec `pfctl -s rules | grep scrub` que des règles de normalisation sont présentes
7. Si des règles `no scrub` existent, documenter formellement leur justification
8. Monitorer les compteurs de fragments dans les logs pour détecter des tentatives d'évasion

Valeur par défaut : Scrubbing activé par défaut avec options de base. Options avancées (random-id, max-mss) non configurées par défaut.

Critère de conformité : `pfctl -s rules | grep scrub` retourne au moins une règle de normalisation. Aucune règle `no scrub` sur les interfaces WAN ou DMZ sans justification documentée. La case `Disable Firewall Scrub` est décochée. Pour les environnements L2 : `random-id` et `max-mss` configurés sur l'interface WAN.

Domaine 5 — Services réseau (DNS, DHCP, NTP, SNMP)

Objectif : *Sécuriser et restreindre les services réseau au minimum nécessaire, en activant les mécanismes de sécurité disponibles (DNSSEC, DNS-over-TLS, SNMPv3) et en désactivant les services non utilisés (UPnP, IGMP, SNMP v1/v2).*

Contrôle 5.1 — Activer DNSSEC et DNS-over-TLS sur le résolveur DNS

CIS Ref : *5.3.1 Ensure 'DNSSEC' is Enable on DNS Service* | ***MITRE :*** *T1557* | ***Niveau :*** 🟢 *L1*

Description du risque

DNSSEC : Sans validation DNSSEC, les requêtes DNS peuvent être falsifiées par des attaques DNS Spoofing ou DNS Cache Poisoning (attaque Kaminsky). Un attaquant peut rediriger les utilisateurs vers des sites malveillants en falsifiant les réponses DNS.

DNS-over-TLS (DoT) : Sans chiffrement des requêtes DNS, toutes les résolutions DNS transitent en clair sur le réseau. Un attaquant sur le même réseau (FAI, réseau public, équipement réseau compromis) peut intercepter l'intégralité de l'activité DNS, révélant les domaines consultés et permettant des attaques de résolution DNS malveillantes. DNS-over-TLS via Stubby ou Unbound (port 853) chiffre les requêtes entre pfSense et les résolveurs upstream.

Impact potentiel

- Redirection des utilisateurs vers des sites de phishing (DNS Spoofing)
- Empoisonnement du cache DNS (attaque Kaminsky)
- Surveillance de l'activité DNS par des tiers non autorisés (FAI, attaquants)
- Interception et modification des réponses DNS par un MITM sur le réseau upstream

Navigation

```
# Activer DNSSEC :
Services > DNS Resolver
→ Cocher "Enable DNSSEC Support"

# Activer DNS-over-TLS avec Unbound :
Services > DNS Resolver > Custom Options
→ Ajouter dans "Custom options" :
   forward-zone:
     name: "."
     forward-tls-upstream: yes
     forward-addr: 9.9.9.9@853#dns.quad9.net
     forward-addr: 149.112.112.112@853#dns.quad9.net
     forward-addr: 1.1.1.1@853#cloudflare-dns.com
     forward-addr: 1.0.0.1@853#cloudflare-dns.com

→ Cliquer sur Save → Apply Changes

# Alternative : utiliser le paquet Stubby (DNS-over-TLS dédié) :
System > Package Manager > Available Packages
→ Installer "stubby"
Services > Stubby
→ Configurer les résolveurs DoT (Quad9, Cloudflare, NextDNS)
```

CLI de vérification

```
# Tester la validation DNSSEC (flag AD = Authenticated Data) :
drill -D sigok.verteiltesysteme.net @127.0.0.1
# La réponse doit contenir le flag "ad" (Authenticated Data)

# Test alternatif via dig :
dig +dnssec sigok.verteiltesysteme.net @127.0.0.1
# Vérifier la présence du flag "ad" dans la section "flags:"

# Tester un domaine DNSSEC invalide — doit retourner SERVFAIL (comportement sécurisé) :
dig sigfail.verteiltesysteme.net @127.0.0.1
# Doit retourner SERVFAIL (Unbound refuse les signatures invalides)

# Tester DNS-over-TLS directement :
kdig -d @9.9.9.9 +tls-ca +tls-host=dns.quad9.net example.com
# Doit résoudre sans erreur TLS

# Vérifier le statut d'Unbound :
unbound-control status
# Doit retourner "is running" avec le PID

# Statistiques DNSSEC Unbound (validations réussies/échouées) :
unbound-control stats | grep dnssec
# Clés : rrset.bogus (signatures rejetées), rrset.secure (signatures valides)

# Vérifier la configuration Unbound :
cat /var/unbound/unbound.conf | grep -A5 "forward-zone"

# Vérifier les root hints à jour :
ls -la /var/unbound/root.hints
```

Remédiation

1. Aller dans `Services > DNS Resolver`
2. Cocher `Enable DNSSEC Support`
3. Ajouter la configuration DNS-over-TLS dans `Custom options`
4. Configurer des serveurs upstream supportant DoT (Quad9, Cloudflare, NextDNS)
5. Sauvegarder et appliquer
6. Tester la validation DNSSEC et la connectivité DoT
7. Bloquer les requêtes DNS directes (port 53) depuis les clients vers Internet via les règles de pare-feu — forcer le passage par le résolveur pfSense

Valeur par défaut : DNSSEC désactivé. DNS-over-TLS non configuré. Résolution DNS en clair.

Critère de conformité : DNSSEC activé. DNS-over-TLS configuré avec au moins deux résolveurs upstream. `dig +dnssec sigok.verteiltesysteme.net @127.0.0.1` retourne le flag `ad`. Aucune requête DNS en clair vers Internet depuis pfSense.

Contrôle 5.2 — SNMP : v3 uniquement ou désactivé

CIS Ref : *5.1 SNMP Policy* | ***MITRE :*** *T1602* | ***Niveau :*** 🟠 *ÉLEVÉ*

Description du risque

SNMP v1 et v2c utilisent des community strings en clair sur le réseau, sans authentification ni chiffrement. Ces community strings sont souvent laissées aux valeurs par défaut (`public` , `private`). Un attaquant sur le réseau peut intercepter les community strings et accéder à toutes les informations MIB, voire modifier la configuration (SNMP write). Si SNMP n'est pas nécessaire, il doit être désactivé.

Impact potentiel

- Collecte d'informations complètes sur la configuration réseau (tables ARP, routes, interfaces)
- Modification de la configuration réseau via SNMP write (v2c)
- Community strings interceptées facilement sur un réseau non chiffré
- Reconnaissance avancée facilitant les attaques ultérieures

Navigation

```
Services > SNMP
→ Si SNMP n'est pas nécessaire : décocher "Enable"
→ Si SNMP est requis :
→   Paquet NET-SNMP : System > Package Manager > Available Packages
→   Installer "net-snmp"
→   Configurer uniquement SNMPv3 avec authentification SHA et chiffrement AES
→   Définir un username et mot de passe complexes
→   Restreindre l'accès par ACL aux seules adresses NMS autorisées
→ Cliquer sur Save
```

CLI de vérification

```
# Vérifier si le démon SNMP est actif
ps aux | grep snmpd
# Vérifier la configuration SNMP
grep -i "snmp" /cf/conf/config.xml | grep "<enable>"
# Tester depuis un NMS si SNMPv1/v2 est refusé :
# snmpwalk -v2c -c public <IP_pfSense> (doit échouer)
```

Remédiation

1. Si SNMP n'est pas nécessaire : aller dans `Services > SNMP`, décocher `Enable`, sauvegarder
2. Si SNMP est requis :
    - Installer le paquet `net-snmp` via `System > Package Manager`
    - Configurer uniquement des utilisateurs SNMPv3 avec `authPriv` (SHA + AES256)
    - Définir une ACL restreignant l'accès aux seules adresses IP du NMS
    - Changer les community strings par défaut si SNMP v2c est temporairement utilisé
3. Bloquer le port UDP 161 dans les règles de pare-feu pour toutes les interfaces sauf management

Valeur par défaut : SNMP désactivé par défaut. Si activé, community string `public` par défaut.

Critère de conformité : SNMP désactivé si non nécessaire. Si actif, uniquement SNMPv3 avec authPriv (SHA + AES). Community strings `public` et `private` inexistantes. Accès restreint par ACL aux adresses NMS autorisées.

Contrôle 5.3 — Configurer des serveurs NTP fiables et le fuseau horaire

> ***CIS Ref :*** *5.2.1 Ensure time zone is properly configured* | ***MITRE :*** *T1562* | ***Niveau :*** 🟢 *L1*

Description du risque

Une horloge système incorrecte invalide tous les timestamps des logs, rendant l'investigation forensique et la corrélation d'événements impossibles. Des attaques de type NTP Amplification peuvent exploiter un serveur NTP mal configuré. L'utilisation de serveurs NTP non authentifiés expose à des attaques de manipulation de l'heure (NTP Spoofing) qui peuvent invalider des certificats ou déclencher des timeouts de session non souhaités.

Impact potentiel

- Timestamps incorrects rendant les logs inutilisables pour l'investigation
- Invalidation de certificats TLS (expiration prématurée ou retardée)
- Attaques de replay facilitées par un décalage horaire
- Corrélation impossible entre équipements lors d'un incident

Navigation

```
System > General Setup
→ Section "Time zone"
→ Sélectionner le fuseau horaire correct (ex: Europe/Paris)
→ Sauvegarder

Services > NTP
→ Activer NTP
→ Saisir des serveurs NTP fiables :
→   0.fr.pool.ntp.org
→   1.fr.pool.ntp.org
→   time.cloudflare.com
→ Cliquer sur Save
```

CLI de vérification

```
date
# Doit afficher la date et heure correctes avec le bon fuseau
ntpq -p
# Lister les pairs NTP et vérifier la synchronisation (colonne "reach" > 0)
# Ou :
ntpdate -q 0.fr.pool.ntp.org
```

Remédiation

1. Aller dans `System > General Setup`, section `Localization`, configurer le fuseau horaire
2. Aller dans `Services > NTP`
3. Cocher `Enable NTP server`
4. Configurer au minimum 3 serveurs NTP de référence (idéalement un stratum 1 ou 2)
5. Si l'environnement est isolé, utiliser un serveur NTP interne synchronisé sur une source fiable (GPS, DCF77)
6. Vérifier la synchronisation via `ntpq -p`

Valeur par défaut : `0.pfsense.pool.ntp.org` comme serveur NTP par défaut. Fuseau horaire UTC.

Critère de conformité : Fuseau horaire correct configuré. Au moins 2 serveurs NTP configurés et joignables. `ntpq -p` montre un pair synchronisé avec `*` ou `o`. Écart temporel < 1 seconde.

Contrôle 5.4 — Désactiver UPnP et NAT-PMP

> ***CIS Ref :*** *Best Practice* | ***MITRE :*** *T1562.004* | ***Niveau :*** 🟠 *ÉLEVÉ*

Description du risque

UPnP (Universal Plug and Play) et NAT-PMP permettent aux applications et équipements du réseau local de créer automatiquement des règles de redirection de port sans intervention de l'administrateur. Des malwares et des applications compromises exploitent UPnP pour ouvrir des ports vers Internet, créant des backdoors persistantes contournant les politiques de sécurité.

Impact potentiel

- Création automatique de règles NAT par des malwares (C2 via UPnP)
- Exposition de services internes sur Internet sans contrôle administratif
- Contournement de toutes les politiques de pare-feu par des équipements IoT compromis
- Impossibilité d'auditer les redirections de port (changeantes dynamiquement)

Navigation

```
Services > UPnP & NAT-PMP
→ Décocher "Enable UPnP & NAT-PMP"
→ Cliquer sur Save
```

CLI de vérification

```
grep -i "upnp\|natpmp" /cf/conf/config.xml | grep "enable"
ps aux | grep miniupnp
# Ne doit retourner aucun processus miniupnpd actif
```

Remédiation

1. Aller dans `Services > UPnP & NAT-PMP`
2. Décocher `Enable UPnP & NAT-PMP`
3. Sauvegarder
4. Vérifier qu'aucune application n'a besoin d'UPnP pour fonctionner (gaming, certains VoIP) — si c'est le cas, créer des règles NAT statiques explicites à la place
5. Vérifier dans `Firewall > NAT > Port Forward` qu'aucune redirection UPnP résiduelle n'est présente

Valeur par défaut : UPnP désactivé par défaut.

Critère de conformité : UPnP et NAT-PMP désactivés. `ps aux | grep miniupnp` ne retourne aucun processus. Aucune entrée UPnP dans `Firewall > NAT > Port Forward`.

Contrôle 5.5 — Traffic Shaping : limitation de bande passante et priorisation sécurité

> ***CIS Ref :*** *Best Practice* | ***MITRE :*** *T1498, T1071* | ***Niveau :*** 🔵 *L2*

Description du risque

Sans traffic shaping configuré, toute source — y compris des hôtes internes compromis ou des flux suspects — peut utiliser toute la bande passante disponible. Un hôte faisant partie d'un botnet peut saturer la connexion Internet pour des communications C2 (Command and Control) volumineuses. Le traffic shaping permet de limiter l'impact des hôtes compromis et de garantir la priorité aux flux de sécurité et de management.

Impact potentiel

- Saturation de la bande passante par des communications C2 ou des attaques volumétriques internes
- Impossibilité d'accéder à la WebGUI pfSense lors d'un incident si la bande passante est saturée

- Exfiltration de données en volume non détectée par l'absence de limitation de flux suspects
- DoS indirect sur les flux de management et supervision

Navigation

```
# Activer le traffic shaping (HFSC ou PRIQ) :
Firewall > Traffic Shaper
→ Onglet "By Interface" > Sélectionner WAN
→ Enable HFSC Scheduler : cocher
→ Bandwidth : définir la bande passante réelle de l'interface WAN
→ Cliquer sur Save

# Créer des queues prioritaires :
Firewall > Traffic Shaper > Queues
→ Créer une queue haute priorité pour le trafic management :
→   Queue Name : qManagement
→   Priority : 7 (haute)
→   Bandwidth : 20% minimum garanti
→   Affecter : trafic SSH, WebGUI, syslog, NTP
→ Créer une queue basse priorité pour les flux suspects :
→   Queue Name : qLowPriority
→   Priority : 1 (basse)
→   Bandwidth : 10% maximum
→   Affecter : protocoles P2P, flux vers pays à risque (GeoIP pfBlockerNG)

# Rate limiting sur les connexions sortantes vers des ports suspects :
Firewall > Traffic Shaper > Limiters
→ Créer un limiter "SuspiciousOutbound" :
→   Bandwidth : 1 Mbps maximum
→   Burst : 512 Kbps
→ Appliquer dans les règles de pare-feu comme "in/out pipe"

# Prioriser les flux de sécurité :
Firewall > Rules > LAN
→ Éditer les règles vers les serveurs syslog et SIEM
→ Advanced Options > In/Out pipe : affecter à qManagement
```

CLI de vérification

```
# Vérifier l'état du traffic shaping :
pfctl -s queue
# Lister toutes les queues configurées et leurs statistiques

# Vérifier les statistiques de chaque queue (hits, drops) :
pfctl -vq
# Les drops élevés sur qLowPriority indiquent des flux suspects limités efficacement

# Vérifier les limiters actifs :
pfctl -s queue | grep "limit\|bandwidth"

# Vérifier l'utilisation en temps réel :
# Status > Queues (WebGUI) — affiche les graphiques de bande passante par queue
```

Remédiation

1. Aller dans `Firewall > Traffic Shaper > By Interface`
2. Activer le scheduler HFSC sur les interfaces WAN et LAN
3. Créer des queues prioritaires pour le trafic de management et sécurité
4. Créer des queues basse priorité pour les flux suspects ou protocoles P2P
5. Configurer des limiters pour les connexions sortantes de volume suspect
6. Appliquer les queues dans les règles de pare-feu via les options avancées `in/out pipe`
7. Monitorer les statistiques de queues pour détecter des anomalies (saturations, drops élevés)
8. Compléter avec pfBlockerNG (blocage des destinations) et Suricata (inspection DPI)

Valeur par défaut : Traffic shaping désactivé par défaut.

Critère de conformité : Traffic shaping activé avec queues différenciées (management haute priorité, flux suspects basse priorité). Limiters configurés pour les flux sortants suspects. `pfctl -s queue` retourne les queues configurées. Trafic de management (SSH, syslog) prioritaire en cas de saturation.

Contrôle 5.6 — Désactiver le proxy FTP et protéger les protocoles à ports dynamiques

CIS Ref : *Best Practice* | ***MITRE :*** *T1071, T1562.004* | ***Niveau :*** 🟡 *MOYEN*

Description du risque

Le proxy FTP de pfSense ouvre dynamiquement des ports supplémentaires pour les connexions FTP passives, contournant potentiellement les règles de pare-feu. FTP est un protocole non chiffré transmettant les identifiants et données en clair. Si le proxy FTP n'est pas nécessaire, il doit être désactivé. De même, TFTP utilise des ports dynamiques et ne doit être autorisé que si strictement nécessaire.

Impact potentiel

- Ouverture dynamique de ports pare-feu non contrôlés par le proxy FTP (contournement des ACL)
- Interception des identifiants FTP en clair par un attaquant réseau
- Surface d'attaque élargie par des ports dynamiques non documentés
- Non-conformité aux politiques de chiffrement des données en transit

Navigation

```
# Désactiver le proxy FTP (si non nécessaire) :
System > Advanced > Firewall & NAT
→ Section "Firewall & NAT"
→ Cocher "Disable FTP Proxy" (si case disponible selon la version)
→ Cliquer sur Save

# Alternative : bloquer FTP au niveau des règles de pare-feu :
Firewall > Rules > LAN
→ Créer une règle Block pour le port TCP 21 (FTP Control) vers Internet
→ Log : activer
→ Description : "Bloquer FTP non chiffré — utiliser SFTP/SCP"

# Vérifier qu'aucune règle ne laisse passer le port 21 vers Internet :
Firewall > Rules
→ Analyser chaque interface
→ Identifier les règles autorisant TCP 21 et les restreindre ou supprimer

# Si TFTP est nécessaire (déploiement PXE interne uniquement) :
Firewall > Rules > LAN
→ Autoriser UDP 69 uniquement depuis les serveurs PXE internes vers les clients TFTP
→ Bloquer UDP 69 vers Internet
```

CLI de vérification

```
# Vérifier l'état du proxy FTP :
grep -i "ftpproxy\|ftp_proxy" /cf/conf/config.xml

# Vérifier qu'aucun processus de proxy FTP n'est actif :
ps aux | grep ftp
# Ne doit retourner que des processus légitimes (pas de ftpproxy)

# Vérifier les règles autorisant FTP vers Internet :
pfctl -sr | grep "port 21\|ftp"
# Analyser chaque règle retournée

# Vérifier les ports ouverts sur pfSense lui-même :
sockstat -4 -l | grep ":21"
# Ne doit rien retourner
```

Remédiation

1. Aller dans `System > Advanced > Firewall & NAT`
2. Si l'option `Disable FTP Proxy` est présente, la cocher
3. Créer une règle de blocage explicite du port TCP 21 vers Internet dans `Firewall > Rules`
4. Documenter toute exception FTP avec justification et fenêtre temporelle
5. Remplacer FTP par SFTP (SSH File Transfer, port 22) ou SCP pour tous les transferts de fichiers
6. Pour TFTP : restreindre UDP 69 aux seuls flux internes nécessaires (PXE boot)
7. Bloquer TFTP vers Internet dans toutes les règles de sortie

Valeur par défaut : Proxy FTP actif par défaut (sur certaines versions pfSense).

Critère de conformité : Proxy FTP désactivé ou non actif. Aucune règle `Pass TCP 21` vers Internet sans justification documentée. Protocole FTP remplacé par SFTP/SCP dans tous les flux documentés. `ps aux | grep ftp` ne retourne pas de proxy actif.

Contrôle 5.7 — Détecter et bloquer le DNS-over-HTTPS (DoH)

> ***CIS Ref :*** *Best Practice* | ***MITRE :*** *T1071.004 (Application Layer Protocol: DNS), T1048.001 (Exfiltration Over Encrypted Non-C2 Protocol)* | ***Niveau :*** 🟠 *ÉLEVÉ*

Description du risque

Le DNS-over-HTTPS (DoH) est un protocole qui encapsule les requêtes DNS dans du trafic HTTPS standard (port 443). Si les hôtes internes peuvent contacter directement des résolveurs DoH publics (Cloudflare 1.1.1.1, Google 8.8.8.8, Quad9 9.9.9.9) sur le port 443, ils contournent intégralement le résolveur DNS pfSense (Unbound), y compris le filtrage pfBlockerNG DNSBL.

Vecteurs d'exploitation concrets : - Un malware utilise Cloudflare DoH (1.1.1.1:443) pour résoudre ses serveurs C2, invisible aux logs DNS pfSense - Un utilisateur configure Firefox en mode DoH natif pour contourner le filtrage DNSBL - Un ransomware utilise DoH pour résoudre ses serveurs de commande sans déclencher les alertes de réputation DNS - Des exfiltrations de données encodées en requêtes DNS passent via DoH en empruntant le port 443 autorisé

Impact potentiel

- Contournement total du filtrage DNSBL de pfBlockerNG (domaines malveillants résolus sans blocage)
- Communications C2 non détectées par les signatures Suricata DNS (T1071.004)
- Exfiltration DNS encodée via port 443 non détectée (T1048.001)
- Perte de visibilité complète sur l'activité DNS des hôtes internes
- Inefficacité du contrôle 5.1 (DNSSEC/DoT) si les clients bypassent pfSense

Navigation

```
# Méthode 1 — Bloquer les résolveurs DoH connus par adresse IP (liste noire IP):
Firewall > Aliases > Add
→ Name : DoH_Resolvers
→ Type : Network
→ Ajouter les IPs connues des résolveurs DoH :
→   1.1.1.1, 1.0.0.1        (Cloudflare)
→   8.8.8.8, 8.8.4.4        (Google)
→   9.9.9.9, 149.112.112.112 (Quad9)
→   208.67.222.222, 208.67.220.220 (OpenDNS)
→   185.228.168.9, 185.228.169.9   (CleanBrowsing)
→   76.76.2.0, 76.76.10.0          (Alternate DNS)
→ Sauvegarder

Firewall > Rules > LAN
→ Créer une règle Block :
→   Action : Block + Log
→   Protocol : TCP
→   Source : LAN net (ou alias réseau interne complet)
→   Destination : Alias DoH_Resolvers
→   Destination Port : 443
→   Description : "Bloquer DoH vers résolveurs publics connus"
→ Répéter pour UDP/443 (QUIC/HTTP3 DoH)

# Méthode 2 — Bloquer via pfBlockerNG (liste DNSBL anti-DoH) :
Firewall > pfBlockerNG > DNSBL
→ Ajouter une liste personnalisée anti-DoH :
→   URL : https://raw.githubusercontent.com/nicehash/doh-blocklist/main/doh.txt
→     (ou liste locale de domaines DoH : cloudflare-dns.com, dns.google, dns.quad9.net,
etc.)
→   Action : Deny Both
→   Frequency : Every 4 Hours

# Méthode 3 — Forcer tout DNS interne vers pfSense (proxy DNS transparent):
Firewall > NAT > Port Forward
→ Créer une règle de redirection :
→   Interface : LAN (et chaque VLAN interne)
→   Protocol : TCP/UDP
→   Destination : any (sauf pfSense lui-même)
→   Destination Port : 53
→   Redirect Target IP : pfSense LAN IP
→   Redirect Target Port : 53
→   Description : "Forcer tout DNS vers pfSense Unbound"
→ Cela capture les requêtes DNS directes des clients vers tout autre résolveur

# Vérification via règle pare-feu DNS strict :
Firewall > Rules > LAN
→ Créer en TÊTE de liste :
→   Pass : TCP/UDP depuis LAN vers pfSense IP port 53 (DNS autorisé vers pfSense)
→ Créer en SUITE :
→   Block + Log : TCP/UDP depuis LAN vers any port 53 (DNS direct interdit)
→   Description : "Bloquer DNS direct hors pfSense"
```

CLI de vérification

```
# Vérifier les règles de blocage DNS et DoH actives :
pfctl -sr | grep -E "port 53|port 443.*DoH|doh"

# Vérifier les règles NAT de redirection DNS :
pfctl -sn | grep "port 53"
# Les redirections vers pfSense IP doivent apparaître

# Tester si un client peut contacter un résolveur DoH public (doit échouer) :
# Depuis un hôte interne :
# curl -s "https://1.1.1.1/dns-query?name=example.com&type=A" (doit être bloqué)

# Vérifier que pfBlockerNG bloque les domaines DoH :
drill cloudflare-dns.com @127.0.0.1
# Doit retourner 0.0.0.0 (bloqué par DNSBL)

# Statistiques des blocages DoH (si log actif) :
clog /var/log/filter.log | grep "DoH\|1\.1\.1\.1\|8\.8\.8\.8" | tail -20
# Voir les tentatives de connexion DoH bloquées

# Vérifier qu'aucune connexion DoH n'est active depuis les hôtes internes :
pfctl -s states | grep ":443" | grep -E "1\.1\.1\.1|8\.8\.8\.8|9\.9\.9\.9"
# Doit retourner vide si le blocage est effectif
```

Remédiation

1. Créer un alias pfSense `DoH_Resolvers` avec les adresses IP des principaux résolveurs DoH publics
2. Créer une règle de blocage TCP/UDP port 443 vers cet alias sur toutes les interfaces internes
3. Configurer la redirection NAT transparente du port 53 vers pfSense Unbound pour capturer les requêtes DNS directes
4. Ajouter les domaines DoH connus dans le DNSBL pfBlockerNG (cloudflare-dns.com, dns.google, etc.)
5. Ajouter une règle de blocage port 53 vers tout autre destination que pfSense sur les interfaces LAN/VLAN
6. Si des navigateurs utilisent DoH nativement (Firefox, Chrome) : configurer les politiques navigateur pour désactiver DoH via GPO/MDM ou via le paramètre `canary.example.com` de Firefox
7. Ajouter cette liste de domaines DoH à bloquer dans Unbound :

   ```
   cloudflare-dns.com, dns.google, dns.quad9.net, doh.opendns.com,
   doh.cleanbrowsing.org, dns.nextdns.io, doh.mullvad.net
   ```

8. Activer le logging sur toutes les règles de blocage DoH pour détecter les tentatives de contournement

Valeur par défaut : Aucun blocage DoH. Les clients peuvent contacter librement les résolveurs DoH publics sur le port 443.

Critère de conformité : Règles de blocage TCP/UDP 443 vers les résolveurs DoH publics connus actives et loggées. Redirection NAT port 53 vers pfSense configurée. Domaines DoH présents dans le DNSBL pfBlockerNG. Test de connexion depuis un hôte interne vers `1.1.1.1:443/dns-query` échoue. `pfctl -s states | grep ":443" | grep "1.1.1.1"` retourne vide.

Domaine 6 — VPN (OpenVPN, IPsec, WireGuard)

Objectif : *Configurer les tunnels VPN avec des algorithmes cryptographiques modernes (AES-256-GCM, TLS 1.3, SHA-512, tls-crypt), une authentification robuste (PKI + MFA) et des protocoles à jour (TLS 1.2+, IKEv2, WireGuard), en éliminant tout algorithme obsolète.*

Contrôle 6.1 — OpenVPN : tls-crypt, AES-256-GCM, SHA-512, TLS 1.3

CIS Ref : *5.5.1 Ensure OpenVPN uses strong ciphers | 5.4.3 TLS encryption |* ***MITRE :*** *T1133 |* ***Niveau :*** 🔴 *CRITIQUE*

Description du risque

Des algorithmes cryptographiques faibles ou obsolètes dans OpenVPN (DES, 3DES, RC4, MD5, SHA1, BF-CBC, CBC modes en général) exposent les tunnels VPN à des attaques de déchiffrement passive et active.

tls-crypt vs tls-auth : `tls-auth` (HMAC authentication uniquement) est inférieur à `tls-crypt` qui fournit à la fois l'authentification ET le chiffrement du canal de contrôle TLS. Avec `tls-crypt`, le contenu des paquets du canal de contrôle est chiffré — un attaquant ne peut même pas distinguer un paquet OpenVPN d'un bruit aléatoire. Cela protège contre les attaques par déni de service ciblant le serveur OpenVPN et contre l'énumération de serveurs VPN.

AES-256-GCM (AEAD) : Contrairement à AES-256-CBC, AES-256-GCM est un mode AEAD (Authenticated Encryption with Associated Data) qui garantit simultanément la confidentialité, l'intégrité et l'authenticité des données — sans nécessiter un HMAC séparé.

Paramètres DH : Des paramètres Diffie-Hellman inférieurs à 4096 bits sont considérés comme insuffisants par les standards actuels (ANSSI recommande DH ≥ 3072 bits).

Impact potentiel

- Déchiffrement rétrospectif du trafic VPN si les clés sont compromises
- Attaques MITM sur le tunnel VPN sans tls-crypt
- DoS sur le serveur OpenVPN par envoi de paquets TLS malformés
- Non-conformité ANSSI (chiffrement symétrique < 256 bits non acceptable)
- Usurpation de serveur VPN côté client sans `--remote-cert-tls server`

Navigation

```
VPN > OpenVPN > Servers
→ Éditer le serveur OpenVPN
→ General Information :
→   TLS Configuration : cocher "Use a TLS Key"
→   TLS Key Usage Mode : "TLS Encryption and Authentication" (= tls-crypt)
→     IMPORTANT : "TLS Encryption AND Authentication" = tls-crypt
→     "TLS Authentication" uniquement = tls-auth (inférieur — à éviter)
→   Générer une nouvelle clé TLS si nécessaire
→ Cryptographic Settings :
→   Data Encryption Algorithms : AES-256-GCM uniquement (supprimer tous les autres)
→   Fallback Data Encryption : DÉSACTIVER (ne pas permettre de fallback vers algorithmes
faibles)
→   Auth digest algorithm : SHA512
→   Minimum TLS protocol : TLS 1.3 (ou TLS 1.2 si compatibilité requise)
→   DH Parameters : 4096 bits minimum
→   Enable NCP (Negotiable Crypto Parameters) : activer
→   NCP Algorithms : AES-256-GCM:AES-128-GCM (AEAD uniquement — pas de CBC)
→ Certificate Settings :
→   Certificate Depth : One (Client+Server)
→     Limite la profondeur de la chaîne à 1 — intermédiaires non acceptés
→ Compression :
→   Compression : DISABLED — NE PAS activer
→     Raison : attaques VORACLE et CRIME exploitent la compression VPN
→ Advanced Configuration :
→   Ajouter dans "Custom options" :
→     remote-cert-tls server
→     tls-version-min 1.3
→     explicit-exit-notify 1
→     compress no
→ Sauvegarder

# Côté clients VPN :
VPN > OpenVPN > Client Export
→ Vérifier que les profils clients incluent "remote-cert-tls server"
→ Vérifier que "compress no" est présent dans les profils clients
```

CLI de vérification

```
grep -E "cipher|tls-crypt|tls-auth|tls-version|auth|dh |compress|ncp-ciphers|cert-depth" /
var/etc/openvpn/server*.conf
# Doit retourner :
# cipher AES-256-GCM
# tls-crypt /var/etc/openvpn/server_tls.key (et non tls-auth)
# auth SHA512
# tls-version-min 1.3
# dh /etc/dh-parameters.4096
# compress no  (compression désactivée)
# ncp-ciphers AES-256-GCM:AES-128-GCM

# Vérifier tls-crypt actif (et non tls-auth) :
ps aux | grep openvpn | grep tls-crypt
# Doit retourner une ligne avec tls-crypt

# Vérifier l'absence d'algorithmes faibles :
grep -E "BF-|DES|RC4|MD5|SHA1\b|compress lzo|comp-lzo" /var/etc/openvpn/server*.conf
# Doit retourner vide

# Vérifier la compression désactivée :
grep "compress" /var/etc/openvpn/server*.conf
# Doit retourner uniquement : compress no (ou absent)
```

Remédiation

1. Aller dans `VPN > OpenVPN > Servers`, éditer chaque serveur
2. Sélectionner `TLS Encryption and Authentication` (tls-crypt) — supprimer les anciennes clés tls-auth
3. Générer de nouveaux paramètres DH 4096 bits dans `System > Cert Manager > DH Parameters`
4. Sélectionner `AES-256-GCM` uniquement dans Data Encryption Algorithms, désactiver le fallback
5. Configurer NCP Algorithms à `AES-256-GCM:AES-128-GCM` (AEAD uniquement)
6. Définir `Auth digest algorithm` à `SHA512`
7. Définir `Minimum TLS protocol` à `TLS 1.3`
8. Définir `Certificate Depth` à `One (Client+Server)` pour empêcher les chaînes de certificats non contrôlées
9. **Désactiver la compression** — sélectionner `Disabled` dans le champ Compression (prévention VORACLE/CRIME)
10. Ajouter `remote-cert-tls server`, `explicit-exit-notify 1`, `compress no` dans les options personnalisées
11. Régénérer les certificats clients avec clés RSA 4096 bits ou ECDSA P-384
12. Séparer les accès admin VPN et user VPN par sous-réseau et règles de pare-feu distinctes
13. Créer des règles de pare-feu sur l'interface OpenVPN pour isoler les clients VPN entre eux (sauf si inter-client justifié)
14. Planifier la rotation des clés tls-crypt et des certificats tous les 12 mois

Valeur par défaut : Chiffrement AES-256-CBC par défaut, tls-auth (pas tls-crypt), TLS 1.0 minimum, compression LZO activée dans les anciennes versions.

Critère de conformité : `cipher AES-256-GCM`, `tls-crypt` actif (pas `tls-auth`), `auth SHA512`, `tls-version-min 1.3`, `dh` avec paramètres 4096 bits. `compress no` présent. Certificate Depth = 1. NCP Algorithms AEAD uniquement. `remote-cert-tls server` présent. Aucun algorithme obsolète. `ps aux | grep openvpn | grep tls-crypt` retourne une ligne.

Contrôle 6.2 — OpenVPN : authentification RADIUS/LDAP

> ***CIS Ref :*** *5.4.1 Ensure RADIUS or LDAP are used for VPN Authentication* | ***MITRE :*** *T1133* | ***Niveau :*** 🟠 *ÉLEVÉ*

Description du risque

Une authentification VPN basée uniquement sur des certificats ou des mots de passe locaux ne permet pas la révocation centralisée immédiate lors d'un départ ou d'un incident. L'intégration au RADIUS ou LDAP permet la révocation instantanée de l'accès VPN en désactivant le compte dans l'annuaire.

Impact potentiel

- Accès VPN persistant après départ d'un collaborateur (compte non révoqué)
- Absence de politique de mots de passe uniforme pour les accès VPN
- Non-conformité aux exigences de gestion des identités (ISO 27001 A.9)

Navigation

```
System > User Manager > Authentication Servers
→ Configurer un serveur RADIUS ou LDAP (voir Contrôle 2.4)

VPN > OpenVPN > Servers
→ Éditer le serveur
→ Backend for authentication : sélectionner le serveur RADIUS ou LDAP configuré
→ Sauvegarder
```

CLI de vérification

```
grep -i "auth\|radius\|ldap" /var/etc/openvpn/server*.conf
# Vérifier la présence d'un plugin d'authentification RADIUS
```

Remédiation

1. S'assurer qu'un serveur RADIUS ou LDAP est configuré dans `System > User Manager > Authentication Servers`
2. Aller dans `VPN > OpenVPN > Servers`, éditer chaque serveur
3. Sélectionner le serveur d'authentification centralisé dans `Backend for authentication`
4. Activer `Require a certificate for authentication` pour une authentification à double facteur (certificat + mot de passe RADIUS)
5. Tester la connexion VPN avec un compte de test

Valeur par défaut : Authentification locale uniquement.

Critère de conformité : Authentification VPN configurée sur RADIUS ou LDAP centralisé. Test de connexion avec un compte révoqué dans l'annuaire doit échouer immédiatement.

Contrôle 6.3 — IPsec : IKEv2 uniquement, chiffrement fort

CIS Ref : *Best Practice* | ***MITRE :*** *T1133* | ***Niveau :*** 🔴 *CRITIQUE*

Description du risque

IKEv1 en mode agressif est vulnérable à des attaques par dictionnaire sur les identifiants PSK (Pre-Shared Key). Des vulnérabilités dans IKEv1 permettent également la collecte d'informations sur le gateway VPN sans authentification. Les algorithmes de chiffrement faibles (DES, 3DES, MD5, DH Group 1/2) dans les proposals IPsec exposent les tunnels à des attaques de déchiffrement.

Impact potentiel

- Attaque par dictionnaire sur les PSK IKEv1 mode agressif
- Fingerprinting et reconnaissance de la passerelle VPN IPsec
- Déchiffrement de tunnels utilisant des algorithmes obsolètes
- Non-conformité ANSSI (recommandations chiffrement VPN)

Navigation

```
VPN > IPsec > Tunnels
→ Éditer chaque Phase 1 (P1) :
→   Key Exchange version : IKEv2
→   Encryption Algorithm : AES-256-GCM ou AES-256
→   Hash Algorithm : SHA256 minimum (SHA512 recommandé)
→   DH Group : 14 minimum (2048 bits), recommandé 19 ou 20 (ECDH)
→   Lifetime : 28800 secondes (8 heures)
→ Éditer chaque Phase 2 (P2) :
→   Protocol : ESP
→   Encryption Algorithms : AES-256-GCM
→   Hash Algorithms : SHA256 (pour AES-GCM, l'authentification est intégrée)
→   PFS key group : DH Group 14 ou 19
→   Lifetime : 3600 secondes
→ Sauvegarder
```

CLI de vérification

```
# Vérifier les SA IPsec actifs
ipsec statusall
# Ou :
setkey -D | grep -i "des\|md5\|group 1\|group 2"
# Toute ligne retournée indique un algorithme faible
```

Remédiation

1. Aller dans `VPN > IPsec > Tunnels`

2. Éditer chaque Phase 1 : sélectionner IKEv2, AES-256-GCM, SHA-512, DH Group 19 (ECP256)
3. Supprimer tous les proposals avec DES, 3DES, MD5, DH Group 1, 2 ou 5
4. Éditer chaque Phase 2 : AES-256-GCM, Perfect Forward Secrecy activé (DH Group 19)
5. Remplacer les PSK par des certificats PKI si l'infrastructure le permet
6. Redémarrer les tunnels après modification : `VPN > IPsec > Status`, déconnecter et reconnecter

Valeur par défaut : IKEv1 ou IKEv2 selon la configuration, algorithmes multiples proposés par défaut incluant des algorithmes faibles.

Critère de conformité : IKEv2 uniquement. Phase 1 et Phase 2 avec AES-256-GCM minimum. DH Group ≥ 14. Aucun algorithme obsolète. `ipsec statusall` ne montre que des SA avec algorithmes conformes.

Contrôle 6.4 — Certificats PKI valides pour VPN

> ***CIS Ref :*** *5.4.2 Apply a Trusted Signed Certificate for VPN Portal* | ***MITRE :*** *T1552.004* | ***Niveau :*** 🟠 *ÉLEVÉ*

Description du risque

L'utilisation de certificats auto-signés ou expirés pour les portails VPN expose les utilisateurs à des attaques MITM. Un utilisateur recevant un avertissement de certificat peut être conditionné à l'ignorer, facilitant une attaque MITM ultérieure avec un faux certificat. Les paires de clés RSA < 2048 bits sont considérées comme compromises.

Impact potentiel

- Interception de credentials VPN via MITM avec faux certificat
- Vol de clés privées si les certificats ne sont pas renouvelés et révoqués
- Conditionnement des utilisateurs à ignorer les avertissements de certificat
- Non-conformité aux politiques PKI de l'organisation

Navigation

```
System > Cert. Manager > CAs
→ Créer ou importer une CA interne fiable (ou CA publique)

System > Cert. Manager > Certificates
→ Créer un certificat serveur signé par la CA
→   Common Name : nom DNS du serveur VPN
→   Key length : 4096 bits (RSA) ou EC P-384
→   Lifetime : 825 jours maximum (recommandé 365 jours)

VPN > OpenVPN > Servers
→ Éditer le serveur
→   Server certificate : sélectionner le certificat signé
→   Certificate Authority : sélectionner la CA correspondante
→ Sauvegarder
```

CLI de vérification

```
# Vérifier les certificats en cours d'utilisation
openssl x509 -in /var/etc/openvpn/server*.crt -text -noout | grep -E "Issuer|Subject|Not After|Public Key"
# Vérifier la taille de clé (minimum 2048 bits)
openssl x509 -in /var/etc/openvpn/server*.crt -text -noout | grep "Public Key Size\|RSA Public-Key"
```

Remédiation

1. Aller dans `System > Cert. Manager` et s'assurer d'avoir une CA d'entreprise valide
2. Générer ou renouveler les certificats serveur VPN avec clé RSA 4096 ou ECDSA P-384
3. Durée de validité ≤ 825 jours
4. Implémenter un processus de renouvellement 30 jours avant expiration
5. Configurer le CRL (Certificate Revocation List) pour la révocation des certificats clients

Valeur par défaut : Certificat auto-signé généré automatiquement à l'installation.

Critère de conformité : Certificats serveur VPN signés par une CA de confiance (interne ou publique). Clé RSA ≥ 2048 bits ou ECDSA P-256/P-384. Durée de validité ≤ 825 jours. Pas de certificats expirés.

Contrôle 6.5 — WireGuard VPN : déploiement sécurisé

> ***CIS Ref :*** *Best Practice* | ***MITRE :*** *T1133* | ***Niveau :*** 🔵 *L2*

Description du risque

WireGuard est un protocole VPN moderne disponible nativement sur pfSense CE (versions récentes) et pfSense Plus. Il fonctionne au niveau noyau (kernel-level), offre des performances proches de l'IPsec accéléré matériellement, et présente une surface d'attaque minimale (~4 000 lignes de code contre ~600 000 pour OpenVPN). Ces caractéristiques en font une alternative sécurisée pour des scénarios d'accès distants de confiance.

Caractéristiques de sécurité de WireGuard : - **Cryptographie moderne et fixe :** Curve25519 (ECDH), ChaCha20-Poly1305 (AEAD), BLAKE2s, SipHash24 — pas de négociation d'algorithme, donc pas d'attaque de downgrade cryptographique - **Stateless par design :** Pas de concept de session au sens traditionnel — réduit la surface d'attaque sur la gestion d'état - **Minimal codebase :** ~4 000 lignes de code vs ~600 000 pour OpenVPN, réduisant considérablement le risque de bugs et vulnérabilités

Limitations importantes : - Pas de révocation de certificat intégrée (pas de PKI) — la révocation se fait par suppression de la clé publique du pair - Pas de gestion de session traditionnelle — un pair supprimé peut continuer à envoyer des paquets jusqu'à ce que sa clé soit retirée - Audit externe limité comparé à OpenVPN (plus récent)

Impact potentiel si mal configuré

- Exposition inutile du port WireGuard (port standard 51820 facilement scannable)
- Fuite DNS si les serveurs DNS ne sont pas explicitement configurés dans le tunnel
- Accès non restreint si AllowedIPs = 0.0.0.0/0 utilisé par inadvertance pour un tunnel split

- Compromission persistante si les clés privées ne sont pas protégées

Navigation

```
# Installation du paquet WireGuard :
System > Package Manager > Available Packages
→ Rechercher "WireGuard"
→ Cliquer sur "Install"

# Création du tunnel :
VPN > WireGuard > Tunnels > "+ Add Tunnel"
→ Description : nom descriptif du tunnel (ex: "WG-Remote-Access-Prod")
→ Listen Port : choisir un port non standard (ex: 51821, 55820 — jamais 51820 par défaut)
→ Interface Keys > "Generate" pour créer la paire de clés
→   La clé privée est générée localement et ne doit JAMAIS être exportée
→   Copier la clé publique pour la communiquer aux pairs
→ Cliquer sur Save Tunnel

# Ajout de pairs (peers) :
VPN > WireGuard > [tunnel créé] > Peers > "+ Add Peer"
→ Public Key : saisir la clé publique du pair (client/remote peer)
→ Pre-Shared Key (PSK) : cliquer sur "Generate" — RECOMMANDÉ
→   Le PSK ajoute une couche de protection post-quantique
→   Stocker le PSK dans le coffre-fort de l'organisation
→ Allowed IPs : RESTREINDRE à la plage spécifique du pair
→   ❌ NE PAS utiliser 0.0.0.0/0 sauf si full tunnel justifié et documenté
→   ✅ Utiliser ex: 10.200.0.2/32 (IP assignée au pair uniquement)
→ Description : nom du pair (utilisateur ou équipement)
→ Cliquer sur Save Peer

# Configuration de l'interface WireGuard :
Interfaces > Assignments
→ Assigner l'interface WireGuard créée (ex: tun_wg0 → OPT_WG)
→ Activer l'interface, configurer l'adresse IP du serveur dans le tunnel
→   ex: 10.200.0.1/24

# Prévention des fuites DNS :
VPN > WireGuard > [interface] > Settings
→ DNS Servers : configurer explicitement les résolveurs DNS
→   Recommandé : pointer vers le DNS Resolver pfSense interne
→   Éviter de laisser le DNS vide (risque de DNS leak)

# Règles de pare-feu WireGuard :
Firewall > Rules > [Interface WireGuard OPT_WG]
→ Créer des règles restrictives autorisant uniquement les flux nécessaires
→ Bloquer tout accès non explicitement autorisé
→ Activer le logging sur toutes les règles

# Ne pas ajouter de route par défaut vers WireGuard sauf si full tunnel :
System > Routing > Gateways
→ Ne pas définir WireGuard comme gateway par défaut sauf besoin justifié
```

CLI de vérification

```
# Vérifier l'état de l'interface WireGuard
wg show
# Retourne : interface, clé publique, port d'écoute, pairs, latest handshake, transfert

# Vérifier l'état des pairs actifs :
wg show all
# Chaque pair doit avoir "latest handshake" récent si actif

# Vérifier la configuration du tunnel :
wg showconf tun_wg0
# Vérifier : ListenPort (non standard), PrivateKey (présent), Peers

# Vérifier l'absence du port 51820 par défaut :
wg show | grep "listening port"
# Doit retourner un port différent de 51820

# Vérifier les règles de pare-feu WireGuard :
pfctl -sr | grep "wg\|51820\|55820"

# Tester l'absence de fuite DNS depuis un client WireGuard :
# (Depuis le client) curl https://dnsleaktest.com/test
```

Remédiation

1. Installer le paquet WireGuard via `System > Package Manager > Available Packages`
2. Créer le tunnel avec un port d'écoute non standard (jamais 51820)
3. Générer les clés sur pfSense — la clé privée ne doit jamais quitter pfSense
4. Pour chaque pair : générer un Pre-Shared Key (PSK) pour la protection post-quantique
5. Restreindre `AllowedIPs` à la plage IP précise de chaque pair (pas de 0.0.0.0/0 sans justification)
6. Configurer explicitement les serveurs DNS dans le profil WireGuard client pour éviter les fuites DNS
7. Créer des règles de pare-feu restrictives sur l'interface WireGuard
8. Ne pas configurer de route par défaut vers WireGuard sauf full tunnel explicitement requis
9. Pour révoquer un pair : supprimer sa clé publique dans `VPN > WireGuard > Peers`
10. Surveiller les connexions via `wg show` (SSH ou console) et les logs de pare-feu

Valeur par défaut : WireGuard non installé par défaut — installation via Package Manager requise.

Critère de conformité : Port d'écoute différent de 51820. PSK configuré pour chaque pair. AllowedIPs restreint (pas de 0.0.0.0/0 sans justification documentée). DNS explicitement configuré dans les profils clients. Règles de pare-feu restrictives sur l'interface WireGuard. `wg show` confirme l'état des tunnels. Aucune clé privée exportée hors pfSense.

Domaine 7 — Paquets et extensions (pfBlockerNG, Snort/Suricata)

Objectif : *Étendre les capacités de sécurité de pfSense avec des paquets éprouvés (pfBlockerNG pour le filtrage IP/DNS/GeoIP, Suricata pour l'IDS/IPS inline avec règles Emerging Threats), tout en maintenant un inventaire minimal de paquets installés.*

Contrôle 7.1 — Configurer pfBlockerNG avec GeoIP, IP Reputation et DNSBL

CIS Ref : *Best Practice* | ***MITRE :*** *T1566, T1071.001, T1048* | ***Niveau :*** 🔵 *L2*

Description du risque

Sans filtrage des listes de réputation IP et DNS, pfSense n'effectue aucun blocage des connexions vers des infrastructures malveillantes connues (serveurs C2, domaines de phishing, adresses de distribution de malwares). pfBlockerNG intègre trois couches complémentaires :

1. Réputation IP : Blocage des adresses IP connues comme malveillantes via Spamhaus, Emerging Threats, Firehol.

2. GeoIP : Blocage du trafic entrant/sortant vers des pays présentant un risque élevé (Corée du Nord, Iran, Russie selon le contexte) via la base MaxMind GeoLite2.

3. DNSBL (DNS Black List) : Blocage au niveau DNS des domaines malveillants (phishing, malware, adware) avant même que la connexion soit établie. Complémentaire à l'inspection Suricata.

Impact potentiel

- Communications non bloquées vers des serveurs C2 connus
- Accès aux domaines de phishing depuis le réseau interne
- Distribution de malwares via des publicités malveillantes (malvertising)
- Connexions vers des pays/régions sans relation avec l'activité légitime de l'organisation

Navigation

```
# Installation :
System > Package Manager > Available Packages
→ Rechercher "pfBlockerNG-devel" (version recommandée)
→ Cliquer sur "Install"

# Configuration générale :
Firewall > pfBlockerNG > General
→ Enable pfBlockerNG : cocher
→ Keep Settings : cocher
→ Cron Interval : 4 heures (recommandé — intervalle de mise à jour des listes)
→ Sauvegarder

# Configuration IP (réputation) :
Firewall > pfBlockerNG > IP
→ Onglet "IPv4" > "+ Add"
→ Ajouter les feeds de réputation :
→   Spamhaus DROP : https://www.spamhaus.org/drop/drop.txt
→   Spamhaus EDROP : https://www.spamhaus.org/drop/edrop.txt
→   Emerging Threats C&C : disponible via ET feeds
→   Firehol Level 1 : https://raw.githubusercontent.com/firehol/blocklist-ipsets/master/
firehol_level1.netset
→   Action : Block Both (entrant et sortant)
→   Frequency : Every 4 Hours

# Configuration GeoIP :
Firewall > pfBlockerNG > IP > GeoIP
→ Configurer la licence MaxMind (gratuite pour GeoLite2)
→ Sélectionner les pays à bloquer sur le WAN entrant
→   Recommandé : bloquer les pays sans relation commerciale avec l'organisation
→ Action sur WAN inbound : Deny Inbound
→ Activer le logging des blocages GeoIP

# Configuration DNSBL :
Firewall > pfBlockerNG > DNSBL
→ Enable DNSBL : cocher
→ DNSBL Mode : Unbound (intégration avec le résolveur DNS)
→ Ajouter les listes DNSBL :
→   EasyList (publicités) : https://easylist.to/easylist/easylist.txt
→   EasyPrivacy (trackers) : https://easylist.to/easylist/easyprivacy.txt
→   Malware domains : https://mirror1.malwaredomains.com/files/justdomains
→   PhishTank : listes de phishing vérifiées
→   Abuse.ch URLhaus : domaines distribués par URLhaus
→   OISD (domaines malveillants) : https://dbl.oisd.nl/
→   Frequency : Every 4 Hours

# Mise à jour initiale forcée :
Firewall > pfBlockerNG > Update
→ Cliquer sur "Run" pour forcer la mise à jour immédiate
```

CLI de vérification

```
pkg info | grep pfblocker
# Vérifier l'installation du paquet

pfctl -t pfBlockerNG -T show | wc -l
# Nombre d'IPs bloquées dans les tables

# Vérifier les tables GeoIP chargées :
pfctl -t pfB_Top -T show | head -10

# Vérifier le DNSBL Unbound :
grep -c "" /var/unbound/pfb_dnsbl.conf 2>/dev/null
# Nombre de domaines bloqués

# Tester le blocage d'un domaine connu malveillant :
drill malware-test.eicar.org @127.0.0.1
# Doit retourner 0.0.0.0 (pfBlockerNG DNSBL block)
```

Remédiation

1. Installer `pfBlockerNG-devel` via `System > Package Manager`
2. Configurer les 3 couches : réputation IP, GeoIP, DNSBL
3. Obtenir une licence MaxMind GeoLite2 (gratuite) pour les données GeoIP
4. Configurer une fréquence de mise à jour de 4 heures pour toutes les listes
5. Démarrer en mode `Deny` après 24-48h d'observation des logs pour identifier les faux positifs
6. Créer des listes blanches (Permit Lists) pour les IPs légitimes bloquées (CDN Cloudflare, services Google, etc.)
7. Activer les alertes email pour les blocages anormaux
8. Compléter avec Suricata pour l'inspection en profondeur des flux autorisés

Valeur par défaut : pfBlockerNG non installé par défaut.

Critère de conformité : pfBlockerNG installé, activé avec réputation IP (≥3 listes), GeoIP (MaxMind), et DNSBL (≥5 listes) actifs. Mises à jour automatiques toutes les 4 heures. Tables pfBlockerNG chargées avec > 10 000 entrées. Logs de blocage visibles et transmis au SIEM.

Contrôle 7.2 — Déployer Suricata en mode IDS puis IPS inline

> ***CIS Ref :*** *Best Practice* | ***MITRE :*** *T1046, T1071.001, T1059* | ***Niveau :*** 🔵 *L2*

Description du risque

pfSense, en tant que pare-feu, applique des politiques basées sur des adresses IP, des ports et des protocoles, mais ne peut pas inspecter le contenu des paquets pour détecter des signatures d'attaques, des exploits ou des comportements malveillants. Suricata ajoute une capacité IDS/IPS (Intrusion Detection/Prevention System) basée sur des signatures (Emerging Threats, Snort VRT) et des analyses comportementales.

IDS vs IPS — migration progressive : Démarrer en mode IDS (détection uniquement) est impératif pour identifier les faux positifs avant de passer en mode IPS (blocage inline via netmap). Le mode IPS inline sur pfSense nécessite la désactivation des offloads matériels qui créent des incompatibilités avec netmap.

Positionnement de l'interface : Placer Suricata sur l'interface **LAN** (et non WAN) est recommandé pour éviter le bruit excessif généré par les scans Internet permanents. L'inspection sur LAN permet de détecter les comportements anormaux des hôtes internes.

Impact potentiel

- Exploits réussis sur des services autorisés par les règles de pare-feu (buffer overflow, SQLi, XSS)
- Absence de détection des attaques applicatives et des malwares réseau
- Reconnaissance réseau non détectée (scans Nmap, Masscan)
- Exfiltration de données non détectée par les règles de pare-feu

Navigation

```
# Prérequis pour le mode IPS inline (netmap) :
System > Advanced > Networking
→ Décocher "Hardware Checksum Offloading"
→ Décocher "Hardware TCP Segmentation Offloading"
→ Décocher "Hardware Large Receive Offloading"
→ Sauvegarder et REDÉMARRER pfSense

# Installation Suricata :
System > Package Manager > Available Packages
→ Rechercher "suricata"
→ Cliquer sur "Install"

# Configuration globale :
Services > Suricata > Global Settings
→ Emerging Threats Open Rules : cocher (sources gratuites ETOpen)
→   Update Interval : 1 DAY (mise à jour quotidienne recommandée)
→ Si licence ET Pro disponible : configurer ETpro pour des règles plus avancées
→ Remove Blocked Hosts Interval : 6 heures (délai avant déblocage automatique)
→ Cliquer sur Save

# Configuration d'interface (commencer par LAN) :
Services > Suricata > Interfaces > "+ Add"
→ Interface : LAN
→ Description : Suricata LAN IDS
→ Enable : cocher
→ IPS Mode : DÉCOCHER initialement (mode IDS pour la période de validation)
→   → Activer après 7-14 jours de validation uniquement
→ Block Offenders : DÉCOCHER initialement
→ Kill States : cocher (une fois IPS activé)
→ Onglet "Categories" :
→   Activer toutes les catégories Emerging Threats recommandées :
→     et.botcc, et.compromised, et.drop, et.dshield, et.emerging-malware
→     et.emerging-exploit, et.emerging-trojan, et.emerging-web_server
→     et.emerging-web_client
→ Onglet "Pass List" :
→   Créer une liste blanche avec les IPs de confiance :
→     Réseaux internes RFC 1918 (éviter les faux positifs intra-réseau)
→     Cloudflare CDN (103.21.244.0/22, 103.22.200.0/22, etc.)
→     Google Services si utilisés
→     IP des serveurs DNS upstream (Quad9, Cloudflare DoT)
→ Detection Engine Settings :
→   Performance Profile : Medium (ou High selon les ressources CPU)
→   Multi-threaded : activer si CPU multi-core disponible
→ Sauvegarder et démarrer l'interface

# Après validation (passage en IPS) :
Services > Suricata > Interfaces
→ Éditer l'interface LAN
→ IPS Mode : cocher (inline via netmap)
→ Block Offenders : cocher
→ Kill States : cocher
→ Sauvegarder
```

CLI de vérification

```
pkg info | grep suricata
# Vérifier l'installation

ps aux | grep suricata
# Vérifier le processus en cours d'exécution

# Vérifier les alertes en temps réel :
tail -f /var/log/suricata/suricata_LAN*.log/eve.json | python3 -m json.tool | grep -E
'"event_type"|"alert"|"signature"' | head -20

# Vérifier les règles chargées :
suricata -T -c /var/db/suricata/suricata_LAN*.conf 2>&1 | grep "rules loaded"

# Vérifier le mode IPS (netmap) :
cat /var/db/suricata/suricata_LAN*.conf | grep "runmode\|netmap"

# Vérifier les offloads désactivés :
sysctl net.inet.tcp.tso
# Doit retourner : 0 (TSO désactivé)
```

Remédiation

1. Désactiver tous les offloads matériels et redémarrer avant d'activer le mode IPS
2. Installer Suricata et configurer ETOpen avec mise à jour quotidienne
3. Démarrer en mode IDS sur l'interface LAN pendant 7 à 14 jours minimum
4. Analyser `eve.json` quotidiennement et créer des suppressions pour les faux positifs documentés
5. Créer des Pass Lists avec les IPs légitimes (CDN, services cloud, DNS upstream)
6. Activer le mode IPS (netmap) après validation, interface par interface
7. Monitorer les performances CPU/RAM : Suricata en mode IPS peut consommer 30-50% de CPU sur trafic intense
8. Compléter avec pfBlockerNG pour le blocage IP/DNS (couches complémentaires)

Valeur par défaut : Aucun IDS/IPS installé par défaut. Offloads matériels activés par défaut.

Critère de conformité : Suricata installé et démarré sur l'interface LAN minimum. Règles ETOpen à jour (< 24h). Mode IPS activé après validation. Offloads matériels désactivés si mode IPS. Pass Lists configurées. Logs eve.json transmis au SIEM.

Contrôle 7.3 — Supprimer les paquets non nécessaires

CIS Ref : *Best Practice* | ***MITRE :*** *T1190* | ***Niveau :*** 🟡 *MOYEN*

Description du risque

Chaque paquet installé sur pfSense augmente la surface d'attaque en ajoutant du code, des services réseau et des ports d'écoute supplémentaires. Des paquets obsolètes ou non maintenus peuvent introduire des vulnérabilités non corrigées. Le principe de minimalité impose de n'installer que les paquets strictement nécessaires à la mission.

Impact potentiel

- Vulnérabilités exploitables dans des paquets non mis à jour
- Services réseau supplémentaires exposant de nouveaux vecteurs d'attaque
- Complexité de la configuration difficile à auditer et à maintenir
- Impact sur les performances du firewall par des services inutiles

Navigation

```
System > Package Manager > Installed Packages
→ Examiner la liste des paquets installés
→ Pour chaque paquet : évaluer la nécessité opérationnelle
→ Cliquer sur "Remove" pour les paquets non nécessaires
→ Confirmer la suppression
```

CLI de vérification

```
pkg info | grep -v "^pfSense\|^php\|^strongswan\|^openvpn\|^unbound"
# Lister les paquets additionnels installés
pkg audit -F
# Vérifier les vulnérabilités connues dans les paquets installés
```

Remédiation

1. Aller dans `System > Package Manager > Installed Packages`
2. Inventorier tous les paquets installés et leur justification opérationnelle
3. Supprimer les paquets non utilisés (ex: paquets de test, paquets dupliqués)
4. Maintenir un registre des paquets autorisés avec justification
5. Exécuter `pkg audit -F` régulièrement pour détecter les paquets avec CVE connus

Valeur par défaut : Aucun paquet additionnel installé par défaut (installation minimale).

Critère de conformité : Seuls les paquets nécessaires à la mission sont installés. Chaque paquet est documenté avec sa justification. `pkg audit -F` ne retourne aucune vulnérabilité critique non traitée.

Domaine 8 — Interface web et accès SSH

Objectif : *Restreindre l'accès à l'interface d'administration (WebGUI et SSH) au minimum nécessaire : HTTPS uniquement avec HSTS, SSH avec clés publiques, port non standard, accès limité à un sous-réseau de management dédié. Ne jamais exposer la WebGUI sur le WAN.*

Contrôle 8.1 — Forcer HTTPS pour la WebGUI et activer HSTS

CIS Ref : *1.8 Ensure Web Management is Set to use HTTPS* | ***MITRE :*** *T1040* | ***Niveau :*** 🔴 *CRITIQUE*

Description du risque

HTTP en clair : L'accès à l'interface d'administration pfSense via HTTP expose les identifiants administrateur et la session à une interception par n'importe quel équipement sur le trajet réseau.

HSTS (HTTP Strict Transport Security) : Sans HSTS, un attaquant MITM peut downgrader une connexion HTTPS vers HTTP (SSLstrip) lors de la première connexion si l'utilisateur saisit l'URL sans `https://`. HSTS force le navigateur à toujours utiliser HTTPS pour ce domaine, même si l'URL HTTP est saisie. L'en-tête `Strict-Transport-Security: max-age=31536000; includeSubDomains` est envoyé par pfSense à chaque connexion HTTPS.

Exposition WebGUI sur WAN : Selon SOCFortress et le CIS Benchmark, la WebGUI ne doit JAMAIS être exposée directement sur l'interface WAN. Tout accès administratif doit passer par un VLAN de management dédié, un bastion host, ou un accès VPN.

Impact potentiel

- Vol d'identifiants administrateur par interception réseau
- Hijacking de session via vol de cookie de session en clair
- Attaque SSLstrip contournant HTTPS en l'absence de HSTS
- Compromission totale de la configuration du firewall depuis Internet si WAN exposé

Navigation

```
System > Advanced > Admin Access
→ Section "webConfigurator"
→ Protocol : HTTPS
→ TCP port : 443 (ou port personnalisé non standard)
→ SSL/TLS Certificate : sélectionner un certificat valide (voir Contrôle 8.4)
→ HTTP Strict Transport Security : COCHER (activer HSTS)
→ HSTS Max Age : 31536000 (1 an recommandé)
→ Cliquer sur Save

# Vérifier que WebGUI n'est pas accessible depuis WAN :
Firewall > Rules > WAN
→ S'assurer qu'aucune règle n'autorise TCP vers port 443 ou port WebGUI depuis WAN
→ Créer une règle de blocage explicite si nécessaire
```

CLI de vérification

```
grep -i "protocol\|port\|hsts" /cf/conf/config.xml | grep -A2 -B2 "webgui"
# Vérifier que "https" et "hsts" sont configurés

# Tenter un accès HTTP : doit rediriger vers HTTPS ou refuser
curl -k http://localhost/ -I 2>/dev/null | grep "Location\|HTTP/"

# Vérifier la présence de l'en-tête HSTS :
curl -k -I https://localhost/ 2>/dev/null | grep -i "strict-transport"
# Doit retourner : Strict-Transport-Security: max-age=...

# Vérifier que WebGUI n'est pas accessible depuis WAN :
pfctl -sr | grep "pass.*wan.*443\|pass.*wan.*webgui"
# Doit retourner vide
```

Remédiation

1. Aller dans `System > Advanced > Admin Access`
2. Changer `Protocol` de `HTTP` à `HTTPS`
3. Activer `HTTP Strict Transport Security` avec `max-age=31536000`
4. Sélectionner un certificat TLS valide (voir Contrôle 8.4)
5. Vérifier les règles pare-feu WAN : aucune règle ne doit autoriser l'accès à la WebGUI depuis WAN
6. Utiliser un VLAN de management dédié ou un accès VPN pour toute administration
7. Sauvegarder et tester

Valeur par défaut : HTTPS par défaut depuis pfSense 2.5.x. HSTS non activé par défaut.

Critère de conformité : WebGUI accessible uniquement en HTTPS. En-tête `Strict-Transport-Security` présent dans les réponses HTTP. Aucune règle WAN autorisant l'accès à la WebGUI. Accès WebGUI uniquement depuis VLAN management ou VPN.

Contrôle 8.2 — SSH avec clés publiques uniquement

CIS Ref : *Best Practice* | ***MITRE :*** *T1110* | ***Niveau :*** 🟠 *ÉLEVÉ*

Description du risque

L'authentification SSH par mot de passe est vulnérable aux attaques par force brute, credential stuffing et phishing. L'utilisation de paires de clés SSH (clé privée sur le poste de l'administrateur, clé publique sur pfSense) élimine ces vecteurs d'attaque car la clé privée ne transite jamais sur le réseau.

Impact potentiel

- Compromission par force brute sur SSH si des mots de passe faibles sont utilisés
- Credential stuffing depuis des fuites de bases de données
- Compromission de l'accès SSH par phishing du mot de passe administrateur

Navigation

```
System > Advanced > Admin Access
→ Section "Secure Shell"
→ Cocher "Enable Secure Shell"
→ Authentication Method : "Public Key Only"
→ SSH Port : saisir un port non standard (ex: 2222)
→ Sauvegarder

System > User Manager
→ Éditer le compte administrateur
→ Authorized SSH Keys : coller la clé publique SSH de l'administrateur
→ Format : ssh-ed25519 AAAA... utilisateur@poste
→ Sauvegarder
```

CLI de vérification

```
grep "^PasswordAuthentication\|^PubkeyAuthentication\|^Port" /etc/ssh/sshd_config
# Doit retourner :
# PasswordAuthentication no
# PubkeyAuthentication yes
# Port <port_non_standard>
```

Remédiation

1. Générer une paire de clés SSH sur le poste de l'administrateur :

   ```
   ssh-keygen -t ed25519 -C "admin@organisation" -f ~/.ssh/pfsense_admin
   ```

2. Copier la clé publique (`~/.ssh/pfsense_admin.pub`) dans `System > User Manager > Authorized SSH Keys`
3. Aller dans `System > Advanced > Admin Access > Secure Shell`
4. Sélectionner `Public Key Only` pour l'authentification

5. Tester la connexion par clé avant de désactiver l'authentification par mot de passe
6. Restreindre SSH aux adresses IP du réseau de management via les règles de pare-feu

Valeur par défaut : SSH désactivé par défaut. Si activé, authentification par mot de passe autorisée.

Critère de conformité : `PasswordAuthentication no` dans `/etc/ssh/sshd_config`. Clé publique SSH configurée pour chaque administrateur. SSH accessible uniquement depuis le réseau de management.

Contrôle 8.3 — SSH sur port non standard, accès restreint par IP

CIS Ref : *Best Practice* | ***MITRE :*** *T1190* | ***Niveau :*** 🟡 *MOYEN*

Description du risque

Le port SSH par défaut (22) est constamment sondé par des botnets et des scanners automatiques qui cherchent des services SSH à attaquer. L'utilisation d'un port non standard réduit drastiquement le bruit des tentatives automatisées et les inscrit dans les logs uniquement si le scanner est sophistiqué. La restriction par IP ajoute une couche de protection supplémentaire.

Impact potentiel

- Tentatives de brute force automatisées constantes sur le port 22
- Saturation des logs par des tentatives automatisées
- Exposition à des exploits 0-day sur OpenSSH si le port est publiquement visible

Navigation

```
System > Advanced > Admin Access
→ Section "Secure Shell"
→ SSH Port : saisir un port entre 1024 et 65535 (ex: 2222, 22022)
→ Sauvegarder

Firewall > Rules (interface de management)
→ Créer une règle :
→   Action : Pass
→   Protocol : TCP
→   Source : adresse IP ou réseau des administrateurs
→   Destination : This Firewall
→   Destination Port : port SSH configuré
→   Log : activé
→ Créer une règle de blocage pour toutes les autres sources sur ce port
```

CLI de vérification

```
grep "^Port" /etc/ssh/sshd_config
# Doit retourner le port non standard configuré
pfctl -sr | grep "port 22\b"
# Doit retourner une règle de blocage du port 22
```

Remédiation

1. Modifier le port SSH dans `System > Advanced > Admin Access`
2. Mettre à jour les configurations SSH des postes d'administration (`~/.ssh/config`)
3. Créer une règle de pare-feu autorisant SSH uniquement depuis le subnet de management
4. Bloquer toute connexion SSH depuis le WAN et tout réseau non autorisé
5. Documenter le nouveau port SSH dans le runbook d'administration

Valeur par défaut : Port SSH 22 par défaut.

Critère de conformité : Port SSH différent de 22. `grep "^Port" /etc/ssh/sshd_config` retourne un port non standard. Règle de pare-feu autorisant SSH uniquement depuis les adresses de management.

Contrôle 8.4 — Certificat TLS valide pour la WebGUI

> ***CIS Ref :*** *Best Practice* | ***MITRE :*** *T1040* | ***Niveau :*** 🟠 *ÉLEVÉ*

Description du risque

Le certificat auto-signé généré par pfSense à l'installation n'est pas approuvé par les navigateurs, forçant les administrateurs à ignorer les avertissements TLS. Cette habitude est dangereuse car elle conditionne les administrateurs à ignorer les alertes de certificat, facilitant les attaques MITM avec un faux certificat. Un certificat signé par une CA de confiance élimine ces avertissements.

Impact potentiel

- Conditionnement des administrateurs à ignorer les alertes de certificat
- MITM possible avec un faux certificat sans détection visuelle
- Impossibilité d'utiliser HSTS efficacement sans certificat de confiance
- Non-conformité aux politiques PKI de l'organisation

Navigation

```
System > Cert. Manager > CAs
→ Importer ou créer la CA interne de l'organisation

System > Cert. Manager > Certificates
→ Créer un certificat serveur signé par la CA interne :
→   Certificate Type : Server Certificate
→   Common Name : nom DNS de l'interface (ex: pfsense.corp.exemple.fr)
→   Alternative Names (SAN) : ajouter tous les noms DNS d'accès
→   Key length : 4096 bits (RSA) ou EC P-384
→ Sauvegarder

System > Advanced > Admin Access
→ SSL/TLS Certificate : sélectionner le nouveau certificat
→ Sauvegarder
```

CLI de vérification

```
openssl s_client -connect localhost:443 </dev/null 2>/dev/null | openssl x509 -text -noout |
grep -E "Subject:|Issuer:|Not After|Subject Alternative"
# Vérifier que le certificat est signé par la CA interne et non auto-signé
# Vérifier la date d'expiration
```

Remédiation

1. Créer ou importer la CA interne dans `System > Cert. Manager > CAs`
2. Générer un certificat serveur avec les bons SAN (Subject Alternative Names)
3. Sélectionner ce certificat dans `System > Advanced > Admin Access`
4. Distribuer le certificat CA aux navigateurs des administrateurs (via GPO ou MDM)
5. Implémenter un processus de renouvellement 30 jours avant expiration (monitoring via `System > Cert. Manager`)

Valeur par défaut : Certificat auto-signé généré à l'installation.

Critère de conformité : Certificat WebGUI signé par une CA approuvée par les navigateurs des administrateurs. Pas d'avertissement de certificat lors de l'accès à la WebGUI. Certificat valide (non expiré). Clé RSA ≥ 2048 bits.

Contrôle 8.5 — Restreindre l'accès management à une interface dédiée

> ***CIS Ref :*** *Best Practice* | ***MITRE :*** *T1190* | ***Niveau :*** 🔵 *L2*

Description du risque

Si la WebGUI et SSH de pfSense sont accessibles depuis le WAN ou depuis tout le réseau LAN, la surface d'attaque est maximale. Une interface de management dédiée (MGMT VLAN, OOB network) permet de restreindre l'accès aux seuls administrateurs légitimes depuis un réseau de management contrôlé et supervisé. SOCFortress et le CIS Benchmark insistent : "Never expose the WebGUI to the WAN" — cette règle est absolue.

Impact potentiel

- Interface d'administration exposée à des attaques depuis Internet (WAN)
- Accès non autorisé à la WebGUI depuis n'importe quel poste du LAN
- Impossibilité de distinguer le trafic de management du trafic utilisateur

Navigation

```
Interfaces > [OPT_MGMT]
→ Configurer une interface de management dédiée (VLAN ou physique)
→ Attribuer une plage IP dédiée (ex: 192.168.99.0/30)

System > Advanced > Admin Access
→ WebGUI : restreindre l'écoute à l'interface MGMT uniquement
→ (Ou utiliser les règles de pare-feu pour restreindre l'accès)

Firewall > Rules [WAN]
→ S'assurer qu'aucune règle n'autorise l'accès à la WebGUI ou SSH depuis le WAN
→ Bloquer explicitement l'accès aux ports 443 et SSH depuis le WAN
```

CLI de vérification

```
pfctl -sr | grep -E "pass.*to.*443|pass.*to.*port 22\b|pass.*to.*port 2222"
# Vérifier que seules les règles depuis le réseau de management sont présentes
netstat -an | grep "LISTEN" | grep -E ":443|:22"
# Vérifier les interfaces d'écoute
```

Remédiation

1. Créer un VLAN ou une interface physique dédiée à la gestion réseau
2. Configurer les règles de pare-feu pour autoriser WebGUI et SSH uniquement depuis ce réseau
3. Bloquer explicitement tout accès WebGUI/SSH depuis le WAN et les autres réseaux
4. Documenter le réseau de management dans la cartographie réseau
5. Superviser les tentatives d'accès à la WebGUI depuis des réseaux non autorisés

Valeur par défaut : WebGUI et SSH accessibles depuis toutes les interfaces par défaut.

Critère de conformité : Aucune règle autorisant l'accès WebGUI ou SSH depuis le WAN. Accès WebGUI et SSH restreints au réseau de management. Test depuis un poste hors-réseau de management doit échouer.

Contrôle 8.6 — HAProxy reverse proxy : durcissement TLS et en-têtes de sécurité

CIS Ref : *Best Practice* | ***MITRE :*** *T1190, T1557* | ***Niveau :*** 🔵 *L2*

Description du risque

HAProxy est fréquemment utilisé avec pfSense comme reverse proxy et load balancer sécurisé pour exposer des applications internes sur Internet. Une configuration TLS faible (TLS 1.0/1.1, chiffres faibles, DH 2048 bits) expose les communications à des attaques de déchiffrement et de downgrade. L'absence d'en-têtes de sécurité HTTP (HSTS, CSP, X-Frame-Options) facilite les attaques de type Clickjacking, XSS et MITM contre les utilisateurs des applications publiées.

Objectif : Atteindre la note A+ sur SSLLabs (https://www.ssllabs.com/ssltest/) pour toutes les applications publiées via HAProxy.

Impact potentiel

- Déchiffrement du trafic HTTPS par attaques POODLE (TLS 1.0), BEAST, CRIME si protocoles faibles activés
- Attaques MITM sur les applications publiées si le chiffrement est insuffisant
- Clickjacking sur les applications sans `X-Frame-Options`
- Vol de session si HSTS absent (SSLstrip possible)
- Exploitation d'applications web via XSS sans `Content-Security-Policy`
- Exposition à des attaques T1190 (Exploit Public-Facing Application) via serveurs backend non chiffrés

Navigation

```
# Installation de HAProxy :
System > Package Manager > Available Packages
→ Rechercher "haproxy"
→ Cliquer sur "Install"

# Configuration générale :
Services > HAProxy > Settings
→ Enable HAProxy : cocher
→ Maximum connections : adapter selon les ressources (ex: 1000)
→ Sauvegarder

# Configuration Frontend (écoute HTTPS) :
Services > HAProxy > Frontend > "+ Add"
→ Name : exemple-https-frontend
→ External address : IP WAN, port 443
→ SSL Offloading : cocher "Use offloading"
→ Certificate : sélectionner le certificat SSL de l'application (Let's Encrypt ou PKI
interne)
→ Additional certificate(s) : ajouter si nécessaire (multi-domaines)
→ SSL Options :
→   Cipher list (TLS 1.2) :
→     ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-RSA-AES128-GCM-SHA256:ECDHE-ECDSA-AES256-GCM-
SHA384:ECDHE-RSA-AES256-GCM-SHA384:ECDHE-ECDSA-CHACHA20-POLY1305:ECDHE-RSA-CHACHA20-POLY1305
→   Cipher suites (TLS 1.3) :
→     TLS_AES_128_GCM_SHA256:TLS_AES_256_GCM_SHA384:TLS_CHACHA20_POLY1305_SHA256
→   Minimum SSL version : TLSv1.2 (désactive TLS 1.0 et TLS 1.1)
→   HSTS : cocher "HTTP Strict Transport Security"
→     HSTS Max Age : 31536000
→     HSTS Subdomains : cocher "includeSubDomains"
→     HSTS Preload : cocher si soumission au registre HSTS preload

# Paramètres Diffie-Hellman 4096 bits :
Services > HAProxy > Settings
→ Max SSL Diffie-Hellman size : 4096
→   (Change le DH par défaut de 2048 à 4096 — impact SSLLabs Key Exchange 100%)
→ Sauvegarder

# En-têtes de sécurité HTTP via actions HAProxy :
Services > HAProxy > Frontend > [frontend] > Actions
→ Action type : "http-response set-header"
→ Ajouter les en-têtes suivants (une action par en-tête) :
→   Strict-Transport-Security: max-age=31536000; includeSubDomains; preload
→   X-Frame-Options: SAMEORIGIN
→   X-Content-Type-Options: nosniff
→   X-XSS-Protection: 1; mode=block
→   Referrer-Policy: strict-origin-when-cross-origin
→   Permissions-Policy: geolocation=(), microphone=(), camera=()
→   Content-Security-Policy: default-src 'self'; script-src 'self'; style-src 'self'
→   (Adapter la CSP selon les besoins applicatifs — trop restrictive peut casser
l'application)

# Configuration Backend avec chiffrement de bout en bout :
Services > HAProxy > Backend > [backend]
→ Server list > Chaque serveur backend :
→   Encrypt(SSL) : cocher (active HTTPS vers le backend)
→   SSL Port : 443
```

```
→   Verify SSL Certificate : cocher si le backend a un certificat valide
→   Ca Certificate : sélectionner la CA signant le certificat backend
→ Mode : http
→ Balance algorithm : roundrobin (ou leastconn pour les sessions longues)
```

CLI de vérification

```
# Vérifier que HAProxy est installé et actif :
pkg info | grep haproxy
ps aux | grep haproxy

# Vérifier la configuration TLS active :
echo | openssl s_client -connect <IP_WAN>:443 2>/dev/null | openssl x509 -noout -text | grep
-E "Protocol|Cipher|Not After"

# Tester le protocole minimum (TLS 1.0 et 1.1 doivent être refusés) :
openssl s_client -connect <IP_WAN>:443 -tls1 2>&1 | grep -E "alert|handshake failure"
# Doit retourner "alert handshake failure" (TLS 1.0 refusé)
openssl s_client -connect <IP_WAN>:443 -tls1_1 2>&1 | grep -E "alert|handshake failure"
# Doit retourner "alert handshake failure" (TLS 1.1 refusé)
openssl s_client -connect <IP_WAN>:443 -tls1_2 2>&1 | grep "Cipher\|Protocol"
# Doit réussir avec TLS 1.2

# Vérifier les en-têtes de sécurité :
curl -sk -I https://<FQDN_APPLICATION>/ | grep -E "Strict-Transport|X-Frame|X-Content|X-XSS|
Content-Security|Referrer|Permissions"
# Tous les en-têtes de sécurité doivent être présents

# Vérifier le DH 4096 bits :
openssl s_client -connect <IP_WAN>:443 2>/dev/null | grep "Server Temp Key"
# Doit retourner : Server Temp Key: X25519, 253 bits (ou ECDH 256+)
# Ou : Server Temp Key: DH, 4096 bits si DHE utilisé

# Vérifier la validité du certificat et la chaîne :
echo | openssl s_client -connect <FQDN_APPLICATION>:443 -verify_hostname <FQDN_APPLICATION>
2>&1 | grep "Verify return code"
# Doit retourner : Verify return code: 0 (ok)
```

Remédiation

1. Installer le paquet `haproxy` via `System > Package Manager > Available Packages`
2. Configurer le frontend HTTPS avec TLS 1.2 minimum — désactiver TLS 1.0 et TLS 1.1 explicitement
3. Configurer les cipher suites en utilisant uniquement les algorithmes ECDHE avec AEAD (GCM, ChaCha20-Poly1305)
4. Passer le `Max SSL Diffie-Hellman size` à 4096 bits dans `Services > HAProxy > Settings`
5. Activer HSTS avec `max-age=31536000; includeSubDomains; preload`
6. Ajouter les 7 en-têtes de sécurité HTTP via les actions HAProxy dans le frontend
7. Activer le chiffrement de bout en bout (SSL backend) pour éviter le déchiffrement en DMZ
8. Tester la note SSLLabs via https://www.ssllabs.com/ssltest/ — objectif A+
9. Configurer un enregistrement DNS CAA pour restreindre l'émission de certificats à l'AC autorisée :

```
exemple.fr.  CAA  0 issue "letsencrypt.org"
exemple.fr.  CAA  0 issuewild ";"
```

10. Tester les en-têtes de sécurité via https://securityheaders.com — objectif A+
11. Adapter la Content-Security-Policy à chaque application pour éviter les ruptures de fonctionnalité

Valeur par défaut : HAProxy non installé par défaut. Si installé, TLS 1.0/1.1 acceptés, DH 2048 bits, aucun en-tête de sécurité configuré.

Critère de conformité : Note SSLLabs A+ sur toutes les applications publiées via HAProxy. TLS 1.0 et TLS 1.1 refusés (`openssl s_client -tls1` retourne une erreur). DH 4096 bits configuré. Les 7 en-têtes de sécurité présents dans les réponses HTTP. Chiffrement bout-en-bout activé vers les backends. HSTS avec `preload` activé.

Contrôle 8.7 — Automatisation des certificats TLS via ACME / Let's Encrypt

> ***CIS Ref :*** *Best Practice* | ***MITRE :*** *T1553 (Subvert Trust Controls — certificat expiré), T1190 (Exploit Public-Facing Application — certificat faible)* | ***Niveau :*** 🟠 *ÉLEVÉ*

Description du risque

La gestion manuelle des certificats TLS est une source fréquente d'incidents de sécurité : certificats oubliés et expirés, certificats auto-signés mal gérés, renouvellements tardifs causant des interruptions de service. Un certificat expiré force les administrateurs à ignorer les alertes de sécurité, ouvrant la voie aux attaques MITM. Le paquet ACME de pfSense automatise entièrement le cycle de vie des certificats via le protocole ACME (RFC 8555) avec les autorités de certification Let's Encrypt, ZeroSSL et Buypass — sans frais.

Vecteurs liés aux certificats mal gérés : - Certificats auto-signés : les administrateurs s'habituent à ignorer les alertes de sécurité navigateur (T1553) - Certificats expirés : indisponibilité de service, bypass des contrôles de sécurité TLS - Certificats avec clés faibles (RSA 1024) : déchiffrement rétrospectif possible - Absence de rotation : une clé privée compromise reste valide indéfiniment

Méthodes de validation ACME disponibles :

MÉTHODE	FONCTIONNEMENT	USAGE RECOMMANDÉ
standalone	pfSense écoute temporairement sur port 80/443	Accès direct depuis Internet sur ce port
DNS-01	pfSense modifie un enregistrement DNS TXT via API	**Recommandé** — fonctionne sans port ouvert, obligatoire pour les wildcards
webroot	pfSense écrit un fichier dans le webroot d'un serveur web existant	Serveur web déjà en place sur pfSense

Autorités de certification supportées : - **Let’s Encrypt (production)** : certificats publiquement reconnus, renouvellement automatique, gratuits - **Let’s Encrypt (staging)** : pour les tests — ne pas utiliser en production - **ZeroSSL** : alternative à Let’s Encrypt, certificats wildcards gratuits - **Buypass Go SSL** : CA européenne (Norvège), conforme RGPD

Impact potentiel

- Certificats expirés forçant les utilisateurs/administrateurs à contourner les alertes TLS
- Indisponibilité de service lors de l’expiration d’un certificat non renouvelé
- Exposition MITM si les alertes de certificat sont régulièrement ignorées
- Non-conformité aux politiques PKI (certificats > 398 jours refusés depuis 2020)

Navigation

```
# Étape 1 — Installation du paquet ACME :
System > Package Manager > Available Packages
→ Rechercher "acme"
→ Cliquer sur "Install" (paquet acme)

# Étape 2 — Création du compte ACME (Let's Encrypt) :
Services > ACME Certificates > Account Keys
→ Cliquer sur "+ Add"
→ Name : lets-encrypt-prod (ou nom descriptif)
→ ACME Server : Let's Encrypt Production ACME v2
→   (Pour tests : Let's Encrypt Staging ACME v2)
→ Account key : cliquer sur "Create new account key"
→ Register ACME account key : cliquer sur "Register"
→ Sauvegarder

# Étape 3 — Création d'un certificat :
Services > ACME Certificates > Certificates
→ Cliquer sur "+ Add"
→ Name : pfsense-webgui-cert (nom descriptif)
→ Status : Active
→ Acme Account : sélectionner le compte créé à l'étape 2
→ Private Key : RSA 4096 (recommandé) ou ECDSA P-384
→ Domain SAN list :
→   Mode : Enabled
→   Domainname : pfsense.corp.exemple.fr
→     (Ajouter tous les noms DNS d'accès à la WebGUI comme SAN)
→   Method : DNS-01 (recommandé pour les domaines internes)
→     → DNS-01 requiert une API DNS (Cloudflare, OVH, Gandi, AWS Route53, etc.)
→     → Configurer les credentials API dans la section "Domain SAN list"
→   Ou Method : Standalone (si pfSense est accessible depuis Internet sur port 80)
→     → Attention : ouvre temporairement le port 80 lors des renouvellements

# Étape 4 — Actions post-renouvellement :
→ Section "Actions list" → Cliquer sur "+ Add action"
→ Ajouter les services à redémarrer après renouvellement :
→   HAProxy :
→     Command : /usr/local/etc/rc.d/haproxy.sh restart
→   WebGUI (nginx) :
→     Command : /usr/local/sbin/pfSsh.php playback svc restart nginx
→   OpenVPN (si certificat utilisé pour OpenVPN) :
→     Command : /usr/local/sbin/pfSsh.php playback svc restart openvpn
→ Ces actions garantissent le rechargement automatique des certificats renouvelés

→ Cliquer sur "Issue/Renew" pour émettre le certificat manuellement (premier démarrage)
→ Vérifier l'émission réussie dans les logs : Services > ACME Certificates > Certificates >
Logs

# Étape 5 — Appliquer le certificat ACME à la WebGUI :
System > Advanced > Admin Access
→ SSL/TLS Certificate : sélectionner le certificat ACME émis
→ Sauvegarder

# Renouvellement automatique :
→ Le paquet ACME configure automatiquement une tâche cron pour renouveler
→   30 jours avant l'expiration (certificats Let's Encrypt valables 90 jours)
→ Vérifier la tâche cron : Status > Cron Jobs (si paquet Cron installé)
```

```
→ Ou via CLI : crontab -l | grep acme

# Wildcards (certificats *.exemple.fr) :
→ Uniquement via DNS-01 — impossible avec standalone ou webroot
→ Permet de couvrir tous les sous-domaines avec un seul certificat
→ Attention : certificat wildcard = valeur plus haute pour un attaquant
→   Stocker la clé privée avec soin, ne pas partager
```

CLI de vérification

```
# Vérifier l'installation du paquet ACME :
pkg info | grep acme

# Lister les certificats ACME émis et leurs dates d'expiration :
ls -la /var/etc/acme/
# Chaque domaine a un sous-répertoire avec les fichiers .cer, .key, .csr

# Vérifier la date d'expiration d'un certificat ACME :
openssl x509 -in /var/etc/acme/pfsense.corp.exemple.fr/pfsense.corp.exemple.fr.cer \
  -noout -dates
# notBefore + notAfter — alerter si notAfter < 30 jours

# Vérifier la chaîne de certificat complète :
openssl verify -CAfile /var/etc/acme/pfsense.corp.exemple.fr/ca.cer \
  /var/etc/acme/pfsense.corp.exemple.fr/pfsense.corp.exemple.fr.cer
# Doit retourner : OK

# Déclencher un renouvellement manuel forcé (test ou urgence) :
/usr/local/pkg/acme/acme.sh --renew -d pfsense.corp.exemple.fr --force \
  --config-home /var/etc/acme
# Surveiller la sortie pour les erreurs

# Vérifier la tâche cron de renouvellement automatique :
crontab -l | grep acme
# Doit retourner une ligne de renouvellement planifiée

# Vérifier le certificat actif sur la WebGUI pfSense :
echo | openssl s_client -connect localhost:443 2>/dev/null | \
  openssl x509 -noout -subject -issuer -dates
# Issuer doit contenir "Let's Encrypt" ou le CA ACME configuré

# Vérifier le renouvellement dans les logs système :
grep -i "acme\|letsencrypt\|renewal" /var/log/system.log | tail -20
```

Remédiation

1. Installer le paquet `acme` via `System > Package Manager > Available Packages`
2. Créer un compte ACME avec Let's Encrypt Production (pas staging en production)
3. Choisir la méthode de validation adaptée :
   - **DNS-01** recommandé pour les domaines internes — configurer les credentials API DNS
   - **Standalone** uniquement si pfSense est directement accessible sur le port 80 depuis Internet
4. Configurer les actions post-renouvellement pour redémarrer HAProxy, nginx (WebGUI) et OpenVPN
5. Émettre le premier certificat manuellement et vérifier dans les logs

6. Appliquer le certificat ACME dans `System > Advanced > Admin Access > SSL/TLS Certificate`
7. Vérifier la tâche cron de renouvellement automatique
8. Alerter sur les certificats proches de l'expiration (< 30 jours) : créer une règle dans le SIEM sur les logs ACME
9. Pour les wildcards : utiliser exclusivement DNS-01 et stocker la clé privée avec le même niveau de protection que les secrets VPN
10. Documenter les credentials API DNS dans le coffre-fort de l'organisation

Valeur par défaut : Paquet ACME non installé par défaut. Certificat auto-signé généré à l'installation.

Critère de conformité : Paquet ACME installé. Certificat WebGUI émis par une CA publique reconnue (Let's Encrypt, ZeroSSL). Renouvellement automatique configuré et fonctionnel (tâche cron présente). Actions post-renouvellement configurées pour tous les services utilisant le certificat. Date d'expiration > 30 jours. `openssl x509 -in /var/etc/acme/[domaine]/*.cer -noout -dates` confirme la validité.

Contrôle 8.8 — Durcissement de l'API REST pfSense

> ***CIS Ref :*** *Best Practice* | ***MITRE :*** *T1190 (Exploit Public-Facing Application), T1078 (Valid Accounts — tokens API), T1552.001 (Unsecured Credentials — API keys)* | ***Niveau :*** 🔵 *L2*

Description du risque

L'API REST pfSense (paquet tiers `pfSense-pkg-API` ou `fauxapi`) permet l'automatisation des opérations d'administration via des requêtes HTTP. Une API mal configurée constitue un vecteur d'attaque supplémentaire : authentification faible, endpoint exposé sans restriction IP, absence de rate limiting, ou API activée sans utilisation réelle. Si l'API n'est pas nécessaire, elle doit être désinstallée.

Risques spécifiques de l'API REST : - **Authentication faible :** Basic Auth transmis en HTTP expose les identifiants en clair - **Tokens non révoqués :** Les tokens API JWT peuvent persister après départ d'un administrateur - **Absence de restriction IP :** L'endpoint API accessible depuis tout le réseau interne ou Internet - **Absence de rate limiting :** Brute force sur l'API non bloqué par Login Protection standard - **Logs insuffisants :** Les appels API non loggés échappent à la détection

Impact potentiel

- Accès administrateur complet à pfSense via API si les tokens sont compromis
- Modification des règles de pare-feu via API sans authentification forte
- Brute force sur l'endpoint API contournant le Login Protection WebGUI
- Exfiltration de la configuration complète (clés VPN, certificats) via l'API

Navigation

```
# Si l'API REST n'est PAS nécessaire — désinstaller :
System > Package Manager > Installed Packages
→ Identifier le paquet API (pfSense-pkg-API, fauxapi, etc.)
→ Cliquer sur "Remove"
→ Confirmer la désinstallation

# Si l'API REST EST nécessaire — durcir :

# 1. Configuration de l'authentification JWT (pas Basic Auth) :
System > Package Manager > Available Packages
→ Installer "pfSense-pkg-API" si non installé

System > API (selon la version du paquet)
→ Authentication mode : JWT (JSON Web Tokens)
→   NE PAS utiliser Basic Authentication (identifiants en clair)
→ Token expiry : définir une durée courte (ex: 3600 secondes = 1 heure)
→   Les tokens doivent expirer régulièrement et nécessiter un renouvellement
→ Allowed IPs : restreindre aux seules adresses IP autorisées à utiliser l'API
→   Ex: 192.168.99.10/32 (poste de gestion automation uniquement)
→ Sauvegarder

# 2. Forcer HTTPS uniquement pour l'endpoint API :
→ Vérifier que l'API écoute uniquement sur HTTPS (port 443 ou port personnalisé)
→ L'API ne doit JAMAIS être accessible via HTTP en clair

# 3. Rate limiting sur l'endpoint API :
System > API > Rate Limiting (si disponible)
→ Configurer une limite de requêtes (ex: 100 req/min par IP)
→ Configurer un blocage automatique après X erreurs d'authentification

# 4. Restriction via règles pare-feu :
Firewall > Rules > [interface management]
→ Créer une règle autorisant TCP vers port API uniquement depuis l'alias IPs autorisées
→ Bloquer tout autre accès à l'endpoint API
→ Log : activer sur toutes les règles API

# 5. Monitoring des logs API :
System > System Logs > API (selon la version)
→ Vérifier la présence de logs d'accès API
→ Envoyer les logs API vers le SIEM avec règles d'alerte spécifiques
```

CLI de vérification

```
# Vérifier si un paquet API est installé :
pkg info | grep -i "api\|fauxapi"
# Si retourne une ligne, l'API est installée — vérifier la configuration

# Vérifier les ports d'écoute de l'API :
sockstat -4 -l | grep -E ":8888|:4444|:443" | grep -v nginx
# Identifier le port d'écoute de l'API REST

# Vérifier qu'aucune connexion HTTP (pas HTTPS) n'est acceptée par l'API :
curl -sk http://localhost:<port_API>/api/v1/ 2>&1 | grep -i "redirect\|https\|forbidden"
# Doit retourner une redirection HTTPS ou un refus de connexion HTTP

# Vérifier la restriction par IP (depuis une IP non autorisée, doit être refusé) :
pfctl -sr | grep "port <port_API>"
# Doit montrer des règles restrictives

# Vérifier les logs d'accès API :
grep -i "api\|jwt\|token" /var/log/system.log | tail -20
# Les appels API doivent apparaître dans les logs

# Vérifier l'absence de Basic Auth en clair :
grep -r "basic_auth\|BasicAuth" /usr/local/pkg/pfSense_pkg_API/ 2>/dev/null | grep -i
"enable\|true"
# Ne doit pas retourner de configuration Basic Auth activée

# Alternative : utiliser SSH avec clés pour l'automation plutôt que l'API REST :
# Cette approche est plus sûre et utilise les contrôles SSH déjà en place (Contrôle 8.2/8.3)
ssh -i ~/.ssh/pfsense_automation admin@pfsense.mgmt pfSsh.php playback svc list
```

Remédiation

1. **Si l'API REST n'est pas nécessaire :** Désinstaller le paquet via `System > Package Manager > Installed Packages`
2. **Si l'API REST est nécessaire :**
    - Configurer l'authentification JWT uniquement (supprimer le Basic Auth)
    - Définir une durée de vie courte pour les tokens (≤ 3600 secondes)
    - Restreindre l'accès par IP aux seules adresses d'automation autorisées
    - Forcer HTTPS uniquement sur l'endpoint API
    - Configurer un rate limiting pour prévenir le brute force
    - Envoyer tous les logs d'accès API vers le SIEM
3. **Alternative recommandée :** Utiliser SSH avec authentification par clé publique (Contrôle 8.2) pour les scripts d'automation — évite complètement l'exposition d'une API REST
4. Révoquer et régénérer tous les tokens API lors de tout changement d'équipe administrative
5. Documenter la liste des systèmes autorisés à utiliser l'API REST avec justification opérationnelle
6. Auditer les logs API mensuellement pour détecter des accès non autorisés

Valeur par défaut : Aucune API REST installée par défaut. Installation explicite requise via Package Manager.

Critère de conformité : Si API REST non nécessaire : paquet absent (`pkg info | grep -i api` retourne vide). Si API REST nécessaire : authentification JWT uniquement, restriction IP configurée, HTTPS obligatoire, logs transmis au SIEM. Aucun token Basic Auth actif. Rate limiting configuré.

Domaine 9 — Haute disponibilité et sauvegardes (CARP)

Objectif : *Assurer la résilience opérationnelle du firewall via CARP/pfsync pour la haute disponibilité, AutoConfigBackup pour les sauvegardes automatiques chiffrées, et des procédures de restauration testées régulièrement. Sécuriser l'architecture CARP selon les bonnes pratiques.*

Contrôle 9.1 — Activer AutoConfigBackup

CIS Ref : *1.2 Ensure AutoConfigBackup is enabled* | ***MITRE :*** *T1490* | ***Niveau :*** 🟢 *L1*

Description du risque

Sans sauvegarde automatique de la configuration, une mise à jour défaillante, une panne matérielle ou une manipulation erronée peut entraîner une perte totale de la politique de sécurité. Le temps de restauration manuelle d'une configuration complexe peut dépasser plusieurs heures, durant lesquelles l'organisation est sans protection réseau.

Impact potentiel

- Perte totale de la politique de sécurité en cas de panne ou d'erreur
- Temps de restauration non maîtrisé (RTO non respecté)
- Reconfiguration manuelle incomplète ou erronée post-incident
- Indisponibilité prolongée de la connectivité réseau

Navigation

```
Services > Auto Config Backup
→ Cliquer sur "Settings" (onglet en haut)
→ Cocher "Enable ACB" (Enable Auto Config Backup)
→ Hostname : saisir un identifiant unique pour cet équipement
→   Format recommandé : fw-site-01 (ex: fw-paris-prd-01)
→ Backup password : saisir un mot de passe fort UNIQUE pour chiffrer les sauvegardes
→   IMPORTANT : ce mot de passe NE DOIT PAS être le même que le mot de passe admin
→   Utiliser un générateur de mot de passe — ≥24 caractères
→   Stocker dans un coffre-fort séparé (pas le même que l'admin pfSense)
→ Automatic backups : laisser activé (sauvegarde à chaque changement de configuration)
→ Cliquer sur Save

Pour forcer une sauvegarde immédiate :
→ Onglet "Backup Now"
→ Cliquer sur "Backup Now"
→ Vérifier que la sauvegarde apparaît dans l'onglet "Restore"

# Vérifier le contenu chiffré :
# La sauvegarde ACB utilise AES-256 pour chiffrer le fichier config.xml
# Le fichier config.xml contient : règles FW, VPN keys, certificats, mots de passe hashés,
# configuration CARP, topologie réseau complète — traiter comme HAUTEMENT CONFIDENTIEL
```

CLI de vérification

```
grep -i "autoconfigbackup\|acb" /cf/conf/config.xml | grep "enable\|hostname"
# Vérifier la configuration ACB

# Vérifier la dernière date de sauvegarde :
# Services > Auto Config Backup > Restore — vérifier la date de la dernière entrée

# Vérifier le chiffrement des sauvegardes locales :
ls -la /cf/conf/backup/
# Vérifier la présence de sauvegardes récentes

# Vérifier le mot de passe de chiffrement configuré (présence uniquement) :
grep -i "encryption_password\|backup_password" /cf/conf/config.xml
# Ne doit PAS afficher le mot de passe en clair
```

Remédiation

1. Aller dans `Services > Auto Config Backup > Settings`
2. Activer ACB
3. Configurer un mot de passe de chiffrement **fort et unique** (≥24 caractères, différent du mot de passe admin)
4. Stocker ce mot de passe dans un coffre-fort physique **séparé** du coffre-fort du mot de passe admin
5. Nommer l'équipement de manière unique pour faciliter la restauration (`fw-site-role-numéro`)
6. Laisser `Automatic backups` activé pour sauvegarder à chaque changement de configuration
7. Forcer une première sauvegarde et vérifier son apparition dans la liste
8. Compléter ACB par une sauvegarde locale chiffrée régulière : `Diagnostics > Backup & Restore > Encrypt this configuration file`
9. Stocker les sauvegardes locales hors-bande (NAS avec chiffrement, S3 avec SSE-KMS)

10. **Planifier un test de restauration trimestriel** sur un équipement de test ou une VM pfSense :
    - Déployer une VM pfSense vierge
    - Restaurer depuis une sauvegarde ACB ou locale
    - Vérifier que les interfaces, règles, VPN, services sont correctement restaurés
    - Documenter le test (date, résultat, responsable) dans le registre de tests PCA

Valeur par défaut : AutoConfigBackup désactivé par défaut. Chiffrement AES-256 disponible.

Critère de conformité : ACB activé et fonctionnel. Sauvegardes chiffrées avec mot de passe fort. Mot de passe différent du mot de passe admin. Au moins une sauvegarde dans les 24 dernières heures visible dans `Services > Auto Config Backup`. Test de restauration trimestriel documenté. Sauvegardes stockées sur au moins deux emplacements hors-bande.

Contrôle 9.2 — Configurer CARP avec interface pfsync dédiée et gestion sécurisée des VIP

CIS Ref : *1.9 Ensure a synchronized High Availability peer is configured* | ***MITRE :*** *T1499* | ***Niveau :*** 🔵 *L2*

Description du risque

Un firewall unique sans haute disponibilité est un Single Point of Failure (SPOF) critique. CARP (Common Address Redundancy Protocol) permet le basculement automatique vers un firewall secondaire en quelques secondes.

Risques spécifiques CARP/pfsync :

1. Interface pfsync non dédiée : Si le trafic pfsync (synchronisation des états de connexion) partage une interface avec le trafic utilisateur, il peut être intercepté ou perturbé, compromettant la haute disponibilité et exposant potentiellement les états de session.

2. Accès WebGUI via VIP CARP : Accéder à la WebGUI via une adresse VIP CARP est une erreur critique. En cas de basculement CARP, la session WebGUI pointe vers n'importe lequel des nœuds de manière imprévisible, causant des incohérences de configuration ou l'application de modifications sur le mauvais nœud.

3. Mot de passe CARP faible : Le mot de passe CARP (utilisé pour signer les messages d'annonce CARP) doit être fort et identique sur tous les nœuds. Un mot de passe faible permet à un attaquant d'injecter de faux messages CARP pour provoquer un basculement non autorisé.

Impact potentiel

- Coupure totale de connectivité réseau en cas de panne du firewall principal
- Interception du trafic pfsync si l'interface sync n'est pas isolée
- Basculement CARP malveillant par injection de faux messages si le mot de passe est faible
- Configuration corrompue si un administrateur modifie accidentellement le nœud secondaire via la VIP

Navigation

```
# Configurer l'interface pfsync dédiée :
Interfaces > [OPT_SYNC]
→ Activer l'interface de synchronisation dédiée (lien direct ou VLAN dédié)
→ IP statique dans un subnet isolé (ex: 10.99.99.0/30)
→ Cette interface NE DOIT PAS avoir accès à Internet ni au réseau utilisateur

System > High Availability Sync
→ Cocher "Enable State Synchronization (pfsync)"
→ pfsync Synchronize Interface : sélectionner OPT_SYNC (interface dédiée)
→ Synchronize States To : adresse IP du nœud secondaire sur l'interface SYNC
→ Cocher "Enable Configuration Synchronization (XMLRPC Sync)"
→ Remote System IP Address : IP de gestion du nœud secondaire (PAS la VIP CARP)
→ Remote System Password : mot de passe administrateur du secondaire
→ Sélectionner les éléments à synchroniser

# Configurer les VIP CARP :
Firewall > Virtual IPs
→ Créer les VIP CARP pour chaque interface :
→   Type : CARP
→   Interface : WAN (une VIP par interface)
→   IP Address : IP virtuelle partagée
→   Virtual IP Password : mot de passe FORT (≥20 caractères aléatoires)
→   VHID : numéro unique par VIP
→   Advertising Frequency : Base 1, Skew 0 (MASTER) / Skew 100 (BACKUP)

# IMPORTANT : NAT sortant avec VIP CARP :
Firewall > NAT > Outbound
→ Mode : Manual Outbound NAT
→ Pour chaque règle NAT WAN :
→   Translation Address : VIP CARP WAN (et non l'IP physique)
→   Cela garantit que le NAT survit au basculement

# Ne JAMAIS accéder à la WebGUI via la VIP CARP :
→ Utiliser toujours l'IP physique de gestion de chaque nœud
→ Documenter les IPs physiques de management dans le runbook
```

CLI de vérification

```
ifconfig pfsync0
# Vérifier l'état de l'interface pfsync et l'interface associée

carpstat
# Vérifier les statistiques CARP

ifconfig | grep "MASTER\|BACKUP\|carp"
# Vérifier le rôle CARP de chaque VIP

# Vérifier que pfsync utilise l'interface dédiée :
ifconfig pfsync0 | grep "syncdev\|syncpeer"

# Tester le basculement :
# Sur le nœud MASTER : ifconfig carp0 down
# Vérifier que le nœud BACKUP prend le rôle MASTER
# Puis : ifconfig carp0 up (retour)
```

Remédiation

1. Déployer un deuxième équipement pfSense avec une configuration matérielle identique
2. Créer une interface physique ou VLAN dédiée pour pfsync (lien direct recommandé)
3. Configurer pfsync sur cette interface dédiée — jamais sur le LAN ou WAN
4. Créer les VIP CARP avec des mots de passe forts (≥20 caractères)
5. Configurer le NAT sortant en mode manuel avec la VIP CARP comme adresse de traduction
6. Ne jamais accéder à la WebGUI via les VIP CARP — documenter les IPs de management physiques
7. Tester le basculement trimestriellement en simulant une panne du nœud MASTER
8. Documenter la procédure de basculement forcé et de retour en arrière

Valeur par défaut : CARP non configuré par défaut.

Critère de conformité : Interface pfsync sur interface dédiée isolée. Mots de passe VIP CARP forts. NAT sortant configuré avec VIP CARP. WebGUI accessible uniquement via IPs physiques. Basculement automatique fonctionnel, testé et documenté. `carpstat` ne montre pas d'erreurs.

Contrôle 9.3 — Sauvegardes chiffrées et testées régulièrement

> ***CIS Ref :*** *Best Practice* | ***MITRE :*** *T1490* | ***Niveau :*** 🟢 *L1*

Description du risque

Une sauvegarde non testée est une sauvegarde qui pourrait être inutilisable au moment critique. Les sauvegardes non chiffrées contenant la configuration pfSense exposent les clés VPN, les certificats, les mots de passe hashés et la topologie réseau complète si elles sont accédées par un tiers non autorisé.

Impact potentiel

- Impossibilité de restaurer la configuration lors d'un incident critique
- Exposition de secrets cryptographiques (clés VPN, certificats, PSK) via des sauvegardes non chiffrées
- Non-conformité aux exigences de continuité d'activité (PCA/PRA)
- Délai de remise en service non maîtrisé

Navigation

```
# Sauvegarde manuelle chiffrée (AES-256) :
Diagnostics > Backup & Restore
→ Download configuration as XML
→ Cocher "Encrypt this configuration file"
→   Le chiffrement utilise AES-256 avec le mot de passe saisi comme dérivation de clé
→   IMPORTANT : sans ce mot de passe, la configuration est irrécouvrable
→ Saisir un mot de passe de chiffrement fort (≥24 caractères)
→ Cliquer sur "Download configuration as XML"
→ Stocker le fichier chiffré dans un espace sécurisé hors-bande :
→   Option 1 : NAS avec chiffrement de volume (LUKS, ZFS encryption)
→   Option 2 : S3 bucket avec Server-Side Encryption (SSE-KMS) + versioning activé
→   Option 3 : Coffre-fort physique (pour les environnements déconnectés)

# Procédure de restauration testée trimestriellement :
Diagnostics > Backup & Restore > Restore
→ Sélectionner "Configuration area" : All (restauration complète)
→ Cocher "Restore from encrypted backup"
→ Upload le fichier de configuration chiffré
→ Saisir le mot de passe de déchiffrement
→ Cliquer sur "Restore Configuration"
→ pfSense redémarre automatiquement après restauration
→ Vérifier : interfaces actives, règles FW, VPN, services

# Contenu de la sauvegarde config.xml (à protéger comme SECRET) :
→ Règles de pare-feu complètes
→ Clés privées VPN (OpenVPN tls-crypt, IPsec PSK, WireGuard)
→ Certificats et clés privées PKI
→ Mots de passe hashés des comptes locaux
→ Configuration CARP/pfsync
→ Topologie réseau complète (IPs, VLANs, routes)
```

CLI de vérification

```
# Lister les sauvegardes locales disponibles :
ls -la /cf/conf/backup/
# Vérifier la présence de sauvegardes récentes (< 24h pour ACB)

# Vérifier le type de chiffrement utilisé :
file /cf/conf/backup/config-*.xml 2>/dev/null | head -5
# Les sauvegardes chiffrées sont en binaire (OpenSSL encrypted)

# Vérifier l'espace disponible pour les sauvegardes :
df -h /cf/conf/backup/
# Surveiller l'espace disponible

# Vérifier la date de la dernière sauvegarde :
ls -lt /cf/conf/backup/ | head -5
# La sauvegarde la plus récente doit dater de moins de 24h (si ACB actif)

# Tester l'intégrité d'une sauvegarde chiffrée (sur poste admin) :
openssl enc -d -aes-256-cbc -in config-backup-chiffre.xml -pass pass:MOTDEPASSE 2>&1 | head
-5
# Doit commencer par "<?xml version" si le déchiffrement réussit
```

Remédiation

1. Configurer des sauvegardes automatiques via AutoConfigBackup (Contrôle 9.1)
2. Programmer des sauvegardes manuelles complémentaires **avec chiffrement AES-256** :
    - Fréquence recommandée : hebdomadaire + après chaque changement majeur
    - Via `Diagnostics > Backup & Restore` avec la case `Encrypt this configuration file` cochée
3. Stocker les sauvegardes dans au moins deux emplacements distincts hors-bande :
    - NAS chiffré (LUKS/ZFS encryption) sur site
    - S3 chiffré (SSE-KMS + versioning) hors site
4. **Procédure de test de restauration trimestrielle :**
    - Documenter dans le registre PCA/PRA avec date et nom du responsable
    - Déployer une VM pfSense vierge (même version que la production)
    - Restaurer depuis la sauvegarde la plus ancienne testée
    - Valider : interfaces réseau, règles de pare-feu, VPN, services, CARP
    - Mesurer le RTO réel (objectif < 1 heure pour une configuration standard)
5. Documenter la procédure de restauration dans le runbook :
    - URL de téléchargement de pfSense (version exacte)
    - Procédure de restauration étape par étape
    - Mot de passe de déchiffrement : **uniquement en coffre-fort physique hors-ligne**
6. Stocker le mot de passe de déchiffrement dans un coffre-fort physique **indépendant et hors-ligne**

Valeur par défaut : Sauvegardes locales automatiques lors des modifications de configuration (non chiffrées par défaut). Pas de sauvegarde hors-site automatique.

Critère de conformité : Sauvegardes AES-256 chiffrées disponibles et stockées hors-site. Test de restauration trimestriel documenté (date + résultat + responsable + RTO mesuré). Mot de passe de déchiffrement stocké en coffre-fort physique hors-ligne. Sauvegardes disponibles sur ≥2 emplacements distincts.

Contrôle 9.4 — Activer l'accélération cryptographique matérielle (AES-NI)

> ***CIS Ref :*** *Best Practice* | ***MITRE :*** *Réduction surface d'attaque side-channel* | ***Niveau :*** 🔵 *L2*

Description du risque

Sans accélération cryptographique matérielle, le chiffrement VPN (AES-256-GCM pour OpenVPN/IPsec, ChaCha20 pour WireGuard) est entièrement exécuté par le CPU principal (software crypto). Cela crée deux problèmes de sécurité complémentaires :

1. Timing side-channel risk : Les implémentations logicielles du chiffrement AES sans AES-NI sont potentiellement vulnérables aux attaques par canal auxiliaire temporel (timing attacks), car la durée des opérations peut varier selon les données traitées. L'instruction AES-NI offre un temps d'exécution constant.

2. Performance et disponibilité : Un CPU saturé par le chiffrement logiciel peut devenir un vecteur de DoS indirect — des sessions VPN nombreuses épuisent le CPU, dégradant la capacité d'inspection du pare-feu. L'offloading crypto matériel libère le CPU pour les fonctions de filtrage.

Prérequis : Vérifier que le processeur supporte AES-NI avant activation. La majorité des processeurs Intel (depuis 2010) et AMD (depuis 2011) incluent AES-NI. Les plateformes Netgate SG-xxxx intègrent généralement AES-NI ou des accélérateurs dédiés.

Impact potentiel si non activé

- Saturation CPU sur des charges VPN élevées, dégradation des performances de filtrage
- Potentielle vulnérabilité aux timing attacks sur les implémentations software AES
- Limitation du débit VPN par rapport aux capacités matérielles disponibles
- Non-utilisation des capacités de sécurité matérielles du processeur

Navigation

```
# Vérifier le support AES-NI du processeur (avant activation) :
Diagnostics > Command Prompt
→ Saisir : grep -c aes /proc/cpuinfo
→   0 = pas d'AES-NI (ne pas activer)
→   1 ou plus = AES-NI supporté (activer)

# Activer l'accélération crypto :
System > Advanced > Miscellaneous
→ Section "Cryptographic & Thermal Hardware"
→ Cryptographic Hardware :
→   Sélectionner "AES-NI CPU-based acceleration and BSD Crypto Device (aesni)"
→   OU : sélectionner "BSD Crypto Device (cryptodev)" si accélérateur autre qu'AES-NI
→ Thermal Sensors : sélectionner le type de capteur correspondant au CPU (si disponible)
→ Cliquer sur Save
→ Redémarrer pfSense pour appliquer
```

> ***Plateformes Netgate spécifiques :*** *Les appliances Netgate SG-4100, SG-6100, SG-7100 et supérieures disposent d'accélérateurs cryptographiques dédiés (QAT pour certains modèles Intel). Consulter la documentation Netgate pour le pilote approprié.*

CLI de vérification

```
# Vérifier que le module AES-NI est chargé :
kldstat | grep aesni
# Doit retourner : aesni.ko chargé

# Vérifier les capacités OpenSSL avec accélération matérielle :
openssl engine -t -c
# Doit retourner : (aesni) Intel AES-NI engine [available]
#   ou : (cryptodev) BSD cryptodev engine [available]

# Vérifier le support AES-NI du CPU :
grep -c aes /proc/cpuinfo
# Si > 0 : AES-NI supporté

# Benchmark AES avant/après activation (comparaison) :
openssl speed -evp aes-256-gcm 2>/dev/null | tail -3
# Comparer les Mo/s avant et après activation

# Vérifier les algorithmes accélérés disponibles :
openssl engine aesni -t 2>/dev/null | head -10
```

Remédiation

1. Vérifier le support AES-NI : `grep -c aes /proc/cpuinfo` — si 0, ne pas activer
2. Aller dans `System > Advanced > Miscellaneous`
3. Section `Cryptographic & Thermal Hardware` → sélectionner `AES-NI CPU-based acceleration and BSD Crypto Device (aesni)`
4. Sauvegarder et redémarrer pfSense
5. Vérifier le chargement du module : `kldstat | grep aesni`
6. Vérifier l'accélération OpenSSL : `openssl engine -t -c`
7. Observer la réduction de charge CPU lors des sessions VPN actives : `top` ou `System > Activity`

Valeur par défaut : Accélération cryptographique non activée par défaut — sélection manuelle requise.

Critère de conformité : Si le CPU supporte AES-NI, le module `aesni.ko` est chargé (`kldstat | grep aesni` retourne une ligne). `openssl engine -t -c` confirme `(aesni)` disponible. Charge CPU VPN réduite par rapport à la configuration sans accélération.

Contrôle 9.5 — Isolation réseau par VLAN pour les services critiques

> ***CIS Ref :*** *CIS Control 12.2 (Managed Network Infrastructure Devices)* | ***MITRE :*** *T1021 (Remote Services — Lateral Movement), T1046 (Network Service Discovery), T1590 (Gather Victim Network Information)* | ***Niveau :*** 🔵 *L2*

Description du risque

Un réseau plat sans segmentation par VLAN est l'un des facteurs aggravants les plus fréquents lors d'incidents de sécurité. Si un hôte est compromis sur un réseau non segmenté, l'attaquant peut atteindre directement tous les autres hôtes : serveurs, équipements IoT, postes de direction, infrastructure de management. La segmentation VLAN limite le rayon de blast d'une compromission et force les mouvements latéraux à passer par le pare-feu (et ses règles).

Architecture VLAN recommandée pour pfSense :

VLAN	NOM	SOUS-RÉSEAU	ACCÈS INTERNET	ACCÈS INTER-VLAN
10	Management	192.168.10.0/28	Non	Vers tout (admin)
20	Servers	192.168.20.0/24	Limité	Non (sauf règles explicites)
30	Users	192.168.30.0/24	Oui (filtré)	Vers Servers (ports spécifiques)
40	DMZ	192.168.40.0/24	Oui	Non (inbound)
50	IoT	192.168.50.0/24	Limité (whitelist)	Non
60	Guest	192.168.60.0/24	Oui	Non
99	CARP Sync	10.99.99.0/30	Non	Non

Risques liés à l'absence de segmentation : - Mouvement latéral direct depuis un poste utilisateur compromis vers les serveurs (T1021) - Scan de tous les équipements réseau depuis un hôte IoT compromis (T1046) - Accès à l'interface de management pfSense depuis le réseau utilisateur ordinaire - Un équipement IoT infecté (caméra, imprimante) peut attaquer directement les serveurs - Reconnaissance réseau facilitée (T1590) en l'absence de séparation des segments

Impact potentiel

- Compromission d'un poste utilisateur permettant d'atteindre directement les serveurs critiques
- Propagation rapide d'un ransomware sur l'ensemble du réseau non segmenté
- Équipements IoT compromis servant de pivot vers l'infrastructure de production
- Impossibilité d'isoler rapidement un hôte compromis sans coupure globale

Navigation

```
# Étape 1 — Création des VLANs :
Interfaces > Assignments > VLANs
→ Cliquer sur "+ Add" pour chaque VLAN :
→   Parent Interface : l'interface physique (ex: em1, igb1)
→   VLAN Tag : numéro VLAN (10, 20, 30, etc.)
→   Description : nom du VLAN (Management, Servers, Users, IoT, Guest)
→ Sauvegarder pour chaque VLAN

# Étape 2 — Assignation des interfaces VLAN :
Interfaces > Assignments
→ Pour chaque VLAN créé :
→   Available network ports : sélectionner le VLAN (ex: VLAN 10 on em1)
→   Cliquer sur "+ Add"
→   Configurer l'interface : activer, IP statique, sous-réseau /24 ou /28

# Étape 3 — Configuration des interfaces VLAN :
Interfaces > [OPT_MGMT] (VLAN 10)
→ Enable : cocher
→ IPv4 Configuration Type : Static IPv4
→ IPv4 Address : 192.168.10.1 /28
→ Sauvegarder
→ Répéter pour chaque VLAN

# Étape 4 — Règles de pare-feu inter-VLAN (isolation par défaut) :
Firewall > Rules > [VLAN Users - OPT3]
→ Règle 1 : Block + Log — Users vers VLAN Management (192.168.10.0/28)
→   Empêche les utilisateurs d'accéder à la gestion réseau
→ Règle 2 : Block + Log — Users vers VLAN Servers port 3389, 22, 5985
→   Empêche l'accès RDP/SSH/WinRM direct depuis les postes utilisateurs
→ Règle 3 : Pass — Users vers Servers port 80, 443, 445 (HTTP/HTTPS/SMB si justifié)
→ Règle 4 : Pass — Users vers Internet port 80, 443 (navigation web filtrée)
→ Règle 5 : Block + Log — Users vers tout (règle de refus implicite explicite)

Firewall > Rules > [VLAN IoT - OPT5]
→ Règle 1 : Block + Log — IoT vers ALL VLANs internes (pas de routage vers LAN)
→ Règle 2 : Pass — IoT vers Internet (services spécifiques whitelist uniquement)
→   Ex: TCP 443 vers serveurs.cloud-iot-provider.com (alias spécifique)
→ Règle 3 : Block + Log — IoT vers tout (refus implicite)

Firewall > Rules > [VLAN Guest - OPT6]
→ Règle 1 : Block + Log — Guest vers ALL VLANs internes (aucun accès interne)
→ Règle 2 : Pass — Guest vers Internet port 80, 443 uniquement
→ Règle 3 : Block + Log — Guest vers tout (refus implicite)

Firewall > Rules > [VLAN DMZ - OPT4]
→ Règle 1 : Block + Log — DMZ vers VLANs internes (la DMZ ne doit PAS initier vers
l'interne)
→   Exception : DMZ vers base de données sur port spécifique si architecture en 3 tiers
→   Documenter chaque exception avec justification et ticket de changement
→ Règle 2 : Pass — DMZ vers Internet (mise à jour logiciels, DNS)

Firewall > Rules > [VLAN Management - OPT1]
→ Règle 1 : Pass — Management vers ALL (accès complet pour les administrateurs)
→   Restreindre aux adresses IP des postes d'administration uniquement
→ Règle 2 : Block + Log — tout vers Management hors OPT1 réseau
→   Créer comme règle flottante pour couvrir toutes les interfaces
```

```
# Étape 5 — Configuration du switch (si switch managé) :
→ Configurer les ports switch en mode Access (untagged) pour chaque VLAN
→ Configurer les ports trunk vers pfSense avec tous les VLANs tagués
→ Désactiver les ports switch non utilisés
→ Documenter la topologie VLAN dans la cartographie réseau

# Étape 6 — DHCP par VLAN :
Services > DHCP Server > [chaque interface VLAN]
→ Activer DHCP pour chaque VLAN avec la plage correspondante
→ DNS Server : pointer vers pfSense (forcer la résolution via Unbound)
→ Default gateway : IP pfSense sur ce VLAN
→ Lease time : adapter selon l'usage (IoT : plus long, Guest : plus court)
```

CLI de vérification

```
# Lister toutes les interfaces VLAN configurées :
ifconfig | grep vlan
# Chaque VLAN doit apparaître avec son adresse IP assignée

# Vérifier les interfaces VLAN et leurs adresses :
ifconfig | grep -A3 "vlan"

# Vérifier les règles inter-VLAN (blocage explicite de la DMZ vers l'interne) :
pfctl -sr | grep "block" | grep -E "vlan|OPT"
# Les règles de blocage inter-VLAN doivent être présentes

# Tester l'isolation (depuis un hôte IoT, ne doit pas pinger un serveur) :
# ping 192.168.20.10 (depuis 192.168.50.x)
# Doit être bloqué et logué

# Vérifier les états actifs inter-VLAN :
pfctl -s states | grep -E "192\.168\.50.*192\.168\.20|192\.168\.60.*192\.168\.10"
# Doit retourner vide si l'isolation est effective

# Vérifier la configuration DHCP par VLAN :
grep -i "subnet\|interface" /var/etc/dhcpd.conf | head -20
# Chaque sous-réseau VLAN doit apparaître

# Vérifier que le trafic inter-VLAN passe par pfSense (pas de routage direct) :
netstat -rn | grep -E "192\.168\."
# Toutes les routes doivent passer par pfSense comme gateway
```

Remédiation

1. Planifier l'architecture VLAN avec l'équipe réseau (cartographie des flux, matrice de flux inter-VLAN)
2. Créer les VLANs dans `Interfaces > Assignments > VLANs`
3. Assigner et configurer chaque interface VLAN avec une adresse IP dédiée
4. Configurer DHCP sur chaque VLAN avec DNS pointant vers pfSense
5. Implémenter les règles de pare-feu inter-VLAN avec le principe du moindre privilège :
    - Management : accès complet aux autres VLANs (depuis IPs admin uniquement)
    - Users : accès aux services spécifiques sur Servers (ports documentés uniquement)
    - IoT : aucun accès aux VLANs internes, Internet whitelist uniquement

- Guest : Internet uniquement (port 80/443), aucun accès interne
- DMZ : pas d'initiation vers l'interne (sauf exceptions documentées avec ticket)

6. Configurer le switch managé avec les VLANs correspondants (ports Access/Trunk)
7. Documenter la topologie VLAN dans la cartographie réseau
8. Tester l'isolation de chaque VLAN avant mise en production
9. Migrer progressivement les équipements par VLAN pour éviter les interruptions

Valeur par défaut : pfSense créé avec une seule interface LAN. Aucune segmentation VLAN par défaut.

Critère de conformité : Au minimum 3 VLANs distincts créés (Management, Users, Services/Servers). VLAN IoT et/ou Guest isolé sans accès aux VLANs internes. VLAN Management accessible uniquement depuis les postes d'administration. Règles de blocage implicites explicites sur chaque VLAN. Topologie VLAN documentée dans la cartographie réseau. Tests d'isolation documentés (ping inter-VLAN bloqué confirmé).

Domaine 10 — Journalisation et supervision

Objectif : *Centraliser les logs pfSense vers un SIEM distant, activer la journalisation de toutes les règles de pare-feu, configurer la rotation et la rétention des logs, et mettre en place des alertes sur les événements critiques de sécurité. Protéger l'intégrité des logs contre toute altération.*

Contrôle 10.1 — Configurer un syslog distant vers SIEM

CIS Ref : *6.1 Ensure syslog is configured* | ***MITRE :*** *T1562.006* | ***Niveau :*** 🟢 *L1*

Description du risque

Les logs stockés localement sur pfSense sont perdus en cas de compromise (un attaquant peut effacer les logs), de panne matérielle ou de réinstallation. Sans syslog distant, il est impossible d'avoir une vue centralisée des événements de sécurité, de corréler les événements avec d'autres équipements, ou de respecter les exigences de rétention légale des logs.

La protection de l'intégrité des logs est fondamentale : un attaquant qui compromet pfSense effacera systématiquement les traces. L'envoi en temps réel vers un SIEM externe protège l'intégrité forensique.

Impact potentiel

- Effacement des logs par un attaquant après compromission (anti-forensique)
- Impossibilité de corréler les événements de sécurité dans un SIEM
- Perte des logs en cas de panne matérielle
- Non-conformité aux exigences légales de rétention des logs (LPM : 1 an, PCI DSS : 1 an)

Navigation

```
Status > System Logs > Settings
→ Section "Remote Logging Options"
→ Cocher "Enable Remote Logging"
→ Remote log servers : saisir l'IP:port du serveur syslog (ex: 192.168.10.10:514)
→ Remote Syslog Contents :
→   Cocher : Firewall Events, DHCP Events, Authentication Events,
→            Network Events, System Events, DNS Events, VPN Events
→ Source Address : sélectionner l'interface de management
→ IP Protocol : IPv4 (ou IPv6 si utilisé)
→ Cliquer sur Save

# Recommandé : syslog over TLS (chiffrement du transport) :
→ Port 6514 avec TLS pour éviter l'interception des logs en transit
```

CLI de vérification

```
grep -i "syslog\|remote" /cf/conf/config.xml | grep "enable\|ipaddr\|port"
# Vérifier la configuration syslog
# Sur le serveur syslog distant, vérifier la réception des logs :
tail -f /var/log/syslog | grep <IP_pfSense>
```

Remédiation

1. Déployer un serveur syslog ou SIEM (Graylog, ELK, Splunk, Wazuh)
2. Aller dans `Status > System Logs > Settings`
3. Activer le logging distant et configurer l'adresse du serveur SIEM
4. Sélectionner toutes les catégories d'événements à envoyer
5. Utiliser syslog over TLS (TCP/6514) pour chiffrer le transport des logs
6. Vérifier la réception des logs sur le SIEM
7. Créer des règles de corrélation pour les événements critiques (authentifications échouées répétées, règles modifiées, connexions admin, activité VPN, changements CARP)

Valeur par défaut : Logging local uniquement, syslog distant désactivé.

Critère de conformité : Syslog distant activé et fonctionnel. Au moins un serveur syslog configuré. Toutes les catégories d'événements de sécurité transmises. Logs visibles et datés correctement sur le serveur distant. Rétention ≥ 90 jours (recommandé : 1 an). Transport TLS activé (port 6514).

Contrôle 10.2 — Activer le logging sur toutes les règles de pare-feu

> ***CIS Ref :*** *4.1.5 Ensure Logging is Enable for All Firewall Rules* | ***MITRE :*** *T1562.006* | ***Niveau :*** 🟢 *L1*

> ***Note :*** *Ce contrôle complète le Contrôle 4.5 en se concentrant sur la vérification et la supervision des logs de règles.*

Description du risque

Sans logging des règles de pare-feu individuelles, les événements de sécurité (tentatives d'intrusion bloquées, communications C2 bloquées) passent inaperçus. La détection et la réponse aux incidents sont impossibles sans visibilité sur le trafic bloqué et autorisé.

Impact potentiel

- Tentatives d'intrusion bloquées non détectées et non alertées
- Impossibilité de construire une ligne de base du trafic normal pour détecter les anomalies
- Investigation forensique impossible après un incident

Navigation

```
Firewall > Rules
→ Pour chaque règle (Pass et Block) :
→   Éditer la règle
→   Section "Extra Options" > cocher "Log"
→   Description : documenter la justification de la règle
→   Sauvegarder

Status > System Logs > Firewall
→ Vérifier que les événements apparaissent en temps réel
→ Filtrer par interface ou adresse pour valider le logging
```

CLI de vérification

```
pfctl -vsr | grep -c "log"
# Compter le nombre de règles avec le flag log
tail -f /var/log/filter.log
# Observer le flux de logs en temps réel
clog /var/log/filter.log | grep -c "."
# Compter les entrées dans le log de filtrage
```

Remédiation

1. Vérifier systématiquement que toutes les règles actives ont `Log` coché
2. Utiliser le script de vérification pour identifier les règles sans logging : `pfctl -vsr | grep -v log | grep "^pass\|^block"`
3. Configurer des alertes sur les événements critiques dans le SIEM (nombreuses connexions bloquées depuis une même IP, tentatives vers des ports sensibles)
4. Mettre en place un tableau de bord de supervision des règles de blocage

Valeur par défaut : Logging activé uniquement sur les règles de blocage par défaut (bogon, private networks).

Critère de conformité : 100% des règles `Pass` et `Block` ont le flag `Log` activé. `pfctl -vsr | grep -v log | grep "^pass"` retourne une liste vide (ou uniquement des règles système justifiées).

Contrôle 10.3 — Rotation et rétention des logs

> ***CIS Ref :*** *Best Practice* | ***MITRE :*** *T1562.006* | ***Niveau :*** 🟢 *L1*

Description du risque

pfSense utilise des logs circulaires (clog) de taille fixe par défaut. Si la taille est trop petite, des événements critiques sont écrasés rapidement, rendant impossible toute investigation rétrospective. La rétention locale doit être dimensionnée en fonction des exigences légales et des besoins forensiques.

Impact potentiel

- Écrasement de logs critiques avant toute détection d'incident
- Impossibilité d'investigation rétrospective sur des incidents anciens
- Non-conformité aux obligations légales de rétention (LPM, PCI DSS, RGPD)

Navigation

```
Status > System Logs > Settings
→ Section "General Logging Options"
→ Log File Size (Bytes) : augmenter à 2097152 (2MB) ou plus selon les ressources
→ Log Rotation : configurer la fréquence de rotation
→ Remote Logging : s'assurer que tous les logs sont envoyés au SIEM (voir 10.1)
→ Cliquer sur Save
```

CLI de vérification

```
ls -la /var/log/*.log
# Vérifier la taille des fichiers de log
df -h /var/log
# Vérifier l'espace disque disponible pour les logs
# Afficher la taille configurée des logs circulaires :
grep -i "logfilesize" /cf/conf/config.xml
```

Remédiation

1. Aller dans `Status > System Logs > Settings`
2. Augmenter la taille des fichiers de log (`Log File Size`) à 2 Mo minimum par catégorie
3. Activer le syslog distant (Contrôle 10.1) pour la rétention long terme
4. Sur le serveur SIEM, configurer une rétention des logs pfSense d'au moins 90 jours (recommandé : 1 an pour PCI DSS)
5. Implémenter la rotation des logs avec archivage compressé sur le SIEM

Valeur par défaut : Taille des logs circulaires à 512 Ko par défaut, sans rotation explicite.

Critère de conformité : Taille des logs locaux ≥ 2 Mo par catégorie. Logs archivés sur SIEM avec rétention ≥ 90 jours. Espace disque suffisant pour les logs sans compression (`df /var/log` montre > 50% d'espace libre).

Contrôle 10.4 — Alertes sur événements critiques

CIS Ref : *Best Practice* | ***MITRE :*** *T1562* | ***Niveau :*** 🔵 *L2*

Description du risque

La collecte de logs sans alertes proactives sur les événements critiques réduit la valeur opérationnelle de la supervision. Les incidents de sécurité détectés tardivement ont des impacts plus importants. Des alertes en temps réel sur les événements critiques permettent une réponse rapide avant que la situation ne s'aggrave.

Impact potentiel

- Délai de détection des incidents augmenté sans alertes proactives
- Brèches non détectées pendant des jours, semaines ou mois
- Non-conformité aux exigences de délai de notification (RGPD : 72h, LPM)

Navigation

```
Status > System Logs > Settings
→ Activer l'envoi des logs vers le SIEM (Contrôle 10.1)

Sur le SIEM (Graylog, Splunk, ELK, Wazuh) :
→ Créer des règles d'alerte pour :
→   - Authentifications SSH échouées > 5 en 1 minute depuis une IP
→   - Connexions à la WebGUI échouées > 3 en 5 minutes
→   - Modification de règles de pare-feu (hors fenêtre de maintenance)
→   - Blocages vers des IPs dans les listes de Threat Intelligence
→   - Démarrage/arrêt de services critiques (OpenVPN, IPsec, pfsync)
→   - Ajout ou suppression d'un compte utilisateur
→   - Changement de configuration système
→   - Connexions administratives WebGUI depuis une IP inconnue
→   - Changement d'état CARP (MASTER→BACKUP ou BACKUP→MASTER)
→   - Installation ou suppression d'un paquet

Diagnostics > Edit File > /etc/rc.local
→ Configurer des alertes email via SMTP pour les événements critiques pfSense
```

CLI de vérification

```
# Tester l'envoi d'une alerte test depuis pfSense
logger -p auth.crit "TEST_ALERT: Verif alerting pfSense"
# Vérifier la réception sur le SIEM et par email
grep "TEST_ALERT" /var/log/system.log
```

Remédiation

1. Configurer les notifications email pfSense dans `System > Advanced > Notifications`
2. Sur le SIEM, créer des règles d'alerte pour chaque type d'événement critique
3. Définir des canaux de notification (email, Slack, PagerDuty, SMS) selon la criticité
4. Tester chaque règle d'alerte avec un événement simulé
5. Documenter le plan d'escalade pour chaque type d'alerte

Valeur par défaut : Aucune alerte configurée par défaut.

Critère de conformité : Alertes configurées pour au moins 5 types d'événements critiques. Tests d'alertes effectués et documentés. Délai de réception des alertes < 5 minutes. Plan d'escalade documenté et distribué à l'équipe SOC/admin.

Réponse à incident

Indicateurs de compromission

INDICATEUR	SIGNIFICATION	ACTION IMMÉDIATE
Connexions administratives depuis une IP inconnue	Accès non autorisé à la WebGUI	Bloquer l'IP source, invalider toutes les sessions, révoquer les tokens
Modifications de règles de pare-feu non planifiées	Manipulation de la politique de sécurité	Restaurer depuis dernière sauvegarde connue bonne, audit forensique complet
Nouveaux comptes utilisateurs créés sans ticket	Persistence d'un attaquant	Désactiver le compte, analyser les actions effectuées, investigation forensique
Trafic sortant vers des IP de Threat Intelligence	Communication C2 potentielle	Isolation réseau immédiate, analyse forensique des hôtes impliqués
Augmentation anormale des connexions bloquées depuis une IP	Scan ou tentative d'exploitation	Blocage automatique via Login Protection, analyse des patterns
Certificats VPN utilisés depuis plusieurs localisations simultanées	Vol de certificat ou credential	Révoquer le certificat, renouveler, investiguer le poste compromis
Modification de la configuration pfsync/CARP	Tentative de désactivation HA	Vérifier intégrité des deux nœuds, basculement sur le secondaire si nécessaire
Services OpenVPN ou IPsec redémarrés sans planification	Injection de configuration malveillante	Vérifier la configuration VPN, comparer avec la sauvegarde de référence
Installation d'un nouveau paquet non planifiée	**Persistence ou backdoor via package**	**Identifier et supprimer le paquet, audit complet de la configuration**
Export de configuration XML via WebGUI	**Exfiltration de secrets (clés, certs)**	**Auditer les sessions administratives, changer toutes les clés et certificats**
Changement d'état CARP inattendu (MASTER↔BACKUP)	**Tentative de DoS ou attaque CARP**	**Vérifier la connectivité réseau, analyser les annonces CARP, contrôler les mots de passe VIP**
Règles bogon ou RFC1918 WAN désactivées	**Suppression délibérée de protection**	**Réactiver immédiatement, auditer qui a effectué la modification**
Processus Suricata ou pfBlockerNG arrêtés	**Désactivation des protections IDS/IPS**	**Redémarrer les services, investiguer la cause, vérifier l'intégrité des configs**
Alertes Suricata sur ET.Botcc ou ET.Trojan	**Hôte interne potentiellement infecté**	**Isoler l'hôte source, investigation forensique, scan antivirus**

Indicateurs de compromission spécifiques pfSense (IoCs)

Les IoCs suivants sont spécifiques à pfSense et indiquent une compromission avancée :

```
# IoC 1 : Nouveau compte local créé sans autorisation
grep -A5 "<user>" /cf/conf/config.xml | grep "<name>" | grep -v "admin\|<known_accounts>"

# IoC 2 : Configuration de règle NAT ajoutée silencieusement
grep -c "<nat>" /cf/conf/config.xml
# Comparer avec la valeur de référence documentée

# IoC 3 : Modification du fichier de configuration hors WebGUI
md5 /cf/conf/config.xml
# Comparer avec le hash enregistré lors de la dernière modification autorisée

# IoC 4 : Paquet installé sans ticket de changement
pkg info | diff - /var/db/known_packages.txt
# Fichier de référence à maintenir manuellement

# IoC 5 : Connexion SSH depuis une IP hors-réseau management
last | grep -v "<IP_MGMT_RANGE>"

# IoC 6 : Modification de la configuration CARP/pfsync
grep -i "carp\|pfsync\|vip" /var/log/system.log | tail -50

# IoC 7 : Règles flottantes inattendues (souvent utilisées par les backdoors)
pfctl -sr | grep "floating"
# Comparer avec la liste des règles flottantes autorisées

# IoC 8 : Processus non attendus sur pfSense
ps aux | grep -v -E "nginx|php-fpm|unbound|ntpd|syslogd|cron|pfSense|kernel|dpinger|
suricata|snort|miniupnpd"
```

Procédure d'isolation d'urgence

```
# 1. Capturer l'état courant avant toute intervention
pfctl -sr > /tmp/rules_backup_$(date +%Y%m%d_%H%M%S).txt
pfctl -sa > /tmp/full_state_$(date +%Y%m%d_%H%M%S).txt
cp /cf/conf/config.xml /tmp/config_backup_$(date +%Y%m%d_%H%M%S).xml
md5 /cf/conf/config.xml > /tmp/config_md5_$(date +%Y%m%d_%H%M%S).txt

# 2. Identifier les connexions administratives actives
who
last | head -20
grep "Successful\|Failed" /var/log/auth.log | tail -50

# 3. Bloquer une IP source suspecte immédiatement
pfctl -t bruteforce -T add <IP_SUSPECTE>
# Ou via les règles de pare-feu dans l'interface WebGUI :
# Firewall > Rules > [interface] > Ajouter une règle Block en haut de liste

# 4. Invalider toutes les sessions WebGUI actives
# System > Advanced > Admin Access > Sessions
# Ou redémarrer le service web :
pfSsh.php playback svc restart nginx

# 5. Extraire les logs pour forensique
clog /var/log/filter.log > /tmp/firewall_logs_$(date +%Y%m%d).txt
clog /var/log/system.log > /tmp/system_logs_$(date +%Y%m%d).txt
clog /var/log/auth.log > /tmp/auth_logs_$(date +%Y%m%d).txt

# 6. Vérifier l'intégrité de la configuration
md5 /cf/conf/config.xml
# Comparer avec le hash de la dernière sauvegarde connue bonne

# 7. Vérifier les paquets installés vs. baseline
pkg info > /tmp/packages_current.txt
# Comparer avec la liste de référence

# 8. En cas de compromission CARP : forcer le basculement vers le secondaire
ifconfig carp0 down
# (Remplacer carp0 par l'interface CARP principale)

# 9. Si compromise confirmée : restaurer depuis sauvegarde
# Diagnostics > Backup & Restore > Restore
# Sélectionner la dernière sauvegarde antérieure à l'incident
```

Commandes forensiques avancées (investigation post-incident)

Les commandes suivantes sont à exécuter depuis la console pfSense (SSH ou physique) lors d'une investigation. Elles permettent d'identifier des signes d'intrusion, de collecte de données et de persistance.

```
# ============================================================
# ANALYSE DES PROCESSUS SUSPECTS
# ============================================================
# Lister les processus par consommation CPU (anomalies potentielles)
ps aux | grep -v "^\[" | sort -k3 -rn | head -20

# Processus avec connexions réseau actives (miners, reverse shells)
sockstat -4 -l
# Vérifier tout port inhabituel écouté sur 0.0.0.0

# ============================================================
# ANALYSE DES FICHIERS MODIFIÉS RÉCEMMENT
# ============================================================
# Fichiers modifiés après le dernier config.xml (modifications suspectes)
find /cf/conf -newer /cf/conf/config.xml -type f 2>/dev/null

# Fichiers PHP modifiés dans le webroot (webshells potentiels)
find /usr/local/www -name "*.php" -newer /cf/conf/config.xml 2>/dev/null

# ============================================================
# INTÉGRITÉ DES PAQUETS INSTALLÉS
# ============================================================
# Vérifier l'intégrité des fichiers des paquets installés
pkg check -s 2>&1 | grep "FAILED"
# Toute ligne FAILED indique une altération de fichier système

# ============================================================
# LOGS D'AUTHENTIFICATION
# ============================================================
# Consulter les logs d'authentification (SSH, WebGUI)
clog /var/log/auth.log | tail -100

# Connexions SSH réussies récentes (par utilisateur et IP source)
last | head -30

# ============================================================
# STATISTIQUES ET ÉTAT DU FIREWALL
# ============================================================
# Statistiques générales pfSense (paquets bloqués, états)
pfctl -s info

# Vérifier l'intégrité des règles de pare-feu (fingerprint)
pfctl -sr | md5
# Comparer avec la valeur de référence documentée

# ============================================================
# CLÉS SSH NON AUTORISÉES
# ============================================================
# Rechercher des clés SSH autorisées potentiellement injectées
cat ~/.ssh/authorized_keys 2>/dev/null
# Comparer avec les clés enregistrées dans System > User Manager

# Vérifier pour tous les comptes :
grep -r "authorized_keys" /home /root /var/home 2>/dev/null

# ============================================================
```

```
# COMPARAISON DE CONFIGURATION (DIFF FORENSIQUE)
# ============================================================
# Comparer la config actuelle avec la dernière sauvegarde
LATEST_BACKUP=$(ls -t /cf/conf/backup/config-*.xml 2>/dev/null | head -1)
if [ -n "$LATEST_BACKUP" ]; then
    diff /cf/conf/config.xml "$LATEST_BACKUP" | head -50
fi
# Toute différence non planifiée est suspecte

# ============================================================
# TRAFIC RÉSEAU EN TEMPS RÉEL
# ============================================================
# Voir les connexions en temps réel par interface
pfctl -s states | head -30

# Trafic par interface (compteurs octets/paquets)
pfctl -vsi

# Identifier des flux suspects vers l'extérieur depuis pfSense lui-même
sockstat -4 | grep -v "127.0.0.1\|::\|LISTEN"
```

```
# ============================================================
# ANALYSE DES CONNEXIONS RÉSEAU ACTIVES (FORENSIQUE AVANCÉE)
# ============================================================
# Connexions actives hors loopback (identifier les connexions suspectes sortantes depuis
pfSense)
sockstat -4 -l | grep -v "127.0.0.1"
# Toute connexion établie depuis pfSense lui-même (PID non standard) est suspecte

# ============================================================
# INVENTAIRE DES PAQUETS INSTALLÉS (BASELINE)
# ============================================================
# Exporter la liste des paquets pour comparaison avec la baseline
pkg info | awk '{print $1}' > /tmp/packages_$(date +%Y%m%d_%H%M%S).txt
# Comparer avec la baseline :
diff /tmp/packages_$(date +%Y%m%d_%H%M%S).txt /var/db/known_packages_baseline.txt

# ============================================================
# TÂCHES CRON SUSPECTES
# ============================================================
# Vérifier les crontabs (vecteur de persistence pour les backdoors)
crontab -l
cat /etc/crontab
ls -la /etc/cron.d/
# Toute entrée non documentée est suspecte — comparer avec la baseline

# ============================================================
# FICHIERS SUID SUSPECTS
# ============================================================
# Rechercher des fichiers SUID non attendus (escalade de privilèges potentielle)
find / -perm -4000 -type f 2>/dev/null | head -20
# Comparer avec la liste standard FreeBSD/pfSense :
# /usr/bin/su, /usr/bin/passwd, /sbin/ping, /usr/sbin/ppp
# Tout fichier SUID hors de cette liste est hautement suspect

# ============================================================
# MONITORING DES LOGS SYSTÈME EN TEMPS RÉEL
# ============================================================
# Surveiller les erreurs et warnings système en temps réel
clog /var/log/system.log | grep -i "error\|warn\|fail\|refused\|denied" | tail -50

# Surveiller les logs d'authentification en temps réel
clog /var/log/auth.log | grep -i "failed\|invalid\|blocked\|unauthorized" | tail -30

# ============================================================
# ÉTAT DU PARE-FEU (VÉRIFICATION D'INTÉGRITÉ)
# ============================================================
# Compter les états actifs dans la table de connexions
pfctl -s state | wc -l
# Une valeur anormalement élevée peut indiquer une attaque DoS ou un scan massif

# Vérifier les états par IP source (identifier les sources volumétriques)
pfctl -s state | awk '{print $3}' | cut -d: -f1 | sort | uniq -c | sort -rn | head -20

# ============================================================
# VÉRIFICATION DES CERTIFICATS
# ============================================================
```

```
# Vérifier la validité des certificats TLS de la WebGUI
echo | openssl s_client -connect localhost:443 2>/dev/null | openssl x509 -noout -dates
# Vérifier : notBefore et notAfter — alerter si expiration < 30 jours

# Vérifier les certificats VPN OpenVPN :
for cert in /var/etc/openvpn/server*.crt; do
  echo "=== $cert ==="
  openssl x509 -in "$cert" -noout -enddate 2>/dev/null
done
```

Recommandation : *Automatiser la collecte de ces données à intervalles réguliers et envoyer les sorties vers le SIEM pour détecter les anomalies par corrélation temporelle. Créer un script cron sur pfSense (`/etc/cron.d/`) qui capture l'état système quotidiennement et le compare à J-1.*

Commandes forensiques avancées — Threat Hunting

Les commandes ci-dessous constituent un kit de threat hunting complet pour pfSense. Elles permettent de détecter des compromissions actives, des mécanismes de persistance et des indicateurs de C2. À exécuter depuis la console pfSense (SSH ou physique) lors d'une investigation proactive ou réactive.

```
# ============================================================
# FORENSIQUE PFSENSE — THREAT HUNTING COMPLET
# ============================================================

# --- INTÉGRITÉ SYSTÈME ---
# Vérifier la version et l'intégrité de pfSense
cat /etc/version
pkg check -s 2>&1 | grep "FAILED"

# Comparer la configuration actuelle avec la dernière sauvegarde
diff /cf/conf/config.xml /cf/conf/backup/$(ls -t /cf/conf/backup/ | head -1)

# --- PROCESSUS SUSPECTS ---
# Top 20 processus par CPU (rechercher des inconnus)
ps aux | grep -v "^\[" | sort -k3 -rn | head -20

# Processus avec connexions réseau ouvertes
sockstat -4 | grep -v "127.0.0.1"

# --- RÈGLES PARE-FEU ---
# Vérifier l'intégrité des règles actives
pfctl -sr | md5sum
pfctl -s rules | wc -l

# Identifier les règles récemment modifiées (floating rules suspectes)
pfctl -s rules | grep "floating"

# --- RÉSEAU ---
# Connexions établies sortantes (potentiel C2)
sockstat -4 -c | grep "ESTABLISHED" | grep -v "127.0.0.1"

# Table d'état du firewall (taille anormale = indicateur)
pfctl -s state | wc -l

# Résolutions DNS suspectes
clog /var/log/resolver.log | tail -200 | grep -iE "(malware|botnet|c2|beacon)"

# --- PERSISTANCE ---
# Jobs cron non autorisés
crontab -l && cat /etc/crontab && ls /etc/cron.d/

# Fichiers SUID inattendus
find / -perm -4000 -type f 2>/dev/null | grep -v "/usr/bin\|/usr/sbin\|/sbin\|/bin"

# Clés SSH non autorisées
find /root /home -name "authorized_keys" -exec cat {} \; 2>/dev/null

# --- LOGS D'AUTHENTIFICATION ---
# Tentatives de connexion échouées
clog /var/log/auth.log | grep "Failed\|Invalid\|refused" | tail -100

# Connexions administratives réussies
clog /var/log/auth.log | grep "Accepted" | tail -50

# Blocages SSHGuard actifs
pfctl -t sshlockout -T show
```

```
# --- VPN ---
# Sessions OpenVPN actives
cat /var/etc/openvpn/server*.log 2>/dev/null | grep "CONNECTED\|Peer Connection"

# Sessions IPsec actives
ipsec statusall | grep "ESTABLISHED"

# Sessions WireGuard
wg show 2>/dev/null

# --- INTÉGRITÉ PACKAGES ---
# Packages installés vs baseline
pkg info | awk '{print $1}' > /tmp/current_packages.txt
diff /tmp/baseline_packages.txt /tmp/current_packages.txt 2>/dev/null

# Packages avec vulnérabilités connues
pkg audit -F 2>/dev/null | grep "Affected package"

# --- CERTIFICATS ---
# Certificats expirés ou proches de l'expiration
echo | openssl s_client -connect localhost:443 2>/dev/null | openssl x509 -noout -dates

# --- PERFORMANCE (indicateurs d'attaque) ---
# Utilisation CPU/RAM
top -b -n 1 | head -20

# Table ARP (rechercher des doublons = ARP poisoning)
arp -a | sort -k1 | uniq -d -f 3
```

Escalade et notification

DÉLAI	ACTION	RESPONSABLE
T+0	Détection de l'incident, blocage immédiat de la menace	Administrateur réseau / SOC
T+15min	Notification de l'équipe sécurité et du responsable SI	SOC / RSSI
T+1h	Évaluation de l'impact et décision d'isolation	RSSI / DG
T+4h	Rapport d'incident préliminaire	RSSI
T+72h	Notification CNIL si données personnelles affectées (RGPD)	DPO / RSSI
T+7j	Rapport d'incident complet et plan de remédiation	RSSI / Équipe sécurité

Références

- CIS pfSense Firewall Benchmark v1.1.0 — Center for Internet Security, 30-06-2023
- Netgate pfSense Documentation — Documentation officielle pfSense
- Netgate Security Advisories — Avis de sécurité Netgate
- FreeBSD Security Advisories — Vulnérabilités FreeBSD (base de pfSense)
- MITRE ATT&CK — Network Devices — Tactiques et techniques pour équipements réseau
- ANSSI — Recommandations de sécurité pour les pare-feux — Guide ANSSI
- ANSSI — Recommandations relatives à IPsec — Guide algorithmes VPN
- CISA Known Exploited Vulnerabilities — Catalogue KEV CISA
- NIST SP 800-41 Rev. 1 — Guidelines on Firewalls and Firewall Policy — NIST
- CIS Controls v8 — CIS Critical Security Controls
- OpenVPN Hardening Guide — OpenVPN Community
- pfBlockerNG Documentation — Netgate pfBlockerNG
- SOCFortress pfSense Hardening Guide 2025 — SOCFortress
- Suricata IDS/IPS Documentation — Suricata Open Source IDS
- Emerging Threats Open Ruleset — ProofPoint / Emerging Threats
- Spamhaus Block Lists — DROP/EDROP IP Reputation
- Firehol Blocklist Project — Composite IP Reputation
- MaxMind GeoLite2 — Base de données GeoIP (pfBlockerNG)
- Quad9 DNS-over-TLS — Résolveur DoT sécurisé
- OpenVPN tls-crypt documentation — tls-crypt vs tls-auth
- WireGuard Official Documentation — WireGuard: Next Generation Kernel Network Tunnel (Jason A. Donenfeld)
- pfSense WireGuard Package Documentation — Netgate Docs WireGuard
- HAProxy Documentation — SSL/TLS Configuration — HAProxy SSL hardening guide
- HAProxy pfSense Package Documentation — Netgate Docs HAProxy
- SSLLabs SSL Test — Qualys SSL Labs — test noté A+ pour les configurations TLS
- Security Headers — Test des en-têtes de sécurité HTTP
- Mozilla SSL Configuration Generator — Référence des configurations TLS par niveau de sécurité
- CVE-2025-13086 — NVD NIST — OpenVPN Management Command Injection pfSense
- CISA Known Exploited Vulnerabilities — pfSense entries — CISA KEV pfSense
- OPNsense Hardening Guide — Alternative HardenedBSD-based firewall
- FreeBSD AES-NI Cryptography — Module noyau FreeBSD AES-NI
- pfSense ACME Package Documentation — Netgate Docs ACME / Let's Encrypt
- Let's Encrypt — ACME Protocol RFC 8555 — RFC 8555 Automatic Certificate Management Environment
- acme.sh — ACME Client — Script ACME shell (base du paquet pfSense ACME)
- ZeroSSL — Free SSL Certificates — CA alternative à Let's Encrypt supportée par ACME
- DNS-over-HTTPS Block Lists — Liste des résolveurs DoH publics (curl/wiki)
- IETF RFC 8484 — DNS Queries over HTTPS (DoH) — Spécification technique DoH
- pfSense-pkg-API GitHub — Paquet API REST pfSense (tiers)

- DORA — Règlement UE 2022/2554 — Digital Operational Resilience Act, effectif 17 janvier 2025 — Art.9 ICT Risk Management (durcissement firewall = exigence directe), Art.10 Business Continuity (CARP/HA), Art.11 Incident Management (syslog/SIEM)
- NIS2 Directive — Article 21 — Directive NIS2 mesures de cybersécurité
- ISO/IEC 27001:2022 — Norme internationale sécurité de l'information
- PCI DSS v4.0 Requirements — PCI Security Standards Council
- CIS Controls v8 — CIS Critical Security Controls version 8

ANNEXE — Checklists de vérification rapide

Checklists condensées pour audit terrain et conformité. Chaque ligne = un contrôle actionnable. Statuts : ✅ Conforme | ❌ Non conforme | ⚠️ Partiel | ☐ Non vérifié

Domaine 1 — Configuration initiale et mises à jour

#	CONTRÔLE	NIVEAU	STATUT
1.1	pfSense à jour (canal stable, patch critique ≤ 30 jours)	🔴 CRITIQUE	☐
1.2	Hostname et domaine configurés (différent de "pfSense" / "home.arpa")	🟢 L1	☐
1.3	IPv6 désactivé si non utilisé OU dual-stack sécurisé (parité règles IPv4/IPv6, RA/RS bloqués depuis WAN)	🟢 L1	☐
1.4	DNS Rebind Check activé avec Alternate Hostnames configurés	🟢 L1	☐
1.5	Bannière SSH configurée (`Banner /etc/issue.net`) + MOTD personnalisé	🟢 L1	☐
1.6	Intégrité du média d'installation vérifiée (SHA256 + source officielle)	🔴 CRITIQUE	☐
1.7	Édition pfSense adaptée (pfSense Plus pour production, paquets signés)	🟡 MOYEN	☐
1.8	Affichage des erreurs PHP désactivé (`display_errors = Off`)	🟡 MOYEN	☐

Domaine 2 — Gestion des utilisateurs

#	CONTRÔLE	NIVEAU	STATUT
2.1	Compte “admin” renommé ou désactivé	🔴 CRITIQUE	☐
2.2	Login Protection activée (seuil ≤30, délai ≥300s)	🟠 ÉLEVÉ	☐
2.3	Console protégée par mot de passe	🟠 ÉLEVÉ	☐
2.4	Authentification LDAP ou RADIUS configurée	🔵 L2	☐
2.5	Session timeout ≤ 10 minutes	🟢 L1	☐

Domaine 3 — Mots de passe et authentification

#	CONTRÔLE	NIVEAU	STATUT
3.1	Comptes locaux non essentiels désactivés (Local Account Status)	🔵 L2	☐
3.2	Mot de passe admin différent de “pfsense” (complexe, ≥16 car.)	🔴 CRITIQUE	☐
3.3	2FA (TOTP) activé pour chaque compte administrateur	🔵 L2	☐

Domaine 4 — Règles de pare-feu

#	CONTRÔLE	NIVEAU	STATUT
4.1	Aucune règle Allow avec Source “any” non justifiée	🟠 ÉLEVÉ	☐
4.2	Aucune règle Allow avec Destination “any” non justifiée	🟠 ÉLEVÉ	☐
4.3	Aucune règle Allow avec Service “any” sans restriction de port	🟠 ÉLEVÉ	☐
4.4	Aucune règle désactivée sans justification documentée	🟢 L1	☐
4.5	Log activé sur toutes les règles Pass et Block	🟢 L1	☐
4.6	ICMP restreint aux types nécessaires + bogons/RFC1918 bloqués sur WAN	🟡 MOYEN	☐

#	CONTRÔLE	NIVEAU	STATUT
4.7	Anti-spoofing actif sur toutes les interfaces (scrubbing + règles flottantes)	🟠 ÉLEVÉ	☐
4.8	Egress filtering actif — règle de blocage implicite explicite sur chaque interface interne	🟠 ÉLEVÉ	☐
4.9	Scrubbing/normalisation actif — case “Disable Scrub” décochée, options avancées (random-id, max-mss, no-df) configurées	🟠 ÉLEVÉ	☐

Domaine 5 — Services réseau

#	CONTRÔLE	NIVEAU	STATUT
5.1	DNSSEC activé + DNS-over-TLS configuré (Unbound ou Stubby)	🟢 L1	☐
5.2	SNMP désactivé ou SNMPv3 uniquement avec authPriv	🟠 ÉLEVÉ	☐
5.3	NTP configuré avec ≥2 serveurs fiables, fuseau horaire correct	🟢 L1	☐
5.4	UPnP et NAT-PMP désactivés	🟠 ÉLEVÉ	☐
5.5	Traffic Shaping activé — queues prioritaires management, limiters sur flux suspects	🔵 L2	☐
5.6	Proxy FTP désactivé — FTP bloqué, remplacé par SFTP/SCP	🟡 MOYEN	☐
5.7	DNS-over-HTTPS (DoH) bloqué — alias résolveurs DoH, règle Block TCP/UDP 443, DNSBL anti-DoH, redirection NAT DNS	🟠 ÉLEVÉ	☐

Domaine 6 — VPN

#	CONTRÔLE	NIVEAU	STATUT
6.1	OpenVPN : AES-256-GCM, TLS 1.3, tls-crypt (pas tls-auth), SHA-512, DH 4096 bits, compression DÉSACTIVÉE, Certificate Depth=1	🔴 CRITIQUE	☐
6.2	OpenVPN : authentification RADIUS ou LDAP configurée	🟠 ÉLEVÉ	☐

#	CONTRÔLE	NIVEAU	STATUT
6.3	IPsec : IKEv2 uniquement, pas d'algorithme faible (DES, MD5, DH<14)	🔴 CRITIQUE	☐
6.4	Certificats VPN signés par CA valide, clé ≥ 2048 bits, non expirés	🟠 ÉLEVÉ	☐
6.5	WireGuard : port non standard, PSK configuré, AllowedIPs restreint, DNS explicite, règles FW restrictives	🔵 L2	☐

Domaine 7 — Paquets et extensions

#	CONTRÔLE	NIVEAU	STATUT
7.1	pfBlockerNG installé avec réputation IP (Spamhaus/ETOpen/Firehol), GeoIP (MaxMind) et DNSBL (≥5 listes, mise à jour toutes les 4h)	🔵 L2	☐
7.2	Suricata installé en mode IPS inline (LAN), offloads matériels désactivés, règles ETOpen à jour (< 24h), Pass Lists configurées	🔵 L2	☐
7.3	Paquets non nécessaires supprimés (`pkg audit -F` sans vulnérabilité critique)	🟡 MOYEN	☐

Domaine 8 — Interface web et SSH

#	CONTRÔLE	NIVEAU	STATUT
8.1	WebGUI en HTTPS uniquement + HSTS activé + aucune exposition sur WAN	🔴 CRITIQUE	☐
8.2	SSH : authentification par clé publique uniquement (PasswordAuthentication no)	🟠 ÉLEVÉ	☐
8.3	SSH sur port non standard, accès restreint au réseau de management	🟡 MOYEN	☐
8.4	Certificat TLS WebGUI signé par CA de confiance (pas auto-signé)	🟠 ÉLEVÉ	☐
8.5		🔵 L2	☐

#	CONTRÔLE	NIVEAU	STATUT
	Accès WebGUI et SSH restreints à l'interface de management (jamais depuis WAN)		
8.6	HAProxy : TLS 1.2+ (pas TLS 1.0/1.1), DH 4096 bits, cipher ECDHE-AEAD, 7 en-têtes sécurité HTTP, SSLLabs A+	🔵 L2	☐
8.7	ACME/Let's Encrypt : paquet installé, certificat émis par CA publique, renouvellement auto (cron), actions post-renouvellement configurées	🟠 ÉLEVÉ	☐
8.8	API REST : désinstallée si non utilisée OU JWT uniquement, restriction IP, HTTPS, rate limiting, logs vers SIEM	🔵 L2	☐

Domaine 9 — Haute disponibilité, sauvegardes et matériel

#	CONTRÔLE	NIVEAU	STATUT
9.1	AutoConfigBackup activé — mot de passe chiffrement AES-256 unique (≠ admin), sauvegardes automatiques à chaque changement	🟢 L1	☐
9.2	CARP avec interface pfsync dédiée isolée, mots de passe VIP forts, NAT via VIP CARP, WebGUI jamais via VIP	🔵 L2	☐
9.3	Sauvegardes AES-256 chiffrées hors-site (≥2 emplacements), test de restauration trimestriel documenté avec RTO mesuré	🟢 L1	☐
9.4	Accélération crypto AES-NI activée si CPU compatible (`kldstat | grep aesni` + `openssl engine -t -c`)	🔵 L2	☐
9.5	Segmentation VLAN — VLAN Management, Users, Servers, IoT, Guest distincts — isolation inter-VLAN, règles FW par VLAN, tests isolation documentés	🔵 L2	☐

Domaine 10 — Journalisation et supervision

#	CONTRÔLE	NIVEAU	STATUT
10.1	Syslog distant vers SIEM, toutes catégories actives, transport TLS (port 6514)	🟢 L1	☐
10.2	Log activé sur 100% des règles de pare-feu actives	🟢 L1	☐
10.3	Taille des logs locaux ≥ 2 Mo, rétention SIEM ≥ 90 jours	🟢 L1	☐
10.4	Alertes configurées pour événements critiques (auth failure, config change, install paquet, changement CARP)	🔵 L2	☐

Tableau récapitulatif par domaine

DOMAINE	CONTRÔLES	CRITIQUE	ÉLEVÉ	MOYEN	L1	L2
D1 — Config initiale	8	2	0	2	3	0
D2 — Utilisateurs	5	1	2	0	1	1
D3 — Mots de passe	3	1	0	0	0	2
D4 — Règles pare-feu	9	0	5	2	2	0
D5 — Services réseau	7	0	3	1	2	1
D6 — VPN	5	2	2	0	0	1
D7 — Paquets	3	0	0	1	0	2
D8 — Interface/SSH	8	1	3	1	0	3
D9 — HA/Sauvegardes	5	0	0	0	2	3
D10 — Journalisation	4	0	0	0	3	1
TOTAL	**57**	**7**	**15**	**7**	**13**	**14**

Note : *Le total contrôles dans l'ANNEXE (57) correspond aux contrôles documentés dans les checklists de vérification rapide. Le compteur ITEMS=80 dans les métadonnées inclut tous les sous-contrôles et contrôles enrichis du corps du document (IPv6 dual-stack, HAProxy, Traffic Shaping, FTP proxy, pfSense Plus, options scrubbing avancées, OpenVPN compression/cert depth, forensique étendue, ACME, DoH, API REST, VLAN isolation, mapping conformité).*

Mapping MITRE ATT&CK — Contrôles couverts

TECHNIQUE MITRE	DESCRIPTION	CONTRÔLES COUVERTS
T1190	Exploit Public-Facing Application	1.1, 1.4, 4.1, 4.2, 7.3, 8.5, **8.6**
T1133	External Remote Services	6.1, 6.2, 6.3, 6.4, **6.5**, 8.3
T1110	Brute Force	2.2, 3.2, 3.3, 8.2
T1078	Valid Accounts	2.1, 2.4, 2.5, 3.1, 3.2
T1562	Impair Defenses	1.3, 4.4, 5.4, 10.2, 10.3
T1048	Exfiltration Over Alternative Protocol	4.2, 4.3, 4.6, 4.8, **5.5**
T1071	Application Layer Protocol	4.3, 5.1, 7.2, **5.6**
T1071.001	Web Protocols (HTTP C2 inspection)	4.8, 7.2
T1040	Network Sniffing	8.1, 8.4, **8.6**
T1557	Adversary-in-the-Middle	4.6, 4.7, 5.1, 6.4, 8.4, **8.6**
T1490	Inhibit System Recovery	9.1, 9.3
T1602	Data from Configuration Repository	5.2
T1195.001	Supply Chain Compromise (installation media + paquets)	1.6, **1.7**
T1036	Masquerading (IP spoofing)	4.6, 4.7
T1498	Network Denial of Service	4.6, 4.7, **5.5**
T1082	System Information Discovery (PHP errors)	1.8
T1499	Endpoint Denial of Service (CARP disruption)	9.2
T1041	Exfiltration Over C2 Channel	4.8, 7.1, 7.2, **5.5**
T1027	Obfuscated Files or Information (fragmentation)	**4.9**
T1499.002	Service Exhaustion Flood (normalisation)	**4.9**
T1552.004	Private Keys (AES-NI timing side-channel)	**9.4**

TECHNIQUE MITRE	DESCRIPTION	CONTRÔLES COUVERTS
T1562.004	Disable or Modify System Firewall	1.3, **5.6**
T1071.004	Application Layer Protocol: DNS (DoH bypass)	**5.7**
T1048.001	Exfiltration Over Encrypted Non-C2 Protocol (DoH)	**5.7**
T1553	Subvert Trust Controls (expired/invalid cert)	**8.7**
T1021	Remote Services (lateral movement, VLAN isolation)	**9.5**
T1046	Network Service Discovery (VLAN segmentation)	**9.5**
T1590	Gather Victim Network Information (VLAN)	**9.5**

Mapping de conformité — Frameworks réglementaires et normatifs

Ce tableau établit la correspondance entre les contrôles pfSense documentés dans ce benchmark et les exigences des principaux référentiels de conformité. Il facilite la production de preuves d'audit et la justification des choix de sécurité.

CONTRÔLE PFSENSE	REF. DOCUMENT	CIS CONTROLS V8	NIS2 (ART.21)	ISO 27001:2022	PCI DSS V4	RGPD ART.32	DORA (2022/2554)
Mise à jour système	1.1	CIS 7.3	✓ Mesures techniques	A.12.6.1	Req 6.3.3	✓	Art.9 ICT Risk Mgmt
Intégrité média installation	1.6	CIS 2.1	✓ Sécurité supply chain	A.12.6.2	Req 6.3.2	✓	Art.9 §2(b)
2FA administrateurs	3.3	CIS 6.3	✓ Authentification forte	A.9.4.2	Req 8.4.2	✓	Art.9 §2(c)
Règles pare-feu strictes	4.1–4.3, 4.8	CIS 12.2	✓ Contrôle accès réseau	A.13.1.1	Req 1.3	✓	Art.9 §2(a)
Blocage bogons/ RFC1918 WAN	4.6	CIS 12.2	✓ Protection périmètre	A.13.1.1	Req 1.3.2	✓	Art.9 §2(a)
	5.1	CIS 9.2		A.13.1.2		✓	Art.9 §2(d)

CONTRÔLE PFSENSE	REF. DOCUMENT	CIS CONTROLS V8	NIS2 (ART.21)	ISO 27001:2022	PCI DSS V4	RGPD ART.32	DORA (2022/2554)
DNSSEC + DNS-over-TLS			✓ Sécurité communications		Req 1.3.3		
Blocage DoH	5.7	CIS 9.2	✓ Filtrage DNS	A.13.1.2	Req 1.3.3	✓	Art.9 §2(d)
VPN chiffré IKEv2/TLS	6.1, 6.3	CIS 12.6	✓ Chiffrement en transit	A.13.2.1	Req 4.2.1	✓	Art.9 §2(d) chiffrement
pfBlockerNG / DNSBL	7.1	CIS 9.3	✓ Filtrage contenu	A.13.1.1	Req 6.4.1	✓	Art.9 §2(a)
IPS Suricata	7.2	CIS 13.4	✓ Détection intrusions	A.13.1.1	Req 6.4.1	✓	Art.9 §2(a)
WebGUI HTTPS + HSTS	8.1	CIS 3.10	✓ Sécurité interface	A.9.4.2	Req 8.6.1	✓	Art.9 §2(c)
SSH clés publiques	8.2	CIS 4.1	✓ Authentification forte	A.9.4.2	Req 8.3.6	✓	Art.9 §2(c)
Certificat TLS valide (ACME)	8.4, 8.7	CIS 3.10	✓ Gestion certificats	A.10.1.1	Req 4.2.1	✓	Art.9 §2(d)
API REST durcie	8.8	CIS 4.1	✓ Contrôle d’accès API	A.9.4.2	Req 8.6	✓	Art.9 §2(c)
CARP / HA	9.2	CIS 11.3	✓ Continuité activité	A.17.2.1	Req 12.4	✓	**Art.10 Continuité**
Sauvegardes chiffrées	9.3	CIS 11.2	✓ Sauvegarde données	A.12.3.1	Req 12.3	✓	**Art.10 §1(b) RTO/ RPO**
Segmentation VLAN	9.5	CIS 12.2	✓ Séparation réseaux	A.13.1.3	Req 1.3.2	✓	Art.9 §2(a) isolation
Syslog distant SIEM	10.1	CIS 8.2	✓ Journalisation	A.12.4.1	Req 10.2	✓	**Art.11 Incident Mgmt**
Alertes événements critiques	10.4	CIS 8.11	✓ Surveillance continue	A.12.4.1	Req 10.7	✓	**Art.11 §1(a) détection**

Légende : - **CIS Controls v8** : référence CIS Control numéro.sous-contrôle - **NIS2 Art.21** : ✓ = la mesure contribue aux obligations de cybersécurité NIS2 Article 21 (mesures appropriées pour gérer les risques) - **ISO 27001:2022** : référence annexe A (contrôles de sécurité) — numérotation 2022 - **PCI DSS v4** : référence Requirement (Req) de la version 4.0 (2022) - **RGPD Art.32** : ✓ = la mesure contribue à l’obligation de sécurité appropriée du traitement (Art.32 RGPD) - **DORA (2022/2554)** : Art.9 = ICT Risk Management Framework

(effectif 17 jan. 2025, entités financières UE) — Art.10 = Business Continuity & Backup — Art.11 = ICT-related Incident Management & Reporting. Source : https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX%3A32022R2554

> ***Avertissement :*** *Ce mapping est fourni à titre indicatif. La conformité effective aux référentiels réglementaires nécessite une évaluation formelle par un auditeur qualifié (CISSP, ISO 27001 Lead Auditor, QSA PCI DSS). Ce document est un guide technique de durcissement, non un rapport d'audit de conformité.*

RÉFÉRENCE RAPIDE — Commandes critiques pfSense

> *Aide-mémoire condensé pour les opérations courantes d'audit, de supervision et d'investigation forensique rapide. Toutes les commandes s'exécutent depuis la console pfSense (SSH ou physique).*

```
# ÉTAT GÉNÉRAL
pfctl -s info          # Statistiques pare-feu (états, compteurs, mémoire)
pfctl -sr              # Règles actives complètes
pfctl -s state | wc    # Nombre de sessions actives (valeur anormale = alerte)

# SANTÉ SYSTÈME
pkg check -s           # Intégrité des fichiers packages (FAILED = altération)
clog /var/log/system   # Logs système récents
clog /var/log/auth     # Logs authentification (SSH + WebGUI)

# RÉSEAU
sockstat -4 -l         # Ports en écoute (services actifs sur pfSense)
sockstat -4 -c         # Connexions établies (détecter C2 sortant)
arp -a                 # Table ARP (doublons = ARP poisoning)

# VPN
ipsec statusall        # État tunnels IPsec (ESTABLISHED = actif)
wg show                # État tunnels WireGuard
pfctl -t sshlockout    # IPs actuellement bloquées par SSHGuard

# INTÉGRITÉ CONFIGURATION
pfctl -sr | md5sum     # Empreinte des règles actives (comparer à la baseline)
md5 /cf/conf/config.xml  # Empreinte de la configuration (comparer à la sauvegarde)

# RÈGLES ET POLITIQUE
pfctl -vsr             # Règles avec compteurs de hits (0 hit = règle jamais utilisée)
pfctl -sn              # Règles NAT actives
pfctl -s queue         # Queues traffic shaping

# PROCESSUS ET SERVICES
ps aux | sort -k3 -rn | head -20   # Top processus par CPU
pkg audit -F | grep "Affected"     # Vulnérabilités dans les packages installés
```

Document produit par AYI NEDJIMI Consultants — https://ayinedjimi-consultants.fr Version 3.3 — Cinquième enrichissement (FINAL) : conformité DORA (Règlement UE 2022/2554, Art.9/10/11), commandes forensiques threat hunting complètes, Quick Reference Card, colonne DORA dans le mapping de conformité Basé sur le CIS pfSense Firewall Benchmark v1.1.0 (30-06-2023) et les sources SOCFortress, Emerging Threats, Netgate 2025-2026, CISA KEV Classification : CONFIDENTIEL — Ne pas diffuser hors du périmètre autorisé