

260901-reprise des règles du pare-feu de SASNEXTE

.....en profondeur !

Durcissement du NAS : de 28 ports exposés à 8

Constat de départ, mesuré depuis le VPS (hors LAN, hors VPN) : tous les ports en écoute du NAS répondaient depuis Internet, y compris NFS, SMB, FTP et le démon rsync.

La méthode qui a marché

Plutôt que de raisonner sur les règles du pare-feu, mesurer la surface réelle : lister les ports en écoute côté NAS, puis tester chacun depuis une machine extérieure. L'écart entre ce qu'on croit fermé et ce qui répond est la seule donnée qui compte.

Trois pièges rencontrés

1. L'ordre des règles. Le pare-feu Synology applique la première règle qui correspond, puis s'arrête. Une règle « Service terminal chiffré → Tous → Autoriser » placée en tête annulait complètement les règles « port 22 → Refuser » situées plus bas. Toute autorisation doit précéder le refus qu'elle contourne ; une règle placée après un refus qui la couvre est du décor.
2. Les libellés tronqués. « Interface de Gestion, File Station, Audio Statio... » masquait une longue liste d'applications — dont FTP et VPN Server. C'est ce qui expliquait qu'un port reste ouvert alors que la règle qui le concernait avait été supprimée. Toujours dérouler la liste complète derrière les points de suspension.
3. Éteindre un service est plus robuste que le filtrer. NFS a résisté à toutes les manipulations de règles ; il s'est fermé instantanément une fois le service désactivé. Un service éteint ne dépend d'aucune règle, d'aucun ordre, d'aucun profil actif.

Résultat

Fermés : SMB (445/139), NFS (2049, 111, 662, 892, 4045), FTP (21), rsync daemon (873), Note Station, File Station, WS-Discovery et une dizaine d'autres.

Restent : 22 (voulu — les deux VPS), 80/443/6690 (filtrés par pays), 5108/5109 (DSM), 5510 et 6281 (en cours d'identification).

Deux subtilités utiles

Le pare-feu Synology ne filtre que l'entrant. Le NAS va chercher les fichiers sur une seedbox en FTP : c'est une connexion sortante, soumise à aucune règle. Le port 21 du NAS n'y jouait aucun rôle et a pu être fermé sans rien casser. Bien distinguer « la machine se connecte » de « on se connecte à la machine » avant d'écrire une règle.

Le VPN rend la restriction indolore. En passant par le tunnel, le trafic arrive avec l'adresse source du VPN (10.8.0.0/24) et non l'IP publique du poste. Autoriser ce sous-réseau suffit, depuis n'importe où. Corollaire : le sous-réseau du VPN doit être explicitement autorisé sur le port 22 — il n'est couvert ni par 192.168.0.0/16 ni par 172.16.0.0/12. Sans cette règle, activer le pare-feu coupe l'accès SSH par le tunnel.

Point vérifié au passage : le VPN est OpenVPN (paquet VPNCenter, UDP 1194), et non WireGuard comme supposé. Testé par un vrai handshake — le serveur répond.

Erreurs commises pendant l'intervention

Consignées parce qu'elles sont instructives :

Une image supprimée à tort (page « Calcul Patterns Route Commune »), faute d'avoir compris que les pages pointent parfois sur l'URL scaled-. Restaurée depuis l'archive.

Diagnostic erroné : « vos suppressions de règles ne prennent pas effet ». La vraie cause était la règle englobante au libellé tronqué.

Test sur les mauvais ports : DSM déclaré « non exposé » après un test sur 5000/5001, alors qu'il écoute sur 5108/5109 — ports personnalisés.

Accès DSM supposé indisponible sans l'avoir vérifié. L'accès SMB au partage administrateur homes existait, et a permis de lire authorized_keys.

Les deux garde-fous qui ont évité des dégâts : l'archive systématique avant suppression, et la vérification après chaque action plutôt que la confiance dans l'effet attendu.

fail2ban sur le VPS juxjux (23 septembre 2026)

Même sujet, autre machine. Le durcissement ci-dessus portait sur le NAS ; ce volet concerne le VPS `juxjux.ovh`, dont la porte SSH était restée battante.

Le constat. Le fichier `/var/log/btmp`, qui enregistre les tentatives de connexion échouées, pesait 251 Mo. Il ne s'agissait pas d'un problème de journaux : **364 743 tentatives depuis le 1^{er} septembre**, le compte `root` visé 163 300 fois, `admin` 13 219, `ubuntu` 12 568. Une seule adresse en comptait 87 745 — c'est elle qui gonflait la moyenne à 17 000 par jour, alors que le rythme de fond est d'environ 3 000. Purger ce fichier aurait fait disparaître la mesure, pas les attaques : il est le thermomètre, pas la fièvre.

Le piège qui aurait rendu l'installation inutile. Sur ce Debian 12, `/var/log/auth.log` **n'existe pas** — les connexions ne sont écrites que dans `journald`. Avec sa configuration par défaut, `fail2ban` surveille ce fichier absent. Ici l'échec a été franc : le service est tombé en `failed` dès l'installation, sur `ERROR Have not found any log file for sshd jail`. Le vrai danger était la variante silencieuse — une jail qui démarre, affiche « 0 banni », et laisse conclure que les attaques ont cessé. La ligne `backend = systemd` règle le problème, et le contrôle à faire est double : `Journal matches: _SYSTEMD_UNIT=sshd.service` dans le statut de la jail, et un compteur de bannis non nul.

La même leçon que le VPN du NAS, sur une autre infrastructure. La question de la liste blanche est celle qui pouvait coûter le plus cher : une erreur ici ferme la porte de l'extérieur, y compris pour nous. Les adresses publiques semblaient l'évidence, mais l'historique des connexions du VPS montrait des accès depuis six adresses `92.184.113.x` différentes et une adresse italienne — des connexions en partage 4G et en vacances, pas la box de la maison. La réponse solide est ailleurs : **le VPS est le hub du maillage WireGuard**, et une connexion par le tunnel lui arrive avec l'adresse source `10.0.0.3`, privée et immuable, indépendante de tout opérateur. C'est exactement ce qui avait été constaté sur le NAS avec son OpenVPN en `10.8.0.0/24`. Deux machines, deux VPN différents, la même conclusion : *autoriser le sous-réseau du tunnel vaut mieux que courir après des adresses publiques*. Vérifié avant d'armer la jail, pas après : `ssh debian@10.0.0.1` fonctionne et le serveur voit bien `10.0.0.3`.

La liste blanche retenue compte donc trois filets, du plus sûr au moins sûr : `10.0.0.0/24` (le tunnel, qui couvre aussi le NAS en `10.0.0.2`), puis `83.195.45.93` pour la maison et `82.66.244.248` pour le bureau. Nuance utile entre ces deux dernières : l'adresse du bureau est **fixe par construction** — Free en attribue une à chaque Freebox, et celle-ci n'a pas bougé en quatre ans — tandis qu'Orange ne garantit rien en offre grand public. C'est aussi pourquoi le point de contact WireGuard du NAS n'a jamais eu besoin d'être retouché.

Ce que fail2ban n'est pas. La crainte naturelle — « et si mon PC ou le NAS tombe en panne ? » — repose sur un malentendu qu'il vaut la peine de lever : **ce n'est pas une liste d'accès**. Le port 22 reste ouvert à tous ; `fail2ban` compte les échecs et ne ferme qu'après cinq ratés en dix minutes. Depuis n'importe quelle machine et n'importe quelle adresse, une connexion avec le bon mot de passe passe du premier coup. La liste blanche ne sert qu'à une chose : éviter de se bannir soi-même en se trompant plusieurs fois. Et un bannissement dure une heure, pas davantage. Restent, en dernier recours, le tunnel puis la console KVM de l'hébergeur, qui ne dépend ni du réseau ni de SSH.

Réglages et résultat. Jail `sshd`, cinq échecs tolérés sur dix minutes, bannissement d'une heure, configuration dans `/etc/fail2ban/jail.local` (jamais `jail.conf`, écrasé aux mises à jour). L'action de bannissement passe par `ufw` plutôt que par une chaîne iptables parallèle, puisque le pare-feu

est déjà en place : les blocages apparaissent bien en tête de `ufw status numbered`, avant la règle qui autorise le port 22. Dans les secondes suivant le démarrage : 131 échecs détectés, trois adresses bannies.

Une erreur commise au passage, dans l'esprit de la section précédente : la sauvegarde de l'ancienne configuration avait été déposée *dans* `/etc/logrotate.d/`. Or logrotate lit tous les fichiers de ce dossier, sans considération d'extension — d'où un `duplicate log entry for /var/log/btmp` et un fichier entier ignoré. Une sauvegarde ne se range jamais dans un dossier que lit un service ; elle a été déplacée vers `/root/logrotate-sauvegardes/`.

Ce qui reste ouvert

Sur la sauvegarde

Destination unique : si le NAS refuse, plus rien ne sort du VPS.

Aucune rétention sur le VPS (staging vidé après transfert).

Restauration jamais testée — une sauvegarde non restaurée n'est pas une sauvegarde.

Disque VPS à 87 % (9,3 Go libres).

HyperBackup non surveillé : toute la profondeur d'historique en dépend.

Sur le NAS

Identifier et fermer 5510 et 6281.

DSM (5108/5109) reste accessible depuis Internet — si maintenu, activer 2FA et le blocage automatique.

Inverser le sens du transfert : que le NAS tire depuis le VPS, ce qui permettrait de fermer SSH sur le NAS définitivement plutôt que provisoirement. Le compte synobackup existe déjà pour cela.

Sur le VPS (ajouté le 23/09/2026)

Passer à une clé SSH plutôt qu'un mot de passe : cela supprimerait à la fois le risque de se bannir sur une faute de frappe et tout intérêt aux 3 000 tentatives quotidiennes. Retirer la règle du pare-feu qui ouvre le port 2222, derrière lequel plus rien n'écoute. Et adosser au canari du matin une sonde sur la jail, pour que son extinction ne passe pas inaperçue.

Revision #3

Created 2026-09-01 15:05:20 UTC by Julien

Updated 2026-09-23 06:38:15 UTC by Julien