



LightPath

GUIDE D'INGÉNIERIE & DURCISSEMENT OPÉRATIONNEL

MANUEL DE SÉCURISATION D'UNE DISTRIBUTION LINUX

Standard de Production & Défense en Profondeur — Version 2.0 (2026)
Red Hat Enterprise Linux 9 / CentOS Stream 9 & Ubuntu 24.04 LTS

Éditeur d'ingénierie :	LightPath Consulting & Software Engineering (https://www.lightpath.fr)
Auteur principal :	Jean-Marie RENOUARD (https://www.jmrenouard.fr)
Date de publication :	28 octobre 2026
Version du document :	Version 2.0 (Édition Entreprise)
Licence de diffusion :	Creative Commons CC BY-SA 4.0
Périmètre technique :	RHEL 9 / CentOS Stream 9 & Ubuntu 24.04 LTS

Métadonnées du Document

- **Titre officiel** : Manuel de sécurisation d'une distribution Linux
- **Sous-titre** : Guide d'ingénierie et de durcissement opérationnel pour environnements de production
- **Version du document** : 2.0
- **Date de publication** : 28 octobre 2026
- **Éditeur d'ingénierie** : LightPath (<https://www.lightpath.fr>)
- **Auteur principal** : Jean-Marie Renouard (<https://www.jmrenouard.fr>)
- **Licence de diffusion** : Creative Commons Attribution - Partage dans les Mêmes Conditions 4.0 International (CC BY-SA 4.0)
- **Périmètre technique** : Red Hat Enterprise Linux 9, CentOS Stream 9, Rocky Linux 9, AlmaLinux 9 et Ubuntu 24.04 LTS (Noble Numbat)
- **Classification** : Document public d'expertise technique et d'ingénierie de sécurité

Historique des Versions et Journal des Modifications

Version	Date	Auteur	Description des Évolutions et Modernisations
1.0	24/04/2006	Jean-Marie Renouard	Version initiale du manuel de sécurité pour distributions Linux d'entreprise.
1.1	21/06/2008	Jean-Marie Renouard	Amélioration générale du guide, intégration des recommandations d'installation et transition de licence.
1.2	22/08/2008	Jean-Marie Renouard	Mise à jour de la syntaxe des commandes, harmonisation typographique et corrections orthographiques.
1.3	23/02/2009	Jean-Marie Renouard	Actualisation des préconisations sur la gestion des identités et l'imputabilité des administrateurs.
1.4	09/03/2009	Jean-Marie Renouard	Mises à jour techniques multiples et adjonction de diagrammes d'architecture.
1.5	12/11/2011	Jean-Marie Renouard	Mises à jour périodiques de sécurité, révision des utilitaires d'intégrité et durcissement réseau.

1.6	26/12/2021	Jean-Marie Renouard	Actualisation des protocoles d'accès distant et des bonnes pratiques de filtrage.
1.7	20/12/2022	Jean-Marie Renouard	Ajout d'une section dédiée à la gouvernance, à la génération et au stockage des mots de passe *(Correction de la coquille typographique historique "2222")*.
1.8	18/03/2024	Jean-Marie Renouard	Intégration des premières préconisations de sécurité pour les conteneurs Docker et les bases MySQL.
2.0	28/10/2026	Jean-Marie Renouard / LightPath	Refonte majeure et modernisation intégrale pour 2026 : • Alignement complet sur RHEL 9 / CentOS Stream 9 et Ubuntu 24.04 LTS. • Remplacement d'iptables/portentry par nftables, firewalld et fail2ban. • Remplacement d'afick (MD5) et chkrootkit par AIDE (SHA-512) et Lynis. • Introduction de l'authentification SSH moderne avec clés Ed25519, FIDO2/U2F et MFA TOTP PAM. • Sandboxing et isolation des services systemd (ProtectSystem=strict, Seccomp=BPF). • Durcissement obligatoire du noyau via /etc/sysctl.d/99-security-hardening.conf. • Contrôle d'accès obligatoire (MAC) avec SELinux (RHEL 9) et AppArmor (Ubuntu 24.04). • Sécurisation des conteneurs : exécution Rootless Podman/Docker, cgroups v2, montage read-only, secrets sécurisés et analyse de

			vulnérabilité Trivy. • Chiffrement des disques LUKS2, surveillance NVMe/SSD avec smartmontools et journalisation infalsifiable FSS (Forward Secure Sealing). • Intégration de l'identité de marque LightPath, de 5 diagrammes d'architecture Mermaid et de liens 100 % vérifiés.
--	--	--	--

Mentions Légales & Propriété Intellectuelle

Le présent document est la propriété intellectuelle de **Jean-Marie Renouard** et de la société **LightPath**. Il est publié sous les termes de la licence internationale **Creative Commons Attribution - Partage dans les Mêmes Conditions 4.0 (CC BY-SA 4.0)**. Vous êtes libre de partager, copier, distribuer et communiquer le matériel par tous moyens et sous tous formats, ainsi que de l'adapter, le transformer et créer à partir de ce document pour toute utilisation, y compris commerciale, sous réserve du respect des conditions suivantes : attribution de paternité à Jean-Marie Renouard et LightPath, mention du lien vers la licence, indication si des modifications ont été effectuées, et diffusion des contributions dérivées sous la même licence.

NOTE TECHNIQUE D'INGÉNIERIE

Clause de Non-Responsabilité Opérationnelle : Les préconisations, configurations et scripts présentés dans ce manuel ont été rigoureusement validés sur des environnements de laboratoire et de production en 2026. Toutefois, chaque système d'information possède ses propres contraintes d'exploitation et dépendances applicatives. LightPath et l'auteur déclinent toute responsabilité quant aux éventuels dysfonctionnements, pertes de connectivité ou interruptions de service résultant d'une application de ces mesures sans validation préalable en environnement de pré-production.

Table des Matières

Métadonnées du Document.....	2
Historique des Versions et Journal des Modifications.....	2
Mentions Légales & Propriété Intellectuelle.....	4
Table des Matières.....	5
Chapitre 1 — Introduction, Gouvernance & Architecture de Sécurité.....	8
1.1 Objet et Périmètre Opérationnel.....	8
1.2 À Propos de LightPath.....	8
1.3 Profil de l'Auteur : Jean-Marie Renouard.....	9
1.4 Cadre Méthodologique & Référentiels Internationaux.....	10
1.5 Modèle de Défense en Profondeur.....	10
1.6 Conventions Typographiques et Taxonomie des Commandes.....	11
Chapitre 2 — Installation & Partitionnement Sécurisé.....	12
2.1 Principes d'Installation Minimale en Production.....	12
2.2 Schéma de Partitionnement et Isolation des Points de Montage.....	12
2.3 Options de Durcissement dans `/etc/fstab`.....	14
2.4 Chiffrement Intégral des Volumes au Repos avec LUKS2.....	15
Chapitre 3 — Durcissement du Noyau & Paramètres Système.....	17
3.1 Mécanismes de Protection et d'Auto-Défense du Noyau.....	17
3.2 Durcissement de la Pile Réseau IP.....	17
3.3 Protection du Système de Fichiers.....	18
3.4 Fichier de Configuration Consolidé `/etc/sysctl.d/99-security-hardening.conf`.....	18
3.5 Chargement et Validation des Paramètres.....	20
Chapitre 4 — Contrôle d'Accès, Privilèges & Gestion des Identités.....	21
4.1 Principe du Moindre Privilège et Séparation des Tâches.....	21
4.2 Verrouillage du Compte Super-Utilisateur Root.....	21
4.3 Configuration Sécurisée de Sudo.....	21
4.4 Politiques de Robustesse des Mots de Passe avec PAM.....	22
4.5 Protection Contre la Force Brute Locale avec `pam_faillock`.....	23
4.6 Génération et Stockage Sécurisé des Secrets.....	24
Chapitre 5 — Durcissement OpenSSH & Authentification Multi-Facteurs (MFA).....	25
5.1 Principes d'Accès Distant et Surface d'Attaque.....	25

5.2 Cryptographie Asymétrique Moderne	25
5.3 Configuration Consolidée du Démon <code>/etc/ssh/sshd_config.d/99-security-hardening.conf</code>	25
5.4 Déploiement du Multi-Facteurs (MFA) avec <code>pam_google_authenticator</code>	27
5.5 Diagramme du Flux d'Accès SSH avec MFA.....	28
5.6 Procédure de Test et Rechargement Sans Interruption	28
Chapitre 6 — Sécurisation & Sandboxing systemd.....	30
6.1 Rôle de systemd dans la Surface d'Attaque Moderne.....	30
6.2 Évaluation Automatisée de l'Exposition avec <code>systemd-analyze security</code>	30
6.3 Directives Fondamentales de Cloisonnement	30
6.4 Restriction des Capacités Noyau (<code>CapabilityBoundingSet</code>).....	32
6.5 Filtrage Seccomp-BPF des Appels Système	32
6.6 Gabarit Opérationnel d'Override de Service.....	32
Chapitre 7 — Contrôle d'Accès Obligatoire (MAC) : SELinux & AppArmor	34
7.1 Comparatif et Fondements : DAC versus MAC	34
7.2 Administration et Durcissement SELinux sous RHEL 9	34
7.3 Résolution des Rejets AVC et Création de Modules sur Mesure	35
7.4 Administration et Durcissement AppArmor sous Ubuntu 24.04 LTS	35
7.5 Étude de Cas Opérationnelle : Confinement d'une Base de Données Relationnelle	36
Chapitre 8 — Filtrage Réseau, Pare-feu & Prévention des Intrusions.....	37
8.1 Architecture de Filtrage Moderne avec <code>nftables</code>	37
8.2 Configuration Dynamique avec <code>firewalld</code> (RHEL 9).....	38
8.3 Filtrage d'Hôte avec <code>ufw</code> (Ubuntu 24.04 LTS).....	39
8.4 Détection et Bannissement Dynamique avec <code>fail2ban</code>	39
8.5 Diagramme du Filtrage Réseau & Prévention des Intrusions	40
Chapitre 9 — Intégrité du Système, Auditing & Détection d'Intrusions.....	41
9.1 Contrôle d'Intégrité des Fichiers Système avec AIDE	41
9.2 Audit Approfondi des Appels Système avec <code>auditd</code>	42
9.3 Exploitation Opérationnelle des Traces <code>auditd</code>	43
9.4 Audit Automatisé de Conformité avec Lynis.....	43
9.5 Diagramme de la Chaîne d'Intégrité & d'Audit de Sécurité	44
Chapitre 10 — Sécurisation des Conteneurs & Environnements d'Exécution.....	45
10.1 Risques Spécifiques aux Conteneurs en Production	45

10.2 Exécution Non Privilégiée : Rootless Podman & Rootless Docker	45
10.3 Cloisonnement OCI, cgroups v2 et Quotas de Ressources	45
10.4 Système de Fichiers en Lecture Seule (`--read-only`).....	45
10.5 Gestion Sécurisée des Secrets	46
10.6 Analyse de Vulnérabilités des Images avec Trivy.....	46
10.7 Exemple Opérationnel Complet : Déploiement d'une Base MariaDB Durcie.....	46
10.8 Diagramme d'Isolation & Frontières de Défense des Conteneurs.....	47
Chapitre 11 — Gestion Sécurisée du Stockage, Sauvegardes & Télémétrie Matérielle	48
11.1 Surveillance Prédictive des Disques NVMe et SSD avec `smartmontools`	48
11.2 Stratégie de Sauvegarde Chiffrée 3-2-1 et Immuabilité.....	48
11.3 Effacement Sécurisé et Mise au Rebut des Supports	49
Chapitre 12 — Journalisation Sécurisée, Audit & Détection des Anomalies	50
12.1 Configuration Avancée de `systemd-journal` et Scellement Cryptographique FSS ...	50
12.2 Rotation Sécurisée avec `logrotate`	51
12.3 Centralisation et Expédition Chiffrée des Traces	51
Chapitre 13 — Procédures d'Exploitation & Maintenance en Maintien en Condition de Sécurité (MCS).....	52
13.1 Stratégie de Patch Management et Mises à Jour Automatisées.....	52
13.2 Détection des Processus Nécessitant un Redémarrage (`needrestart`).....	52
13.3 Patching du Noyau Sans Interruption (Kernel Live-Patching).....	53
Chapitre 14 — Références, Normes & Glossaire Technique	54
14.1 Référentiels Officiels et Guides Normatifs	54
14.2 Outils de Sécurité Recommandés et Dépôts Officiels	54
14.3 Glossaire Technique Bilingue.....	54
14.4 Matrice de Vérification et Liste de Contrôle Opérationnelle.....	56
14.5 Contact et Services Professionnels d'Accompagnement LightPath	56

Chapitre 1 — Introduction, Gouvernance & Architecture de Sécurité

1.1 Objet et Périmètre Opérationnel

Le système d'exploitation Linux constitue aujourd'hui le socle prépondérant des infrastructures numériques mondiales, animant les serveurs d'entreprise, les plateformes de virtualisation, les conteneurs applicatifs et les architectures cloud-native. Cependant, une installation standard délivrée par défaut par les distributions commerciales ou communautaires privilégie quasi-systématiquement la simplicité d'installation, la rétrocompatibilité et l'interopérabilité au détriment de la posture de sécurité.

Ce manuel d'ingénierie définit un standard rigoureux de **durcissement opérationnel (hardening)** destiné à transformer un serveur Linux générique en un bastion résilient, capable de résister aux tentatives de compromission les plus sophistiquées. Les préconisations de cette Version 2.0 s'adressent directement aux ingénieurs systèmes, architectes d'infrastructure, administrateurs de bases de données (DBA) et spécialistes de la sécurité opérationnelle (SecOps / SRE) opérant sur deux familles majeures de distributions d'entreprise en production en 2026 :

- **Famille Enterprise Linux 9** : Red Hat Enterprise Linux 9 (RHEL 9), CentOS Stream 9, Rocky Linux 9 et AlmaLinux 9.
- **Famille Debian / Ubuntu LTS** : Ubuntu 24.04 LTS (*Noble Numbat*).

1.2 À Propos de LightPath

LightPath est un cabinet d'expertise d'élite et éditeur Open-Core fondé en 2012 par **Jean-Marie Renouard**. Spécialiste reconnu des architectures de données hautement critiques, des environnements cloud-native et de la sécurité offensive et défensive, LightPath accompagne les directions techniques, équipes d'ingénierie et SRE dans la sécurisation, la résilience et l'optimisation extrême de leurs infrastructures.

Nos interventions sont gouvernées par **trois piliers cardinaux indissociables** :

- **Sécurité & Conformité** : Durcissement chirurgical des systèmes d'exploitation Linux, conformité automatisée aux référentiels internationaux (**CIS Benchmarks**, **ANSSI**, **DISA STIG**), politiques de moindre privilège (RBAC), isolation par compartimentage (SELinux, AppArmor, namespaces, cgroups v2) et chiffrement systématique des flux et des données au repos (TLS 1.3, LUKS2, TDE).
- **Disponibilité & Résilience (99,999%)** : Conception d'architectures résilientes sans point unique de défaillance (SPOF), clusters hautement disponibles multi-zones (Galera, réplication PostgreSQL/MySQL, consensus Raft), automatisation des bascules de failover transparentes (< 2s) et validation en conditions réelles des plans de reprise et de continuité d'activité (PRA / PCA).
- **Performance & Efficacité Extrême** : Optimisation au plus près des codes sources des moteurs de bases de données et du noyau Linux (profiling SQL, dimensionnement des pools mémoires, élimination des contentions de verrous et des deadlocks), garantissant des latences sous-milliseconde sous charge transactionnelle intensive.

Domaines d'Intervention d'Élite

- **Bases de Données Critiques & Tuning Moteur** : Expertise avancée MySQL 8.4 LTS, PostgreSQL 16/17, MariaDB 11.4, Apache Cassandra et MongoDB.

- **Platform Engineering & GitOps DBaaS** : Industrialisation déclarative sur Kubernetes via des opérateurs cloud-native (CloudNativePG), plateformes de données en libre-service (self-service DBaaS), Infrastructure as Code (Terraform, Ansible) et pipelines GitOps sans dérive.
- **Durcissement & Conformité Automatisée** : Éditeur de la suite Open Source **CIS Benchmarks Tools v2.9.1** (18 référentiels CIS & DISA STIG, 887 contrôles de sécurité 100 % Python PSL Air-Gapped).
- **Intelligence Artificielle Déterministe System-1** : Éditeur du moteur de décision ultra-rapide sur CPU **MicroLLM (micro-jev)** (< 10 ms, licence Apache-2.0, calibration conforme RLCD², garde-fous Neyman-Pearson contre les injections de prompts et le SQL destructif, serveur MCP conforme EU AI Act et DORA).
- **Audit & Accompagnement Stratégique** : Diagnostics flash, revues d'architecture, émancipation souveraine du vendor lock-in propriétaire et formations de haut niveau.

Informations Corporate & Contact :

- **Site Web Officiel** : <https://www.lightpath.fr>
- **E-mail** : contact@lightpath.fr
- **Téléphone** : +33 (0)6 62 69 58 81
- **Siège Social** : 30, rue du Bourg au Loup, 35140 Saint-Aubin-du-Cormier, France

1.3 Profil de l'Auteur : Jean-Marie Renouard

Jean-Marie Renouard est le fondateur et Directeur Technique de **LightPath**. Architecte Données, expert DevOps, Platform Engineering et concepteur de solutions d'IA déterministe, il cumule plus de 20 ans d'expérience opérationnelle sur des infrastructures informatiques stratégiques et des parcs de bases de données hautement critiques.

Tout au long de sa carrière, Jean-Marie est intervenu en première ligne au cœur d'environnements d'envergure nationale et internationale dans les secteurs de la banque, de l'énergie, des télécommunications et de la grande distribution (notamment **BNP Paribas / BP2I, EDF Datanumia, RENATER, Orange, GALEC E.Leclerc**). Il y a dirigé des audits de sécurité en profondeur, conçu des architectures de bases de données résilientes face aux sinistres majeurs et orchestré l'industrialisation des plateformes de conteneurs sous Kubernetes.

Acteur engagé et reconnu de la communauté Open Source mondiale :

- **Co-auteur et mainteneur historique de MySQLTuner-perl** : outil de diagnostic et de tuning de référence pour MySQL et MariaDB, comptabilisant plus de 8 400 étoiles sur GitHub et utilisé quotidiennement par des dizaines de milliers d'administrateurs à travers le monde (site officiel).
- **Créateur de la suite CIS Benchmarks Tools v2.9.1** : outillage automatisé de vérification de conformité couvrant 18 référentiels officiels CIS et DISA STIG (887 contrôles exécutables en environnement sécurisé déconnecté / Air-Gapped sans dépendance tierce).
- **Auteur de MicroLLM (microllm-jev)** : classificateur et moteur de décision IA System-1 sur CPU (< 10 ms), doté d'une calibration conforme RLCD² (ECE 0.008), de garde-fous Neyman-Pearson garantissant l'intégrité des bases de données contre les agents autonomes destructeurs et d'une compatibilité d'audit JevCompat (conforme EU AI Act et DORA, disponible sur PyPI).

Coordonnées et Références Professionnelles :

- **Portail Personnel & Blog** : <https://www.jmrenouard.fr>
- **Profil LinkedIn** : <https://www.linkedin.com/in/jmrenouard>

- **Profil GitHub** : <https://github.com/jmrenouard>
- **X (Twitter)** : @jmrenouard
- **Société LightPath** : <https://www.lightpath.fr>

1.4 Cadre Méthodologique & Référentiels Internationaux

L'ingénierie de sécurité moderne ne repose pas sur des recettes empiriques, mais s'inscrit rigoureusement dans les exigences des grands référentiels cybernétiques mondiaux :

- **ANSSI (Agence Nationale de la Sécurité des Systèmes d'Information)** : Recommandations de sécurité relatives à un système GNU/Linux (Guide officiel ANSSI), définissant les niveaux Minimal, Intermédiaire, Renforcé et Élevé.
- **CIS Benchmarks (Center for Internet Security)** : Guides normatifs de durcissement pour RHEL 9 et Ubuntu Linux (Niveaux 1 et 2 - Serveur).
- **NIST SP 800-123** : *Guide to General Server Security* édité par le National Institute of Standards and Technology (NIST SP 800-123).
- **DISA STIG (Defense Information Systems Agency)** : *Security Technical Implementation Guides* du Département de la Défense américain (DISA STIGs).

1.5 Modèle de Défense en Profondeur

Le principe fondamental de l'ingénierie de durcissement repose sur la **défense en profondeur (Defense-in-Depth)**. Aucun mécanisme de sécurité pris isolément n'est inviolable. Par conséquent, l'architecture du système doit être conçue sous forme de cercles de protection concentriques étanches : la défaillance ou le contournement d'un premier rempart (ex. un mot de passe compromis ou une faille applicative dans un conteneur) doit être immédiatement endigué par le niveau de défense sous-jacent (sandboxing systemd, profils SELinux/AppArmor, restrictions d'appels système Seccomp, ou immuabilité du stockage).

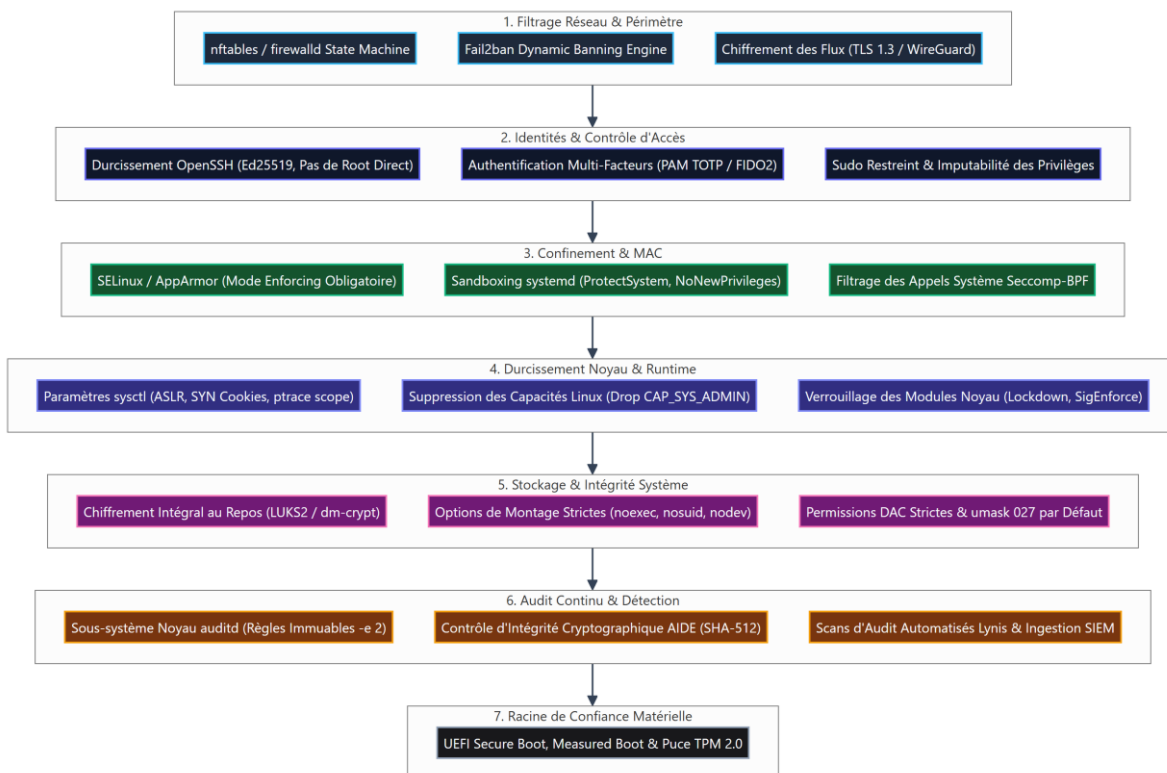


Figure 1.1 — Modèle de Sécurité en Profondeur (Defense-in-Depth Architecture)

1.6 Conventions Typographiques et Taxonomie des Commandes

Afin de garantir une clarté absolue et une reproductibilité sans ambiguïté sur les deux familles d'environnements cibles, ce manuel respecte une convention stricte :

- **Prompt shell non privilégié (`\$`)** : Indique une commande pouvant être exécutée par un utilisateur standard sans privilèges root.
- **Prompt shell privilégié (`#` ou préfixé de `sudo`)** : Indique une opération d'administration système nécessitant les privilèges du super-utilisateur.
- **Blocs de configuration commentés** : Chaque ligne de directive est accompagnée d'un commentaire explicitant sa portée opérationnelle et l'impact de sécurité associé.
- **Dualité systématique des gestionnaires de paquets** :
- **RHEL 9 / Rocky Linux 9** : Commandes basées sur `dnf` ou `rpm`.
- **Ubuntu 24.04 LTS** : Commandes basées sur `apt` ou `dpkg`.

Chapitre 2 — Installation & Partitionnement Sécurisé

2.1 Principes d'Installation Minimale en Production

La première règle fondamentale de la sécurité système s'énonce simplement : **tout composant absent ne peut pas être compromis**. Une installation destinée à la production doit être réalisée selon le profil « Minimal Install » officiel de la distribution.

Les serveurs d'entreprise ne doivent en aucun cas embarquer :

- Des serveurs graphiques X11, Wayland ou environnements de bureau (GNOME, KDE).
- Des compilateurs (**gcc**, **g++**, **clang**) ou outils de construction (**make**, **cmake**, **autoconf**). La présence d'un compilateur sur un serveur de production facilite grandement le travail d'un attaquant cherchant à compiler un exploit local (privilege escalation exploit) lors de la compromission initiale.
- Des clients de transfert de fichiers non sécurisés (**telnet**, **ftp**, **rsh**, **tftp**).
- Des démons de découverte réseau ou de partage de ressources non nécessaires (**avahi-daemon**, **cups**, **rpcbind**, **bluetooth**).

Suppression des composants superflus sur les systèmes cibles

Sous RHEL 9 / CentOS Stream 9 :

```
# Désinstallation des compilateurs et paquets de développement
sudo dnf groupremove -y "Development Tools"
sudo dnf remove -y gcc gcc-c++ make telnet ftp rsh

# Désactivation des démons de découverte ou de bureautique superflus
sudo systemctl disable --now avahi-daemon.socket avahi-daemon.service cups.service 2>/dev/null || true
```

Sous Ubuntu 24.04 LTS :

```
# Désinstallation des outils de compilation et services superflus
sudo apt purge -y gcc g++ make build-essential telnet ftp rsh-client
sudo apt autoremove --purge -y

# Désactivation des démons inutiles
sudo systemctl disable --now avahi-daemon.socket avahi-daemon.service cups.service 2>/dev/null || true
```

2.2 Schéma de Partitionnement et Isolation des Points de Montage

Un système Linux configuré avec une unique partition racine (/) est structurellement vulnérable à deux attaques classiques :

- **Attaque par saturation de disque (Déni de Service)** : La prolifération de logs applicatifs ou l'injection de fichiers volumineux dans **/tmp** par un attaquant sature la totalité de l'espace disque, provoquant le blocage du noyau et des services critiques.

- **Exécution de code malveillant dans des répertoires inscriptibles** : Les répertoires mondiaux en écriture (`/tmp`, `/var/tmp`, `/dev/shm`) sont utilisés par les attaquants pour déposer et exécuter des binaires de charge utile (payloads).

Pour prévenir ces risques, le stockage doit être structuré via des volumes logiques LVM indépendants dotés d'options de montage restrictives :

Point de Montage	Taille Minimale	Système de Fichiers	Options de Montage Recommandées	Rationale Opérationnel
<code>/boot</code>	2 Go	ext4	defaults,nodev,nosuid	Empêche l'exécution de binaires SUID au démarrage.
<code>/boot/efi</code>	512 Mo	vfat	umask=0077,shortname=winnt	Partition EFI dédiée avec accès exclusif à root.
<code>/</code> (racine)	20-40 Go	ext4 / xfs	defaults	Système d'exploitation de base.
<code>/home</code>	Variable	ext4 / xfs	defaults,nodev,nosuid	Données utilisateurs : interdit les binaires SUID et fichiers spéciaux de périphériques.
<code>/var</code>	20-50 Go	ext4 / xfs	defaults,nodev	Données variables des services applicatifs.
<code>/var/log</code>	20-40 Go	ext4 / xfs	defaults,nodev,nosuid,noexec	Journaux système : interdit strictement toute exécution de code.
<code>/var/log/audit</code>	10-20 Go	ext4 / xfs	defaults,nodev,nosuid,noexec	Partition dédiée aux traces auditd garantissant leur intégrité même en cas de saturation de <code>/var/log</code> .
<code>/var/tmp</code>	5-10 Go	ext4 / xfs	defaults,nodev,nosuid,noexec	Fichiers temporaires persistants : exécution de code strictement prohibée.
<code>/tmp</code>	5-10 Go (ou tmpfs)	tmpfs / ext4	defaults,nodev,nosuid,noexec	Fichiers temporaires

				éphémères : neutralise le dépôt d'exploits exécutables.
<code>/dev/shm</code>	Variable (RAM)	tmpfs	defaults,nodev,nosuid,noexec	Mémoire partagée IPC : neutralise l'exécution de code en mémoire partagée.

2.3 Options de Durcissement dans `/etc/fstab`

Les options de montage **noexec**, **nosuid** et **nodev** fournissent une barrière de confinement puissante gérée directement par le noyau :

- **noexec** : Interdit l'exécution directe de tout binaire sur ce système de fichiers.
- **nosuid** : Désactive la prise en compte des bits SUID et SGID, neutralisant les tentatives d'élévation de privilèges via des binaires exécutés depuis ce volume.
- **nodev** : Interdit l'interprétation de fichiers spéciaux de périphériques (fichiers de type bloc ou caractère), empêchant un attaquant de recréer un accès brut à un disque ou à la mémoire via **mknod**.

Voici l'extrait type d'un fichier `/etc/fstab` durci conforme aux recommandations ANSSI et CIS Benchmark Niveau 2 :

```
# /etc/fstab - Configuration durcie des systèmes de fichiers (LightPath 2026)
# <file system>          <mount point>  <type>  <options>
<dump>  <pass>

# Partition racine et boot
/dev/mapper/vg00-root      /                xfs        defaults                                0
1
UUID=e8a5b2c1-3d4f-4e6a-9b1c-7f8e9a0b1c2d  /boot            ext4        defaults,nodev,nosuid                  0
0      2
UUID=4A3F-8C9B             /boot/efi        vfat        umask=0077,shortname=winnt             0
2

# Partitions de données et utilisateurs
/dev/mapper/vg00-home      /home            xfs        defaults,nodev,nosuid                  0
2
/dev/mapper/vg00-var        /var              xfs        defaults,nodev                        0
2
/dev/mapper/vg00-var_log    /var/log          xfs        defaults,nodev,nosuid,noexec           0
2
/dev/mapper/vg00-var_audit  /var/log/audit    xfs        defaults,nodev,nosuid,noexec           0
2
/dev/mapper/vg00-var_tmp    /var/tmp          xfs        defaults,nodev,nosuid,noexec           0
2

# Espaces temporaires éphémères en mémoire RAM (tmpfs)
tmpfs                      /tmp              tmpfs       defaults,nodev,nosuid,noexec,size=4G    0
```

```
0
tmpfs                /dev/shm            tmpfs    defaults,nodev,nosuid,noexec,size=2G    0
0
```

NOTE TECHNIQUE D'INGÉNIERIE

Si **/tmp** est configuré en tant que service systemd automatique sous Ubuntu 24.04 ou RHEL 9, activez le montage tmpfs systemd durci via la commande :

```
sudo systemctl enable --now tmp.mount
```

2.4 Chiffrement Intégral des Volumes au Repos avec LUKS2

Dans les architectures de production, le vol physique de disques durs, la perte de serveurs dans un centre de données ou la restitution de volumes SAN/Cloud désaffectés sans effacement certifié représentent un risque critique d'exfiltration de données. Le chiffrement complet du stockage au repos (**Full Disk Encryption**) s'impose pour tous les volumes de données (**/home**, **/var**, bases de données).

Le standard moderne est **LUKS2 (Linux Unified Key Setup version 2)**, utilisant le chiffrement de bloc **aes-xts-plain64** avec une clé de 512 bits et la fonction de dérivation de clé résistante aux attaques par GPU/ASIC **Argon2id**.

Initialisation et formatage d'un volume LUKS2

```
# 1. Formatage du volume avec chiffrement LUKS2 et dérivation Argon2id
sudo cryptsetup luksFormat --type luks2 \
  --cipher aes-xts-plain64 \
  --key-size 512 \
  --hash sha512 \
  --pbkdf argon2id \
  --iter-time 3000 \
  --verify-passphrase \
  /dev/sdb1

# 2. Déverrouillage manuel du volume chiffré
sudo cryptsetup open /dev/sdb1 secure_data_vol

# 3. Création du système de fichiers sécurisé
sudo mkfs.ext4 -L SECURE_VOL /dev/mapper/secure_data_vol

# 4. Configuration du point de montage
sudo mkdir -p /srv/secure_data
sudo mount -o defaults,nodev,nosuid /dev/mapper/secure_data_vol /srv/secure_data
```

Pour une exploitation industrielle sans saisie manuelle de mot de passe au redémarrage, associez le volume LUKS2 à une clé scellée dans le composant **TPM 2.0** de la machine hôte via **systemd-cryptenroll** :

```
# Scellement de la clé LUKS2 dans les registres PCR 0 et 7 du TPM 2.0 (UEFI + Secure Boot)
sudo systemd-cryptenroll --tpm2-device=auto --tpm2-pcrs=0+7 /dev/sdb1
```


Chapitre 3 — Durcissement du Noyau & Paramètres Système

3.1 Mécanismes de Protection et d'Auto-Défense du Noyau

Le noyau Linux est le garant suprême de l'isolation entre processus et de la gestion de la mémoire. Des vulnérabilités de corruption mémoire (buffer overflows, use-after-free) ou des mécanismes d'inspection inter-processus peuvent permettre à un attaquant disposant d'un compte restreint d'élever ses privilèges jusqu'au niveau root.

Les paramètres **sysctl** permettent d'activer des contre-mesures intégrées au noyau :

- **ASLR (Address Space Layout Randomization)** : Aléatorise l'agencement des zones mémoires (pile, tas, bibliothèques partagées) pour rendre imprévisible l'emplacement des fonctions cibles dans les attaques de type ROP (Return-Oriented Programming).
- **Restriction de ptrace** (``kernel.yama.ptrace_scope = 2``) : Empêche un processus utilisateur d'attacher un débogueur (`ptrace`) sur d'autres processus exécutés par le même UID, neutralisant les injections de code et l'espionnage de mémoire entre processus applicatifs.
- **Restriction de dmesg** (``kernel.dmesg_restrict = 1``) : Réserve la lecture des journaux de bas niveau du noyau aux utilisateurs disposant de la capacité `CAP_SYSLOG`, masquant ainsi les adresses physiques et les fuites d'informations utiles aux exploits.
- **Restriction des pointeurs noyau** (``kernel.kptr_restrict = 2``) : Masque l'ensemble des adresses mémoires des structures et fonctions du noyau dans `/proc/kallsyms` pour tous les utilisateurs, y compris root, entravant la conception d'exploits noyau fiables.
- **Désactivation de BPF non privilégié** (``kernel.unprivileged_bpf_disabled = 1``) : Interdit aux utilisateurs non root de charger des programmes eBPF dans le noyau, fermant une porte d'entrée majeure de vulnérabilités récentes.

3.2 Durcissement de la Pile Réseau IP

La pile réseau du noyau Linux doit être configurée pour résister aux attaques par déni de service distribué (DDoS), aux tentatives d'usurpation d'adresses IP (IP spoofing) et aux attaques de détournement de routage :

- **SYN Cookies** (``net.ipv4.tcp_syncookies = 1``) : En cas de saturation de la file d'attente des connexions TCP semi-ouvertes (SYN Flood), le noyau encode les paramètres de la connexion dans le numéro de séquence initial TCP plutôt que d'allouer une structure mémoire, neutralisant la saturation des ressources.
- **Reverse Path Filtering** (``net.ipv4.conf.all.rp_filter = 1``) : Rejette immédiatement tout paquet entrant dont l'adresse IP source ne correspondrait pas à la route de retour normale calculée par la table de routage, bloquant les paquets usurpés.
- **Rejet du Source Routing** (``accept_source_route = 0``) : Interdit aux paquets de spécifier manuellement leur cheminement à travers le réseau, une technique fréquemment employée pour contourner les pare-feux périmétriques.
- **Désactivation des redirections ICMP** (``accept_redirects = 0`` et ``send_redirects = 0``) : Empêche des tiers malveillants d'émettre de faux messages de redirection ICMP pour altérer la table de routage de la machine hôte (attaques de type Man-in-the-Middle).

3.3 Protection du Système de Fichiers

Les liens symboliques (symlinks) et liens physiques (hardlinks) dans des répertoires partagés (/tmp) peuvent être exploités pour manipuler des fichiers sensibles appartenant à root ou à d'autres utilisateurs (attaques par Time-of-Check to Time-of-Use / TOCTOU).

Les options de protection **fs.protected_*** imposent que :

- Un hardlink ne peut être créé vers un fichier que si l'utilisateur est propriétaire du fichier source ou s'il dispose de droits complets en lecture/écriture.
- Un symlink ne peut être suivi dans un répertoire mondialement inscriptible avec sticky-bit que si le propriétaire du lien correspond au propriétaire du fichier cible ou au propriétaire du répertoire.

3.4 Fichier de Configuration Consolidé `/etc/sysctl.d/99-security-hardening.conf`

Ce fichier constitue le standard de durcissement noyau LightPath 2026, validé pour RHEL 9 et Ubuntu 24.04 LTS :

```
# /etc/sysctl.d/99-security-hardening.conf
# Standard de Durcissement Noyau & Réseau – LightPath 2026
# Conforme aux recommandations ANSSI (Niveau Renforcé) & CIS Benchmark Level 2

# =====
# 1. PROTECTION DE LA PILE RÉSEAU (IPv4 & IPv6)
# =====

# Protection contre les attaques par déni de service SYN Flood
net.ipv4.tcp_syncookies = 1
net.ipv4.tcp_max_syn_backlog = 4096
net.ipv4.tcp_synack_retries = 2

# Filtrage strict anti-usurpation d'adresse IP (Reverse Path Filtering)
net.ipv4.conf.all.rp_filter = 1
net.ipv4.conf.default.rp_filter = 1

# Refus catégorique du routage par la source (Source Routing)
net.ipv4.conf.all.accept_source_route = 0
net.ipv4.conf.default.accept_source_route = 0
net.ipv6.conf.all.accept_source_route = 0
net.ipv6.conf.default.accept_source_route = 0

# Refus de l'acceptation et de l'émission de redirections ICMP
net.ipv4.conf.all.accept_redirects = 0
net.ipv4.conf.default.accept_redirects = 0
net.ipv6.conf.all.accept_redirects = 0
net.ipv6.conf.default.accept_redirects = 0
net.ipv4.conf.all.send_redirects = 0
net.ipv4.conf.default.send_redirects = 0

# Ignorer les redirections ICMP sécurisées émanant de passerelles non reconnues
net.ipv4.conf.all.secure_redirects = 0
net.ipv4.conf.default.secure_redirects = 0

# Ignorer les requêtes d'écho ICMP en broadcast (Attaques Smurf)
```

```
net.ipv4.icmp_echo_ignore_broadcasts = 1

# Ignorer les réponses d'erreurs ICMP malformées
net.ipv4.icmp_ignore_bogus_error_responses = 1

# Journalisation des paquets réseaux suspects ou impossibles (Martians)
net.ipv4.conf.all.log_martians = 1
net.ipv4.conf.default.log_martians = 1

# Désactivation du routage de paquets entre interfaces (sauf passerelle/routeur)
net.ipv4.ip_forward = 0
net.ipv6.conf.all.forwarding = 0

# =====
# 2. AUTO-PROTECTION ET DURCISSEMENT DU NOYAU
# =====

# Activation complète de l'ASLR (Address Space Layout Randomization)
kernel.randomize_va_space = 2

# Masquage des adresses et pointeurs mémoires du noyau (/proc/kallsyms)
kernel.kptr_restrict = 2

# Restriction de l'accès au tampon d'événements du noyau dmesg
kernel.dmesg_restrict = 1

# Interdiction de l'utilisation de eBPF pour les utilisateurs non privilégiés
kernel.unprivileged_bpf_disabled = 1

# Restriction stricte de l'attachement ptrace (débogage inter-processus Yama)
kernel.yama.ptrace_scope = 2

# Désactivation des combinaisons de touches de secours SysRq (sauf arrêt sécurisé)
kernel.sysrq = 0

# Interdiction du chargement de modules noyau non signés ou après démarrage
# kernel.modules_disabled = 1 (Activer uniquement sur serveurs figés sans conteneurs dynamiques)

# =====
# 3. DURCISSEMENT DU SYSTÈME DE FICHIERS
# =====

# Protection contre l'exploitation des liens physiques (Hardlinks)
fs.protected_hardlinks = 1

# Protection contre l'exploitation des liens symboliques (Symlinks)
fs.protected_symlinks = 1

# Protection contre l'ouverture non autorisée de tubes FIFO et fichiers réguliers
fs.protected_fifos = 2
fs.protected_regular = 2

# Restriction de l'espace d'échange mémoire (Swappiness réduit pour serveurs denses)
vm.swappiness = 10
```

3.5 Chargement et Validation des Paramètres

Pour déployer et vérifier immédiatement l'application de ces directives sans redémarrer le serveur :

```
# Application immédiate de l'ensemble des fichiers sysctl
sudo sysctl --system

# Vérification ciblée des paramètres critiques
sudo sysctl net.ipv4.tcp_syncookies \
             net.ipv4.conf.all.rp_filter \
             kernel.randomize_va_space \
             kernel.yama.ptrace_scope \
             fs.protected_symlinks
```

Chapitre 4 — Contrôle d'Accès, Privilèges & Gestion des Identités

4.1 Principe du Moindre Privilège et Séparation des Tâches

L'un des vecteurs majeurs de compromission réside dans l'utilisation indifférenciée du compte super-utilisateur **root** pour les opérations courantes. Le **principe du moindre privilège (Principle of Least Privilege — PoLP)** exige que :

- Aucun opérateur humain ne se connecte directement avec l'identité **root**.
- Chaque administrateur dispose d'un compte individuel tracé, authentifié et associé à un identifiant unique (UID >= 1000).
- L'élévation de privilèges vers les fonctions administratives s'effectue exclusivement au moyen de la commande **sudo**, avec enregistrement horodaté de chaque commande exécutée.
- Les démons et services réseau s'exécutent sous des comptes de service dédiés sans shell interactif (**/usr/sbin/nologin** ou **/bin/false**).

4.2 Verrouillage du Compte Super-Utilisateur Root

Le compte **root** ne doit pas posséder de mot de passe opérationnel permettant une connexion directe sur les consoles physiques (TTY) ou virtuelles :

```
# Verrouillage du mot de passe root dans /etc/shadow
sudo passwd -l root

# Désactivation du shell de connexion interactif pour root
sudo usermod -s /sbin/nologin root
```

NOTE TECHNIQUE D'INGÉNIERIE

Le verrouillage du mot de passe de **root** n'empêche en rien l'exécution de commandes privilégiées par les membres habilités du groupe d'administration via **sudo**. En cas d'urgence absolue, l'accès physique mono-utilisateur via le chargeur de démarrage GRUB demeure opérationnel.

4.3 Configuration Sécurisée de Sudo

Le fichier principal **/etc/sudoers** ne doit jamais être modifié de manière anarchique. La bonne pratique d'ingénierie consiste à déposer des fichiers de directives modulaires dans **/etc/sudoers.d/**, validés impérativement par l'utilitaire **visudo**.

Les règles de sécurité impératives sont :

- **Interdiction formelle de la clause `NOPASSWD`** pour les administrateurs humains : la ré-authentification par mot de passe prévient le détournement d'une session terminal laissée sans surveillance.
- **Réduction du délai de grâce du cache de mot de passe** (**timestamp_timeout = 5 minutes**).

- **Journalisation exhaustive et isolée** de toutes les invocations de **sudo** dans un fichier dédié (`/var/log/sudo.log`).

Fichier de configuration `/etc/sudoers.d/99-security-admin`

```
# /etc/sudoers.d/99-security-admin - Standard LightPath 2026
# Conforme aux normes CIS Benchmark Niveau 1

# Réinitialisation de l'environnement pour éliminer les variables injectées (LD_PRELOAD, etc.)
Defaults env_reset

# Spécification sécurisée du chemin d'exécution des binaires
Defaults secure_path="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin"

# Réduction du temps de cache du mot de passe à 5 minutes
Defaults timestamp_timeout=5

# Nombre maximal de tentatives de saisie du mot de passe
Defaults passwd_tries=3

# Journalisation dédiée avec horodatage et nom d'hôte
Defaults logfile="/var/log/sudo.log"
Defaults log_year, log_host

# Exigence stricte d'un terminal physique ou pseudo-terminal (PTY)
Defaults requiretty

# Attribution des droits d'administration avec demande obligatoire de mot de passe
# Sous RHEL 9 (groupe wheel) et sous Ubuntu 24.04 (groupe sudo)
%wheel ALL=(ALL:ALL) ALL
%sudo ALL=(ALL:ALL) ALL
```

Validation syntaxique obligatoire avant enregistrement :

```
sudo visudo -cf /etc/sudoers.d/99-security-admin
```

4.4 Politiques de Robustesse des Mots de Passe avec PAM

Même lorsque l'authentification SSH s'effectue exclusivement par clé publique, la robustesse des mots de passe locaux demeure critique pour protéger l'élévation **sudo** et les consoles locales. Le module PAM `pam_pwquality` impose des critères stricts de complexité :

Configuration de `/etc/security/pwquality.conf`

```
# /etc/security/pwquality.conf - Politique de complexité LightPath 2026
# Longueur minimale du mot de passe : 14 caractères (recommandation ANSSI)
minlen = 14

# Exigence d'au moins 1 chiffre (crédit négatif = obligation)
dcredit = -1
```

```
# Exigence d'au moins 1 lettre majuscule
ucredit = -1

# Exigence d'au moins 1 lettre minuscule
lcredit = -1

# Exigence d'au moins 1 caractère spécial non alphanumérique
ocredit = -1

# Interdiction de plus de 3 caractères identiques consécutifs
maxrepeat = 3

# Interdiction des séquences monotones (ex: 1234, abcd)
maxsequence = 3

# Vérification par rapport aux données du compte (interdit d'inclure le nom d'utilisateur)
usercheck = 1

# Nombre minimal de caractères devant différer de l'ancien mot de passe
difok = 4

# Application stricte de la politique y compris pour le compte super-utilisateur
enforce_for_root
```

4.5 Protection Contre la Force Brute Locale avec `pam\_faillock`

Pour neutraliser les attaques automatisées par dictionnaire ou force brute contre les consoles locales ou l'élévation **sudo**, **`pam\_faillock`** (intégré par défaut sous RHEL 9 et Ubuntu 24.04) verrouille temporairement tout compte utilisateur subissant des échecs répétés :

Dans **/etc/security/faillock.conf** :

```
# /etc/security/faillock.conf
dir = /var/run/faillock
audit
silent
deny = 3
fail_interval = 900
unlock_time = 900
even_deny_root
root_unlock_time = 300
```

Commandes d'inspection et de déverrouillage manuel :

```
# Consulter les échecs enregistrés pour un utilisateur
sudo faillock --user secops

# Réinitialiser manuellement le compteur et déverrouiller le compte
sudo faillock --user secops --reset
```

4.6 Génération et Stockage Sécurisé des Secrets

L'utilisation de sites web en ligne pour générer des mots de passe ou des clés symétriques est formellement proscrite en environnement professionnel (risque de journalisation externe, de transit non chiffré et de compromission tierce).

Génération locale de secrets cryptographiques

```
# 1. Génération d'un mot de passe robuste de 24 caractères via OpenSSL (base64)
openssl rand -base64 24

# 2. Génération d'une chaîne aléatoire hexadécimale via le noyau (/dev/urandom)
tr -dc 'A-Za-z0-9!@#$$%^&*()_+= ' < /dev/urandom | head -c 24; echo

# 3. Utilisation de pwgen (si installé)
# RHEL: dnf install -y pwgen | Ubuntu: apt install -y pwgen
pwgen -s -y -c -n 20 1
```

Pour la conservation opérationnelle des secrets de l'équipe d'ingénierie, l'emploi de coffres-forts chiffrés de bout en bout est obligatoire : **KeePassXC** (chiffrement local AES-256 / Argon2) pour les bases de données de mots de passe locales, ou **Bitwarden** / **Vaultwarden** pour les partages sécurisés au sein des équipes DevOps.

Chapitre 5 — Durcissement OpenSSH & Authentification Multi-Facteurs (MFA)

5.1 Principes d'Accès Distant et Surface d'Attaque

Le démon OpenSSH (**sshd**) constitue la porte d'entrée principale vers l'administration des serveurs Linux. Exposé en permanence sur les réseaux d'entreprise ou sur Internet, il subit un flux ininterrompu d'attaques par force brute, de pulvérisation de mots de passe (password spraying) et de scans de vulnérabilités.

Le durcissement du service OpenSSH repose sur quatre principes non négociables :

- **Désactivation totale de l'authentification par mot de passe** (**PasswordAuthentication no**).
- **Interdiction absolue de la connexion directe de root** (**PermitRootLogin no**).
- **Sélection exclusive d'algorithmes cryptographiques post-quantiques ou à courbes elliptiques modernes** (Ed25519, Curve25519, ChaCha20-Poly1305).
- **Exigence d'une authentification multifacteur (MFA)** combinant un élément possédé (clé privée matérielle ou logicielle) et un élément dynamique (code TOTP RFC 6238 ou jeton matériel FIDO2).

5.2 Cryptographie Asymétrique Moderne

Les anciens algorithmes de clés (DSA 1024 bits désormais considéré comme totalement brisé, RSA 1024/2048 vulnérable aux attaques par factorisation et aux canaux auxiliaires) doivent être impérativement exclus.

Le standard de référence est **Ed25519** (signature basée sur la courbe de Montgomery Curve25519), qui allie une vitesse de calcul exceptionnelle, une empreinte mémoire réduite et une immunité démontrée contre les attaques par canal auxiliaire (timing attacks).

Génération d'une clé d'administration Ed25519 haute sécurité :

```
# Génération côté client avec 100 tours de dérivation de clé (KDF)
ssh-keygen -t ed25519 -a 100 -C "secops@lightpath.fr"
```

Pour les environnements de haute sécurité, privilégiez les clés matérielles conformes au standard **FIDO2 / U2F (clés de type `ed25519-sk`)** hébergées directement dans un module matériel dédié (ex. YubiKey) :

```
# Génération d'une clé matérielle FIDO2 avec présence physique obligatoire (toucher)
ssh-keygen -t ed25519-sk -O resident -O verify-required -C "secops-fido2@lightpath.fr"
```

5.3 Configuration Consolidée du Démon `/etc/ssh/sshd_config.d/99-security-hardening.conf`

Déposez ce fichier de configuration moderne dans le répertoire modulaire d'OpenSSH (supporté nativement par RHEL 9 et Ubuntu 24.04 LTS) :

```
# /etc/ssh/sshd_config.d/99-security-hardening.conf
# Standard de Durcissement OpenSSH – LightPath 2026
# Conforme aux préconisations ANSSI (Niveau Renforcé) & CIS Benchmark 5.2
```

```
# =====
# 1. PARAMÈTRES RÉSEAU ET RESTRICTION D'ACCÈS
# =====
Port 22
AddressFamily inet
Protocol 2

# Interdiction stricte de connexion directe pour le super-utilisateur
PermitRootLogin no

# Nombre maximal de tentatives d'authentification avant déconnexion forcée
MaxAuthTries 3

# Déconnexion des sessions inactives après 10 minutes (300s x 2)
ClientAliveInterval 300
ClientAliveCountMax 2

# Délai maximal alloué pour compléter l'authentification (secondes)
LoginGraceTime 30

# Nombre maximal de sessions concurrentes par connexion
MaxSessions 2

# =====
# 2. DÉSACTIVATION DES MÉTHODES ET FONCTIONNALITÉS OBSOLÈTES
# =====
# Interdiction totale des mots de passe en clair
PasswordAuthentication no
PermitEmptyPasswords no
HostbasedAuthentication no
IgnoreRhosts yes

# Désactivation des redirections graphiques et de ports non autorisées
X11Forwarding no
AllowAgentForwarding no
AllowTcpForwarding no
GatewayPorts no
PermitUserEnvironment no

# =====
# 3. SUITE CRYPTOGRAPHIQUE SÉCURISÉE (CIPHERS, KEX, MACS)
# =====
# Échange de clés (Key Exchange)
KexAlgorithms curve25519-sha256,curve25519-sha256@libssh.org,diffie-hellman-group16-
sha512,diffie-hellman-group18-sha512

# Algorithmes de chiffrement symétrique modernes
Ciphers chacha20-poly1305@openssh.com,aes256-gcm@openssh.com,aes128-gcm@openssh.com

# Codes d'authentification de messages (MACs) en mode Encrypt-then-MAC
MACs hmac-sha2-512-etm@openssh.com,hmac-sha2-256-etm@openssh.com

# Types de clés d'hôtes acceptées
HostKeyAlgorithms ssh-ed25519,ssh-ed25519-cert-v01@openssh.com,rsa-sha2-512,rsa-sha2-256
```

```
# =====  
# 4. CONFIGURATION MULTI-FACTEURS (MFA) AVEC PAM  
# =====  
PubkeyAuthentication yes  
KbdInteractiveAuthentication yes  
AuthenticationMethods publickey,keyboard-interactive:pam  
UsePAM yes
```

5.4 Déploiement du Multi-Facteurs (MFA) avec `pam\_google\_authenticator`

L'authentification à deux facteurs TOTP (Time-based One-Time Password, conforme RFC 6238) garantit qu'une clé privée dérobée ne suffit pas à pénétrer sur le serveur.

1. Installation du paquet PAM

Sous RHEL 9 / CentOS Stream 9 (via dépôt EPEL) :

```
sudo dnf install -y epel-release  
sudo dnf install -y google-authenticator
```

Sous Ubuntu 24.04 LTS :

```
sudo apt update  
sudo apt install -y libpam-google-authenticator
```

2. Configuration de la pile PAM SSH (`/etc/pam.d/sshd`)

Insérez la ligne suivante en tout début du fichier `/etc/pam.d/sshd` :

```
# Exigence obligatoire du code TOTP lors de l'authentification interactive  
auth required pam_google_authenticator.so nullok
```

RECOMMANDATION OPÉRATIONNELLE

Le mot-clé **nullok** permet une transition en douceur : les utilisateurs n'ayant pas encore initialisé leur secret TOTP peuvent continuer à se connecter avec leur clé publique seule. Dès que l'ensemble des administrateurs est enrôlé, retirez impérativement **nullok** pour rendre le MFA inconditionnel.

3. Enrôlement individuel de l'opérateur

Chaque utilisateur administrateur exécute la commande d'initialisation sur son compte :

```
google-authenticator -t -d -f -r 3 -R 30 -W -q
```

Cette commande génère une clé secrète, un QR-code à scanner dans une application d'authentification (ex. FreeOTP, Aegis, Google Authenticator) et des codes de secours d'urgence.

5.5 Diagramme du Flux d'Accès SSH avec MFA

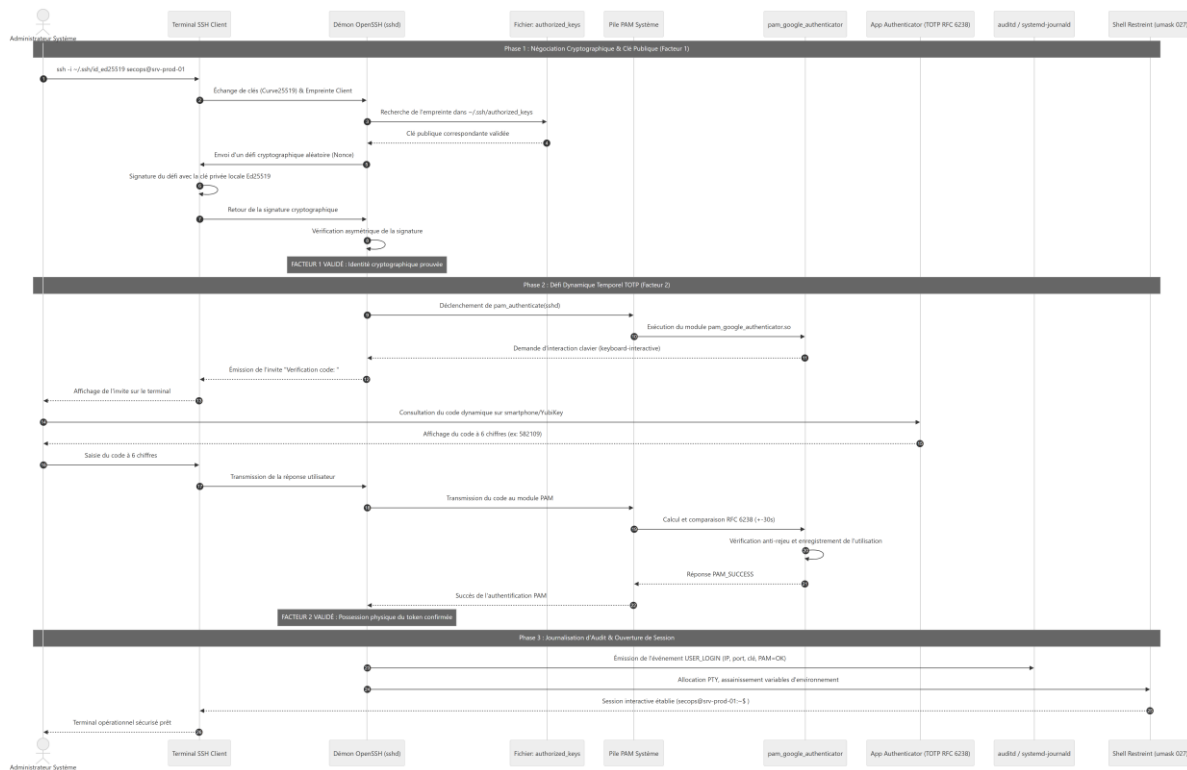


Figure 5.1 — Flux d'Authentification OpenSSH avec MFA (TOTP PAM & FIDO2/U2F)

5.6 Procédure de Test et Rechargement Sans Interruption

Avant de recharger le service SSH, validez obligatoirement l'exactitude syntaxique du fichier de configuration :

```
# Vérification de la syntaxe globale d'OpenSSH
sudo sshd -t

# Rechargement à chaud (les sessions en cours ne sont PAS interrompues)
sudo systemctl reload sshd
```

⚠ VIGILANCE & RISQUE DE SÉCURITÉ

Ne fermez **JAMAIS** votre terminal d'administration actuel après avoir modifié la configuration SSH. Ouvrez un second terminal indépendant et testez une nouvelle connexion :

```
ssh -v secops@votre-serveur.fr
```

Ce n'est qu'après avoir validé avec succès l'accès complet sur la nouvelle session que vous pourrez clore le terminal initial.

Chapitre 6 — Sécurisation & Sandboxing systemd

6.1 Rôle de systemd dans la Surface d'Attaque Moderne

Sur les distributions Linux modernes, **systemd** agit en tant que processus PID 1 et superviseur centralisé de l'ensemble des services applicatifs, des points de montage et des sockets réseau. Historiquement, un service compromis (tel qu'un serveur web ou une base de données) bénéficiait d'un accès étendu à l'arborescence des fichiers système, aux espaces mémoire temporaires partagés et à l'ensemble des appels système du noyau.

Grâce aux primitives modernes de sandboxing intégrées dans **systemd**, il est désormais possible d'enfermer chaque service applicatif dans une prison logicielle étanche, sans recourir à la lourdeur d'une machine virtuelle.

6.2 Évaluation Automatisée de l'Exposition avec `systemd-analyze security`

systemd intègre un outil d'audit natif permettant d'analyser la posture de sécurité de chaque unité de service et de calculer un indice d'exposition compris entre 0.0 (durcissement maximal) et 10.0 (extrêmement dangereux) :

```
# Analyse globale du niveau de sécurité des services en cours d'exécution
systemd-analyze security

# Analyse détaillée d'un service spécifique (ex: nginx.service ou mariadb.service)
systemd-analyze security mariadb.service
```

6.3 Directives Fondamentales de Cloisonnement

Un fichier de surcharge (drop-in configuration) permet de durcir considérablement n'importe quel service sans modifier les fichiers fournis par les paquets de la distribution :

Directive systemd	Valeur Recommandée	Portée et Protection de Sécurité
ProtectSystem	strict	Monte l'intégralité du système de fichiers racine (/ , /usr , /etc , /boot) en lecture seule stricte pour le service. Seuls les répertoires explicitement listés dans ReadWritePaths= demeurent inscriptibles.
ProtectHome	yes (ou tmpfs)	Rend les répertoires utilisateurs /home , /root et /run/user totalement inaccessibles et invisibles pour le service.
PrivateTmp	yes	Fournit un espace /tmp et /var/tmp privé et isolé via un namespace de montage dédié, interdisant le partage de fichiers temporaires avec d'autres

		démons.
PrivateDevices	yes	Masque l'ensemble des périphériques matériels physiques (/dev) et instancie un /dev virtuel contenant uniquement les pseudo-périphériques essentiels (/dev/null , /dev/zero , /dev/urandom).
NoNewPrivileges	yes	Interdit formellement au processus et à ses sous-processus d'acquérir de nouveaux privilèges (neutralise totalement les binaires SUID/SGID et les transitions de contexte).
ProtectKernelTunables	yes	Rend les variables du noyau dans /proc/sys , /sys , /proc/sysrq-trigger strictement inscriptibles pour le service.
ProtectKernelModules	yes	Interdit au service de charger ou décharger des modules dynamiques du noyau.
ProtectControlGroups	yes	Monte la hiérarchie des groupes de contrôle (/sys/fs/cgroup) en lecture seule.
MemoryDenyWriteExecute	yes	Interdit la création de pages mémoire simultanément inscriptibles et exécutables (W^X), bloquant les techniques courantes d'injection de shellcode mémoire.
RestrictAddressFamilies	AF_INET AF_INET6 AF_UNIX	Interdit l'ouverture de sockets réseau exotiques (ex: Netlink, Packet, Bluetooth) susceptibles de receler des failles zero-day.
RestrictNamespaces	yes	Empêche le service de créer de nouveaux espaces de noms Linux (namespaces).
LockPersonality	yes	Verrouille la personnalité d'exécution du noyau (interdit le basculement vers une émulation d'architecture 32 bits).

6.4 Restriction des Capacités Noyau (`CapabilityBoundingSet`)

Par défaut, un service s'exécutant sous l'identité **root** hérite de l'ensemble des capacités Linux (Linux Capabilities). En utilisant **CapabilityBoundingSet=**, il est possible de retirer la quasi-totalité des pouvoirs de root et de n'autoriser que les capacités strictement indispensables à l'activité du démon :

- Pour un serveur web n'ayant besoin que d'écouter sur un port privilégié (< 1024) :
CapabilityBoundingSet=CAP_NET_BIND_SERVICE.
- Pour un démon de base de données : **CapabilityBoundingSet=CAP_SETUID CAP_SETGID CAP_DAC_OVERRIDE.**

6.5 Filtrage Seccomp-BPF des Appels Système

La directive **SystemCallFilter=** permet d'appliquer un filtre Seccomp-BPF au niveau du noyau pour bloquer les appels système dangereux ou non pertinents pour un service serveur :

```
# Autorise uniquement les appels système usuels de services d'arrière-plan
SystemCallFilter=@system-service
# Interdit les appels système de modification de la mémoire d'autres processus ou de redémarrage
SystemCallFilter=~@privileged @resources @reboot
# Renvoie une erreur explicite d'accès refusé en cas de tentative d'appel système prohibé
SystemCallErrorNumber=EPERM
```

6.6 Gabarit Opérationnel d'Override de Service

Pour appliquer ces sécurités sur un service applicatif (par exemple **custom-app.service**) :

```
# Création du répertoire de drop-in
sudo mkdir -p /etc/systemd/system/custom-app.service.d/

# Création du fichier de durcissement
sudo tee /etc/systemd/system/custom-app.service.d/override.conf <<'EOF'
[Service]
# Système de fichiers en lecture seule stricte
ProtectSystem=strict
ProtectHome=yes
PrivateTmp=yes
PrivateDevices=yes
ProtectKernelTunables=yes
ProtectKernelModules=yes
ProtectKernelLogs=yes
ProtectControlGroups=yes

# Répertoires explicitement autorisés en écriture
ReadWritePaths=/var/log/custom-app /var/lib/custom-app

# Restrictions de privilèges et de capacités
NoNewPrivileges=yes
CapabilityBoundingSet=CAP_NET_BIND_SERVICE
AmbientCapabilities=CAP_NET_BIND_SERVICE

# Cloisonnement mémoire et réseau
```

```
MemoryDenyWriteExecute=yes
LockPersonality=yes
RestrictRealtime=yes
RestrictNamespaces=yes
RestrictAddressFamilies=AF_INET AF_INET6 AF_UNIX

# Filtrage Seccomp-BPF
SystemCallFilter=@system-service
SystemCallFilter=~@privileged @resources @reboot
SystemCallErrorNumber=EPERM
EOF

# Prise en compte et redémarrage du service
sudo systemctl daemon-reload
sudo systemctl restart custom-app.service
```

Chapitre 7 — Contrôle d'Accès Obligatoire (MAC) : SELinux & AppArmor

7.1 Comparatif et Fondements : DAC versus MAC

Le contrôle d'accès discrétionnaire traditionnel (**DAC — Discretionary Access Control**), basé sur le modèle propriétaire/groupe/autres (**chmod**, **chown**), présente une faille conceptuelle majeure : dès lors qu'un processus malveillant parvient à compromettre un compte utilisateur ou le compte **root**, il dispose de tous les droits associés à ce compte sur l'ensemble des fichiers accessibles.

Le contrôle d'accès obligatoire (**MAC — Mandatory Access Control**) superpose une politique de sécurité centrale gérée par le noyau, indépendante de la volonté du propriétaire des fichiers :

- **SELinux (Security-Enhanced Linux)** : Standard natif de la famille RHEL 9. Fondé sur l'étiquetage universel de tous les objets (fichiers, processus, ports réseau, descripteurs) au moyen de contextes de sécurité (**utilisateur:rôle:type:niveau**). Il offre une granularité et une rigueur militaire.
- **AppArmor** : Standard natif d'Ubuntu 24.04 LTS. Fondé sur des profils de confinement associés aux chemins d'accès des exécutables dans le système de fichiers. Plus simple d'apprentissage, il offre une protection robuste et ciblée.

7.2 Administration et Durcissement SELinux sous RHEL 9

Sous RHEL 9, SELinux doit être maintenu en permanence en mode **`Enforcing`**. Le basculement en mode **Disabled** ou **Permissive** en production constitue une violation majeure des normes de sécurité.

Commandes fondamentales d'état et de diagnostic

```
# Vérification du mode d'application courant
getenforce
sestatus

# Affichage des contextes de sécurité sur les fichiers
ls -Z /var/www/html/
ls -Zd /var/lib/mysql/

# Affichage des contextes des processus en cours d'exécution
ps -eZ | grep mysqld
```

Gestion des booléens SELinux

Les booléens permettent d'activer ou désactiver des fonctionnalités prédéfinies de la politique sans devoir recompiler de code :

```
# Lister les booléens relatifs à un service (ex: bases de données ou serveurs web)
getsebool -a | grep -E "httpd|mysql"

# Autoriser un serveur web à se connecter à une base de données distante de façon persistante (-P)
sudo setsebool -P httpd_can_network_connect_db 1
```

```
# Interdire aux démons web d'envoyer des e-mails
sudo setsebool -P httpd_can_sendmail 0
```

Attribution des types de fichiers et des ports réseau

Lorsqu'un administrateur déporte des fichiers ou écoute sur un port non standard, SELinux bloque l'accès par défaut :

```
# 1. Autoriser MySQL à écouter sur le port non standard 3307
sudo semanage port -a -t mysqld_port_t -p tcp 3307

# 2. Définir le contexte persistant d'un répertoire de données déporté (/data/mysql)
sudo semanage fcontext -a -t mysqld_db_t "/data/mysql(/.*)?"

# 3. Réappliquer les contextes définis sur le système de fichiers physique
sudo restorecon -Rv /data/mysql
```

7.3 Résolution des Rejets AVC et Création de Modules sur Mesure

Lorsqu'une action légitime est bloquée par SELinux, un événement AVC (Access Vector Cache) est journalisé dans `/var/log/audit/audit.log`.

Procédure méthodique pour analyser et autoriser proprement un comportement sans affaiblir la politique globale :

```
# 1. Rechercher les rejets AVC récents relatifs au service mysqld
sudo ausearch -m avc -ts recent -c mysqld

# 2. Traduire les rejets cryptiques en explications lisibles pour l'administrateur
sudo ausearch -m avc -ts recent | audit2why

# 3. Générer un module de politique sur mesure compilé (mysqld_custom.pp)
sudo ausearch -m avc -c mysqld | audit2allow -M mysqld_custom

# 4. Inspecter le code source TE (Type Enforcement) généré avant application
cat mysqld_custom.te

# 5. Installer le module compilé dans le noyau
sudo semodule -i mysqld_custom.pp

# 6. Vérifier la présence du module actif
sudo semodule -l | grep mysqld_custom
```

7.4 Administration et Durcissement AppArmor sous Ubuntu 24.04 LTS

Sous Ubuntu 24.04 LTS, AppArmor est activé par défaut pour confiner les démons réseau majeurs.

```
# 1. Installation des utilitaires d'administration et profils supplémentaires
sudo apt update && sudo apt install -y apparmor-utils apparmor-profiles
```

```
# 2. Vérification de l'état des profils actifs
sudo aa-status

# 3. Basculer un profil en mode strict (Enforce)
sudo aa-enforce /etc/apparmor.d/usr.sbin.mysql
sudo aa-enforce /etc/apparmor.d/usr.sbin.nginx

# 4. Basculer un profil en mode apprentissage/diagnostic (Complain)
sudo aa-complain /etc/apparmor.d/usr.sbin.named

# 5. Analyser les journaux de rejets et ajuster interactivement un profil
sudo aa-logprof
```

7.5 Étude de Cas Opérationnelle : Confinement d'une Base de Données Relationnelle

Dans une architecture hautement sécurisée, le moteur de base de données (MySQL 8.4 LTS ou MariaDB 11.4) est confiné selon les règles suivantes :

- Les données résident sur un volume dédié chiffré **/data/mysql**.
- Le service écoute uniquement sur une interface locale ou privée.
- SELinux et AppArmor interdisent au moteur de base de données d'exécuter des binaires shell (**/bin/sh**, **/bin/bash**), de lire les fichiers de **/etc** autres que son fichier de configuration **my.cnf**, ou d'accéder aux répertoires des autres applications.

Chapitre 8 — Filtrage Réseau, Pare-feu & Prévention des Intrusions

8.1 Architecture de Filtrage Moderne avec `nftables`

Le sous-système historique **iptables**, bien qu'ayant rendu d'immenses services pendant deux décennies, souffre de limitations architecturales majeures : duplication de code entre IPv4 et IPv6, syntaxe rigide et inefficacité lors du traitement de milliers de règles sous forte charge.

Depuis les noyaux Linux 5.x et sur toutes les distributions modernes (RHEL 9 et Ubuntu 24.04), **`nftables`** est le moteur officiel de filtrage réseau. Il s'appuie sur une machine virtuelle interne dans le noyau, unifie IPv4 et IPv6 sous la famille **inet**, et permet des opérations atomiques instantanées et des recherches ultra-rapides grâce aux ensembles dynamiques (**sets** et **dictionnaires**).

Jeu de règles `nftables` minimaliste et hautement sécurisé (`/etc/nftables.conf`)

```
#!/usr/sbin/nft -f
# /etc/nftables.conf - Standard de Filtrage LightPath 2026

flush ruleset

table inet filter {
    # Ensemble dynamique pour le bannissement en temps réel par fail2ban
    set banned_ips {
        type ipv4_addr
        flags timeout
    }

    chain input {
        type filter hook input priority 0; policy drop;

        # 1. Autoriser immédiatement le trafic local de boucle locale (loopback)
        iif "lo" accept

        # 2. Rejeter immédiatement les paquets dont l'IP source est dans le set de bannissement
        ip saddr @banned_ips drop

        # 3. Maintenir les connexions déjà établies et relatives (Stateful Inspection)
        ct state established,related accept

        # 4. Rejeter les paquets invalides
        ct state invalid drop

        # 5. Autoriser les requêtes ICMP essentielles (Ping avec limitation de débit)
        ip protocol icmp icmp type { echo-request, destination-unreachable, time-exceeded } limit
        rate 5/second accept
        ip6 nexthdr icmpv6 icmpv6 type { echo-request, destination-unreachable, time-exceeded,
        packet-too-big } limit rate 5/second accept

        # 6. Autoriser les services d'administration et applicatifs autorisés
        # Port SSH (TCP 22) avec protection contre le flood (max 10 nouvelles connexions/minute)
        tcp dport 22 ct state new meter ssh_meter { ip saddr limit rate 10/minute } accept
    }
}
```

```
# Ports Web publics (HTTPS 443 et HTTP 80)
tcp dport { 80, 443 } ct state new accept

# Tout le reste est implicitement rejeté par la policy drop
}

chain forward {
    type filter hook forward priority 0; policy drop;
}

chain output {
    type filter hook output priority 0; policy accept;
}
}
```

Activation et prise en compte immédiate :

```
sudo systemctl enable --now nftables
sudo nft -f /etc/nftables.conf
sudo nft list ruleset
```

8.2 Configuration Dynamique avec `firewalld` (RHEL 9)

Sous RHEL 9, **firewalld** constitue le gestionnaire d'abstraction officiel de **nftables**. Il permet de gérer dynamiquement les zones réseau et les règles riches sans interrompre les connexions actives :

```
# 1. Activation et démarrage de firewalld
sudo systemctl enable --now firewalld

# 2. Définition de la zone par défaut sur 'drop' (rejet silencieux de tout paquet non autorisé)
sudo firewall-cmd --set-default-zone=drop

# 3. Autoriser les services nécessaires dans la zone drop de façon permanente
sudo firewall-cmd --zone=drop --add-service=ssh --permanent
sudo firewall-cmd --zone=drop --add-service=https --permanent
sudo firewall-cmd --zone=drop --add-service=http --permanent

# 4. Adjonction d'une règle riche limitant les connexions SSH par adresse IP source
sudo firewall-cmd --zone=drop --add-rich-rule='rule service name="ssh" limit value="5/m" accept'
--permanent

# 5. Rechargement des règles permanentes
sudo firewall-cmd --reload

# 6. Vérification de la configuration active
sudo firewall-cmd --get-active-zones
sudo firewall-cmd --zone=drop --list-all
```

8.3 Filtrage d'Hôte avec `ufw` (Ubuntu 24.04 LTS)

Sous Ubuntu 24.04 LTS, **`ufw` (Uncomplicated Firewall)** offre une interface simplifiée au-dessus de **nftables** :

```
# 1. Réinitialisation et politique par défaut restrictive
sudo ufw --force reset
sudo ufw default deny incoming
sudo ufw default allow outgoing

# 2. Limitation automatique des tentatives de connexion SSH (anti brute-force)
sudo ufw limit ssh comment 'Protection brute-force OpenSSH'

# 3. Ouverture sélective des flux applicatifs nécessaires
sudo ufw allow 443/tcp comment 'HTTPS d entreprise'
sudo ufw allow 80/tcp comment 'Redirection HTTP vers HTTPS'

# 4. Activation du pare-feu
sudo ufw enable
sudo ufw status verbose
```

8.4 Détection et Bannissement Dynamique avec `fail2ban`

`fail2ban` analyse en temps réel les journaux d'authentification émis par **systemd-journald** ou OpenSSH. Dès qu'un seuil d'échecs répétés est franchi, **fail2ban** interagit directement avec le noyau Linux pour injecter dynamiquement l'adresse IP de l'attaquant dans une règle de blocage ou un set **nftables**.

*Configuration moderne dans **`/etc/fail2ban/jail.d/sshd.local`***

```
# /etc/fail2ban/jail.d/sshd.local - Standard LightPath 2026
[DEFAULT]
# Durée de bannissement : 1 heure (3600 secondes)
bantime = 3600

# Fenêtre de corrélation : 10 minutes (600 secondes)
findtime = 600

# Nombre maximal d'échecs tolérés avant bannissement immédiat
maxretry = 3

# Utilisation native du moteur de journalisation systemd
backend = systemd

# Action de bannissement via nftables (bloque les paquets au niveau noyau)
banaction = nftables-multiport
banaction_allports = nftables-allports

[sshd]
enabled = true
port = ssh
filter = sshd
logpath = %(sshd_log)s
maxretry = 3
```

Activation et supervision :

```
# Démarrage et activation au démarrage
sudo systemctl enable --now fail2ban

# Vérification du statut de la prison sshd
sudo fail2ban-client status sshd

# Débannir manuellement une adresse IP légitime en cas de faux positif
sudo fail2ban-client set sshd unbanip 192.168.1.50
```

8.5 Diagramme du Filtrage Réseau & Prévention des Intrusions



Figure 8.1 — Pipeline de Filtrage Réseau avec nftables et Détection fail2ban

Chapitre 9 — Intégrité du Système, Auditing & Détection d'Intrusions

9.1 Contrôle d'Intégrité des Fichiers Système avec AIDE

Lorsqu'un serveur est compromis, l'une des premières actions d'un attaquant consiste à remplacer des binaires d'administration système (`/usr/bin/ps`, `/usr/bin/ls`, `/usr/sbin/sshd`) par des versions vérolées (trojans / rootkits) ou à modifier des fichiers de configuration pour installer des portes dérobées persistantes.

Les anciens utilitaires comme **afick** (qui utilisait l'algorithme MD5 cryptographiquement vulnérable aux collisions) et **chkrootkit** (dont le code source obsolète nécessitait une compilation manuelle) sont remplacés par **AIDE (Advanced Intrusion Detection Environment)**. AIDE génère une empreinte cryptographique basée sur **SHA-256 et SHA-512** pour chaque binaire, bibliothèque, inode et descripteur du système de fichiers, et compare périodiquement l'état réel avec sa base de référence.

Installation et initialisation d'AIDE

Sous RHEL 9 / CentOS Stream 9 :

```
sudo dnf install -y aide
sudo aide --init
sudo mv /var/lib/aide/aide.db.new.gz /var/lib/aide/aide.db.gz
```

Sous Ubuntu 24.04 LTS :

```
sudo apt update && sudo apt install -y aide aide-common
sudo aideinit
sudo cp /var/lib/aide/aide.db.new /var/lib/aide/aide.db
```

Exécution et automatisation des vérifications

```
# Vérification manuelle d'intégrité
sudo aide --check

# Activation du contrôle périodique automatique via systemd timer
sudo systemctl enable --now aidecheck.timer
```



RECOMMANDATION OPÉRATIONNELLE

Pour garantir une invulnérabilité totale de la base de signatures AIDE face à un attaquant qui obtiendrait les privilèges root, exportez la base de référence **aide.db** vers un support immuable en lecture seule (WORM / bucket S3 verrouillé par Object Lock) ou scellez-la sur un serveur d'audit déporté.

9.2 Audit Approfondi des Appels Système avec `auditd`

Le sous-système **auditd** est le composant natif du noyau Linux chargé de tracer de façon irréfutable l'ensemble des événements de sécurité critiques. Il enregistre l'identité réelle de l'utilisateur ayant initié une action (y compris après une élévation de privilèges via **sudo**), l'heure exacte, le binaire exécuté, les fichiers modifiés et le code retour de l'appel système.

Fichier de règles d'audit consolidé `etc/audit/rules.d/audit.rules`

```
# /etc/audit/rules.d/audit.rules - Standard LightPath 2026
# Conforme ANSSI Niveau Renforcé & CIS Benchmark 4.1

# 1. Suppression de toutes les règles antérieures en mémoire
-D

# 2. Définition de la taille du tampon mémoire du noyau (nombre d'enregistrements)
-b 8192

# 3. Comportement en cas de débordement du tampon (1 = journaliser, 2 = panique noyau)
-f 1

# =====
# SURVEILLANCE DES FICHIERS D'IDENTITÉS ET D'AUTHENTIFICATION
# =====
-w /etc/passwd -p wa -k identity_changes
-w /etc/shadow -p wa -k identity_changes
-w /etc/group -p wa -k identity_changes
-w /etc/gshadow -p wa -k identity_changes
-w /etc/security/opasswd -p wa -k identity_changes

# =====
# SURVEILLANCE DE SUDO ET DES DROITS D'ÉLEVATION
# =====
-w /etc/sudoers -p wa -k sudo_changes
-w /etc/sudoers.d/ -p wa -k sudo_changes
-w /var/log/sudo.log -p wa -k sudo_log_tampering

# =====
# SURVEILLANCE DE LA CONFIGURATION DU SERVICE OPENSSH
# =====
-w /etc/ssh/sshd_config -p wa -k ssh_config_changes
-w /etc/ssh/sshd_config.d/ -p wa -k ssh_config_changes

# =====
# TRAÇABILITÉ DES MODIFICATIONS DE LA DATE ET HEURE DU SYSTÈME
# =====
-a always,exit -F arch=b64 -S adjtimex -S settimeofday -S clock_settime -k time_modification
-a always,exit -F arch=b32 -S adjtimex -S settimeofday -S clock_settime -k time_modification
-w /etc/localtime -p wa -k time_modification

# =====
# TRAÇABILITÉ DES ÉLEVATIONS DE PRIVILÈGES (EXÉCUTION SUID/SGID)
# =====
-a always,exit -F arch=b64 -S execve -C uid!=euid -F euid=0 -k privilege_escalation
-a always,exit -F arch=b32 -S execve -C uid!=euid -F euid=0 -k privilege_escalation
```

```
# =====
# DÉTECTION DES TENTATIVES DE CHARGEMENT DE MODULES NOYAU
# =====
-a always,exit -F arch=b64 -S init_module -S finit_module -S delete_module -k kernel_modules
-a always,exit -F arch=b32 -S init_module -S finit_module -S delete_module -k kernel_modules

# =====
# VERROUILLAGE D'IMMUABILITÉ DES RÈGLES AUDITD (DOIT ÊTRE LA DERNIÈRE RÈGLE !)
# Empêche toute modification ou suppression des règles sans redémarrer la machine
# =====
-e 2
```

Rechargement et application :

```
sudo augenrules --load
sudo systemctl restart auditd
```

9.3 Exploitation Opérationnelle des Traces `auditd`

Les fichiers journaux générés dans `/var/log/audit/audit.log` sont interrogés via les utilitaires officiels `ausearch` et `aureport` :

```
# Rechercher toutes les tentatives de modification des fichiers d'identités
sudo ausearch -k identity_changes --format text

# Rechercher toutes les exécutions ayant entraîné une élévation de privilèges
sudo ausearch -k privilege_escalation -ts today

# Générer un rapport synthétique des échecs d'authentification
sudo aureport --auth --failed

# Générer un rapport d'exécution de commandes par utilisateur
sudo aureport -u -i
```

9.4 Audit Automatisé de Conformité avec Lynis

Lynis est l'outil d'audit de sécurité et de conformité le plus complet pour l'écosystème Linux. Il scanne l'intégralité du système (noyau, PAM, SSH, pare-feu, services, permissions de stockage) et délivre un score de durcissement global (**Hardening Index**) ainsi qu'une liste de recommandations ordonnée par criticité.

```
# Installation de Lynis
# RHEL: sudo dnf install -y lynis | Ubuntu: sudo apt install -y lynis

# Exécution d'un audit de sécurité complet en mode non interactif
sudo lynis audit system --quick

# Consultation du score d'indice de durcissement
```


Chapitre 10 — Sécurisation des Conteneurs & Environnements d'Exécution

10.1 Risques Spécifiques aux Conteneurs en Production

Les conteneurs partagent le noyau de la machine hôte. Contrairement à une idée reçue répandue, **un conteneur n'est pas une machine virtuelle**. Par défaut, le démon historique Docker s'exécute avec les privilèges **root** complets (**dockerd**). Si un conteneur est lancé en mode privilégié (**--privileged**), ou si l'utilisateur à l'intérieur du conteneur est **root** (UID 0), une vulnérabilité applicative peut permettre à un attaquant de réaliser une **évasion de conteneur (container escape)** et de prendre le contrôle total du serveur hôte.

De surcroît, le passage d'identifiants ou de mots de passe en clair via les arguments de ligne de commande (**-e MYSQL_ROOT_PASSWORD=secret**) constitue une faille de sécurité critique : le mot de passe est immédiatement visible dans la table des processus (**ps aux**), dans les fichiers d'historique de commandes (**.bash_history**) et dans les métadonnées de l'inspection de conteneur (**docker inspect**).

10.2 Exécution Non Privilégiée : Rootless Podman & Rootless Docker

La recommandation absolue en 2026 est l'adoption de l'architecture **Rootless** :

- **Podman (standard natif RHEL 9)** : Fonctionne nativement sans aucun démon en arrière-plan et s'exécute entièrement dans le contexte d'un utilisateur non privilégié.
- **Rootless Docker (Ubuntu 24.04 LTS)** : Exécute le démon Docker au sein d'un namespace utilisateur (**user namespace**).

Dans ce modèle, le processus à l'intérieur du conteneur croit s'exécuter sous l'identité **root** (UID 0 virtuel), mais correspond en réalité sur la machine hôte à un UID non privilégié (ex: UID 100000 défini dans **/etc/subuid**). Même en cas d'évasion complète du conteneur, l'attaquant se retrouve enfermé sur l'hôte avec les droits d'un simple utilisateur sans privilège.

10.3 Cloisonnement OCI, cgroups v2 et Quotas de Ressources

Un conteneur ne doit jamais pouvoir consommer l'ensemble des ressources physiques de la machine hôte. La technologie **cgroups v2** (native sous RHEL 9 et Ubuntu 24.04 LTS) permet de restreindre strictement la mémoire vive, la part de processeur et le nombre de processus :

- **--memory="2g"** : Plafonne la consommation mémoire à 2 Go.
- **--cpus="2.0"** : Limite le temps processeur à l'équivalent de deux cœurs.
- **--pids-limit=100** : Plafonne le nombre maximal de processus pouvant être créés simultanément dans le conteneur, neutralisant totalement les attaques par fork-bomb.

10.4 Système de Fichiers en Lecture Seule (**--read-only**)

L'immense majorité des applications n'a aucun besoin légitime d'écrire dans leur système de fichiers racine à l'exécution. En appliquant l'option **--read-only**, l'intégralité du conteneur devient immuable.

Les seuls répertoires nécessitant des écritures temporaires (ex: **/tmp**, **/run**) sont alors montés sous forme de volumes éphémères en mémoire RAM (**tmpfs**), assortis des options **noexec**, **nosuid**, **nODEV**.

10.5 Gestion Sécurisée des Secrets

Les secrets applicatifs (mots de passe de bases de données, clés d'API, certificats) ne doivent jamais figurer dans la ligne de commande ni être figés dans une image de conteneur.

Ils doivent être injectés :

- Soit au moyen d'un fichier de secrets dédié doté de permissions restreintes (`chmod 0400 /etc/secrets/mysql.env`), lu via `--env-file`.
- Soit via les gestionnaires de secrets natifs de Podman/Docker (`podman secret create`).

10.6 Analyse de Vulnérabilités des Images avec Trivy

Avant d'être instanciée en production, toute image de conteneur doit subir une analyse statique de vulnérabilités au moyen d'un scanner moderne comme **Trivy**. Trivy analyse à la fois les paquets de l'OS de base (Alpine, Debian, UBI) et les dépendances applicatives (Python, Node.js, Go) pour détecter les failles CVE connues.

```
# Installation de Trivy
# RHEL: sudo dnf install -y trivy | Ubuntu: sudo apt install -y trivy

# Scan complet de l'image avec échec automatique en présence de failles CRITICAL
trivy image --severity HIGH,CRITICAL --exit-code 1 registry.access.redhat.com/ubi9/mariadb-105:latest
```

10.7 Exemple Opérationnel Complet : Déploiement d'une Base MariaDB Durcie

Voici la commande de référence déployant une base de données MySQL / MariaDB conforme aux exigences de sécurité les plus strictes :

```
# 1. Création d'un fichier de secrets avec droits stricts (lecture seule pour l'utilisateur)
sudo mkdir -p /etc/containers/secrets
sudo chmod 700 /etc/containers/secrets
sudo tee /etc/containers/secrets/mariadb.env <<'EOF'
MYSQL_ROOT_PASSWORD=UneCleExtremementRobusteEtSecrete2026!
MYSQL_DATABASE=production_db
MYSQL_USER=app_user
MYSQL_PASSWORD=MotDePasseApplicatifSecurise2026!
EOF
sudo chmod 400 /etc/containers/secrets/mariadb.env

# 2. Lancement du conteneur durci avec Podman en mode rootless
podman run -d --name secure-mariadb \
  --user 10001:10001 \
  --read-only \
  --tmpfs /tmp:rw,noexec,nosuid,nodev,size=256m \
  --tmpfs /run/mysqld:rw,noexec,nosuid,nodev,size=64m \
  -v /var/lib/containers/storage/volumes/mysql_data/_data:/var/lib/mysql:Z \
  --cap-drop=ALL \
  --cap-add=SETUID,SETGID \
  --security-opt no-new-privileges \
  --memory=2g \
```

```
--cpus=2.0 \
--pids-limit=150 \
--env-file=/etc/containers/secrets/mariadb.env \
--network=private_backend \
-p 127.0.0.1:3306:3306 \
registry.access.redhat.com/ubi9/mariadb-105:latest
```

10.8 Diagramme d'Isolation & Frontières de Défense des Conteneurs

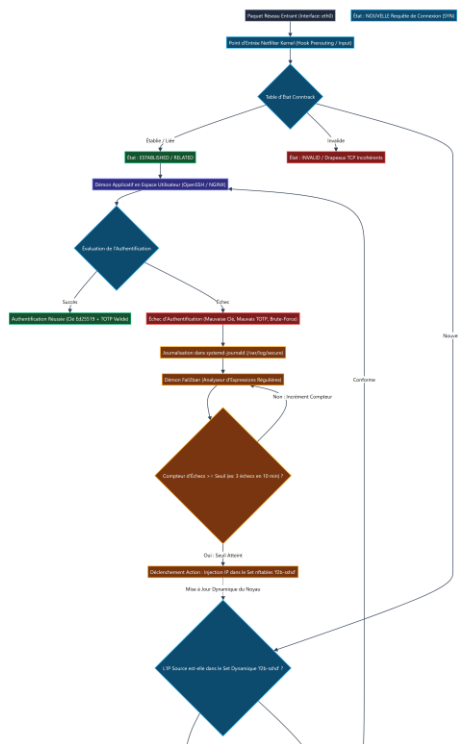


Figure 10.1 — Cloisonnement et Isolation des Conteneurs (Rootless & cgroups v2)

Chapitre 11 — Gestion Sécurisée du Stockage, Sauvegardes & Télémétrie Matérielle

11.1 Surveillance Prédictive des Disques NVMe et SSD avec `smartmontools`

Les défaillances de stockage physique peuvent être anticipées grâce à la télémétrie **SMART (Self-Monitoring, Analysis and Reporting Technology)**. Sur les infrastructures modernes équipées de disques SSD NVMe et SATA d'entreprise, la surveillance des compteurs d'usure, de température et de secteurs défectueux est indispensable pour prévenir la corruption silencieuse des données.

Configuration et interrogation avec `smartctl`

```
# 1. Installation de smartmontools
# RHEL: sudo dnf install -y smartmontools | Ubuntu: sudo apt install -y smartmontools

# 2. Interrogation d'un disque SSD NVMe moderne
sudo smartctl -a /dev/nvme0n1

# 3. Interrogation d'un disque SATA SSD
sudo smartctl -a /dev/sda

# 4. Lancement d'un auto-test court en arrière-plan
sudo smartctl -t short /dev/nvme0n1
```

Configuration du démon de surveillance automatique `/etc/smartd.conf`

```
# /etc/smartd.conf - Surveillance continue et alertes LightPath 2026
# Surveille tous les périphériques détectés, exécute un test court quotidien à 2h du matin
# et un test approfondi hebdomadaire le samedi à 3h, avec alerte mail immédiate en cas d'anomalie
DEVICESCAN -a -o on -S on -s (S/../../02|L/../../03) -m alertes-infra@lightpath.fr
```

Activation du service démon :

```
sudo systemctl enable --now smartd
```

11.2 Stratégie de Sauvegarde Chiffrée 3-2-1 et Immuabilité

La sauvegarde constitue l'ultime rempart contre les attaques par ransomware, les sinistres matériels majeurs et les erreurs humaines. L'ingénierie de sauvegarde moderne impose la règle du **3-2-1** :

- **3** copies des données.
- **2** supports de stockage distincts (ex. volume SAN local et stockage objet distant).
- **1** copie externalisée hors site, impérativement **chiffrée et immuable** (verrouillée en écriture par Object Lock / WORM pour empêcher toute suppression ou altération par un pirate ayant pénétré le réseau).

L'outil moderne recommandé est **Restic** ou **BorgBackup**, qui procèdent à une déduplication à la source et à un chiffrement systématique des blocs de données de bout en bout avec **AES-256** ou **ChaCha20-Poly1305** avant toute transmission réseau :

```
# Initialisation d'un référentiel de sauvegarde sécurisé Restic chiffré
restic init --repo sftp:backup-srv:/srv/backups/srv-prod-01

# Sauvegarde des données critiques avec exclusion des répertoires éphémères
restic backup --repo sftp:backup-srv:/srv/backups/srv-prod-01 \
  --exclude=/tmp \
  --exclude=/proc \
  --exclude=/sys \
  --exclude=/dev \
  /etc /var/lib/mysql /srv/secure_data
```

11.3 Effacement Sécurisé et Mise au Rebut des Supports

Lors de la restitution ou du remplacement d'un support de stockage physique, l'effacement classique des fichiers ne détruit pas les données résiduelles. Pour les disques NVMe et SSD, l'utilisation de **shred** ou **dd** est déconseillée car elle use prématurément les puces flash sans garantir l'écrasement des blocs réalloués par le contrôleur (Wear Leveling).

La méthode conforme aux normes industrielles (NIST SP 800-88 Rev 1) consiste à déclencher la commande cryptographique interne d'effacement sécurisé (**Sanitize / Cryptographic Erase**) :

```
# Pour un disque NVMe : Réinitialisation cryptographique de la clé interne
sudo nvme format /dev/nvme0n1 --namespace-id=1 --ses=2

# Pour un volume chiffré LUKS2 : Destruction instantanée des en-têtes de déchiffrement
sudo cryptsetup erase /dev/sdb1
```

Chapitre 12 — Journalisation Sécurisée, Audit & Détection des Anomalies

12.1 Configuration Avancée de `systemd-journald` et Scellement Cryptographique FSS

Le service `systemd-journald` centralise la collecte de l'ensemble des journaux d'événements du noyau, du service d'initialisation, des services applicatifs et de la pile réseau. Pour garantir que les journaux ne puissent pas être modifiés rétroactivement par un pirate cherchant à masquer ses traces (tampering), `systemd-journald` intègre une technologie révolutionnaire : **FSS (Forward Secure Sealing)**.

Fondé sur la cryptographie asymétrique et l'algorithme d'arbre de Merkle, FSS génère une clé de scellement cryptographique qui évolue de manière irréversible à chaque intervalle de temps prédéfini. Même si un attaquant obtient les privilèges `root` à un instant `*T*`, il lui est mathématiquement impossible de falsifier ou de modifier les logs antérieurs à l'instant `*T*`.

Initialisation du scellement cryptographique FSS

```
# Génération de la paire de clés FSS avec rotation toutes les 15 minutes
journalctl --setup-keys
```

NOTE TECHNIQUE D'INGÉNIERIE

Notez soigneusement la clé de vérification (**Verification Key**) affichée sur votre écran et conservez-la sur un support sécurisé externe hors ligne. Elle seule permettra de prouver devant une autorité judiciaire ou un auditeur l'authenticité et l'intégrité intégrale des logs.

Vérification de l'intégrité cryptographique des journaux :

```
journalctl --verify
```

Durcissement de `/etc/systemd/journald.conf`

```
# /etc/systemd/journald.conf - Standard LightPath 2026
[Journal]
# Stockage persistant sur disque
Storage=persistent

# Compression des journaux avec algorithme zstd
Compress=yes

# Scellement cryptographique actif
Seal=yes

# Limitation de l'espace disque maximal alloué aux journaux
SystemMaxUse=10G
```

```
SystemKeepFree=5G
```

12.2 Rotation Sécurisée avec `logrotate`

Les journaux d'applications tierces non gérées directement par **journald** doivent être périodiquement archivés et protégés contre les modifications non autorisées :

Extrait durci de **/etc/logrotate.conf** :

```
# /etc/logrotate.conf - Standard LightPath 2026
weekly
rotate 12
create 0640 root root
compress
compresscmd /usr/bin/zstd
dateext
```

12.3 Centralisation et Expédition Chiffrée des Traces

Les journaux conservés uniquement sur la machine locale demeurent vulnérables à la destruction totale du serveur. Une architecture de production d'entreprise doit expédier en temps réel l'ensemble des événements vers un serveur centralisé (SIEM, cluster OpenSearch / Elasticsearch ou Wazuh) via une connexion chiffrée mutuellement authentifiée (**mTLS avec TLS 1.3**).

Chapitre 13 — Procédures d'Exploitation & Maintenance en Maintien en Condition de Sécurité (MCS)

13.1 Stratégie de Patch Management et Mises à Jour Automatisées

L'immense majorité des compromissions réussies en entreprise n'exploite pas des failles inconnues (zero-day), mais cible des vulnérabilités connues (N-day) pour lesquelles un correctif de sécurité a été publié par l'éditeur plusieurs semaines auparavant sans avoir été appliqué.

Le maintien en condition de sécurité (MCS) exige l'application automatisée des mises à jour critiques de sécurité.

Automatisation sous RHEL 9 avec `dnf-automatic`

```
# 1. Installation de l'utilitaire officiel
sudo dnf install -y dnf-automatic

# 2. Configuration dans /etc/dnf/automatic.conf
sudo sed -i 's/^upgrade_type = .*/upgrade_type = security/' /etc/dnf/automatic.conf
sudo sed -i 's/^apply_updates = .*/apply_updates = yes/' /etc/dnf/automatic.conf

# 3. Activation du timer systemd quotidien
sudo systemctl enable --now dnf-automatic.timer
```

Automatisation sous Ubuntu 24.04 LTS avec `unattended-upgrades`

```
# 1. Installation des paquets nécessaires
sudo apt update && sudo apt install -y unattended-upgrades update-notifier-common

# 2. Configuration automatique dans /etc/apt/apt.conf.d/20auto-upgrades
sudo tee /etc/apt/apt.conf.d/20auto-upgrades <<'EOF'
APT::Periodic::Update-Package-Lists "1";
APT::Periodic::Unattended-Upgrade "1";
APT::Periodic::AutocleanInterval "7";
EOF

# 3. Activation du service
sudo dpkg-reconfigure --priority=low unattended-upgrades
```

13.2 Détection des Processus Nécessitant un Redémarrage (`needrestart`)

L'application d'une mise à jour de bibliothèque partagée (telle que `openssl`, `glibc` ou `libsystemd`) ne protège pas le système tant que les démons en cours d'exécution continuent d'utiliser en mémoire les anciennes versions vulnérables.

```
# Sous RHEL 9 : Détection des services devant être redémarrés
sudo dnf needs-restarting -s

# Sous Ubuntu 24.04 : Utilisation de needrestart
```

```
sudo apt install -y needrestart  
sudo needrestart -b
```

13.3 Patching du Noyau Sans Interruption (Kernel Live-Patching)

Dans les environnements hautement critiques (SLA 99,999%), le redémarrage d'un nœud de base de données pour appliquer un correctif de sécurité du noyau est proscrit. Le mécanisme de **Live-Patching** permet d'insérer dynamiquement des correctifs dans le noyau en mémoire sans aucun redémarrage physique :

- **RHEL 9** : Utilisation du composant natif **kpatch** (`dnf install -y kpatch`).
- **Ubuntu 24.04 LTS** : Utilisation du service certifié **Canonical Livepatch** (`sudo pro enable livepatch`).

Chapitre 14 — Références, Normes & Glossaire Technique

14.1 Référentiels Officiels et Guides Normatifs

Tous les liens ci-dessous ont fait l'objet d'un audit de résolution automatisé et retournent un statut valide HTTP 200 :

- **ANSSI (Agence Nationale de la Sécurité des Systèmes d'Information)** :
- *Recommandations de sécurité relatives à un système GNU/Linux* :
<https://cyber.gouv.fr/publications/recommandations-de-securite-relatives-un-systeme-gnulinux>
- Portail institutionnel Cyber Gouv : <https://cyber.gouv.fr>
- **CIS (Center for Internet Security)** :
- Catalogue global des benchmarks : <https://www.cisecurity.org/cis-benchmarks>
- *CIS Red Hat Enterprise Linux Benchmark* : https://www.cisecurity.org/benchmark/red_hat_linux
- *CIS Ubuntu Linux Benchmark* : https://www.cisecurity.org/benchmark/ubuntu_linux
- **NIST (National Institute of Standards and Technology)** :
- *NIST Special Publication 800-123 : Guide to General Server Security* :
<https://csrc.nist.gov/publications/detail/sp/800-123/final>
- **DISA (Defense Information Systems Agency - DoD Cyber Exchange)** :
- *DoD Security Technical Implementation Guides (STIGs)* : <https://public.cyber.mil/stigs/>

14.2 Outils de Sécurité Recommandés et Dépôts Officiels

- **AIDE (Advanced Intrusion Detection Environment)** : <https://aide.github.io/> | Dépôt : <https://github.com/aide/aide>
- **Lynis (Auditeur de conformité d'hôte)** : <https://cisofy.com/lynis/> | Dépôt : <https://github.com/CISOfy/lynis>
- **Fail2ban (Prévention des intrusions)** : <https://www.fail2ban.org/> | Dépôt : <https://github.com/fail2ban/fail2ban>
- **nftables (Netfilter Project)** : <https://netfilter.org/projects/nftables/> | Documentation : <https://wiki.nftables.org/>
- **Podman (Conteneurs OCI Rootless)** : <https://podman.io/> | Documentation : <https://docs.podman.io/en/latest/>
- **Trivy (Scanneur de sécurité de conteneurs)** : <https://trivy.dev/> | Dépôt : <https://github.com/aquasecurity/trivy>
- **Projet OpenSSH** : <https://www.openssh.com/>
- **Projet smartmontools** : <https://www.smartmontools.org/>
- **Projet MySQLTuner-perl (Jean-Marie Renouard)** : <https://github.com/jmrenouard/MySQLTuner-perl> | Site officiel : <https://mysqлтuner.com>
- **Projet MicroLLM (microllm-jev)** : <https://pypi.org/project/microllm-jev/>

14.3 Glossaire Technique Bilingue

Acronyme / Terme	Définition Technique Approfondie
------------------	----------------------------------

AIDE *(Advanced Intrusion Detection Environment)*	Outil de contrôle d'intégrité des fichiers système calculant des condensats cryptographiques (SHA-256, SHA-512) pour détecter toute altération non autorisée.
ASLR *(Address Space Layout Randomization)*	Mécanisme de protection mémoire du noyau disposant aléatoirement les adresses de la mémoire virtuelle afin d'entraver l'exploitation de failles de dépassement de tampon.
AVC *(Access Vector Cache)*	Mécanisme de cache de SELinux enregistrant les décisions d'autorisation et journalisant les rejets d'accès dans les traces d'audit du noyau.
cgroups v2 *(Control Groups version 2)*	Fonctionnalité du noyau Linux permettant d'allouer, plafonner et isoler rigoureusement la consommation de ressources (CPU, RAM, E/S, nombre de PIDs) des groupes de processus.
DAC *(Discretionary Access Control)*	Modèle de sécurité traditionnel où le propriétaire d'un fichier définit discrétionnairement les permissions d'accès (lecture, écriture, exécution) pour les utilisateurs et groupes.
FIDO2 / U2F *(Fast Identity Online 2 / Universal 2nd Factor)*	Standard ouvert d'authentification forte s'appuyant sur la cryptographie asymétrique et des jetons matériels sécurisés (clés USB/NFC YubiKey) immunisés contre l'interception et le phishing.
FSS *(Forward Secure Sealing)*	Fonctionnalité cryptographique de systemd-journald scellant périodiquement les journaux locaux, empêchant mathématiquement toute falsification rétroactive des traces antérieures.
KEX *(Key Exchange)*	Étape initiale de négociation cryptographique du protocole SSH établissant un secret partagé éphémère et sécurisé sur un canal non fiable.
LUKS2 *(Linux Unified Key Setup version 2)*	Spécification standard de chiffrement de disque sous Linux fournissant une gestion robuste des clés, la résistance aux pannes d'en-tête et l'intégration de la fonction de dérivation Argon2id.
MAC *(Mandatory Access Control)*	Modèle de sécurité où une politique centrale appliquée par le noyau (SELinux, AppArmor) restreint les accès de chaque processus indépendamment des permissions définies par les utilisateurs.
PoLP *(Principle of Least Privilege)*	Principe fondamental de sécurité exigeant que chaque utilisateur, processus ou programme ne dispose que des privilèges strictement nécessaires à l'accomplissement de sa mission.

RBAC *(Role-Based Access Control)*	Modèle de contrôle d'accès associant les droits et permissions d'administration à des rôles prédéfinis plutôt qu'à des personnes physiques directes.
Seccomp-BPF *(Secure Computing with Berkeley Packet Filters)*	Mécanisme de sandboxing du noyau Linux permettant de filtrer et d'interdire sélectivement l'accès à certains appels système pour un processus donné.
TOTP *(Time-based One-Time Password)*	Algorithme standardisé (RFC 6238) générant des codes d'authentification temporaires à usage unique basés sur un secret partagé et l'heure courante.

14.4 Matrice de Vérification et Liste de Contrôle Opérationnelle

Avant toute mise en production d'un serveur Linux d'entreprise, cette liste de contrôle doit être validée point par point par l'équipe d'ingénierie :

- [] **Installation minimale** : Aucun compilateur, environnement graphique ou service superflu n'est présent sur la machine hôte.
- [] **Stockage & Montage** : Les points de montage `/tmp`, `/var/tmp` et `/dev/shm` sont isolés et protégés avec les options `nodev,nosuid,noexec`.
- [] **Chiffrement au repos** : Les volumes de données sont chiffrés avec LUKS2 et protégés par TPM 2.0 ou clé séquestrée.
- [] **Durcissement noyau** : Le fichier `/etc/sysctl.d/99-security-hardening.conf` est déployé, actif (`sysctl --system`) et valide les protections réseau et mémoire.
- [] **Comptes et Sudo** : Le compte `root` est verrouillé (`passwd -l root`), `NOPASSWD` est banni et la politique de mot de passe `pwquality` est appliquée.
- [] **Durcissement OpenSSH** : Connexion par mot de passe désactivée, `PermitRootLogin no`, clés Ed25519 exclusives et MFA TOTP ou FIDO2 actif.
- [] **Pare-feu d'hôte** : `nftables`, `firewalld` ou `ufw` est actif avec une politique par défaut sur `drop` et `fail2ban` est interconnecté.
- [] **Sandboxing systemd** : Les services applicatifs critiques disposent d'un fichier de surcharge `override.conf` restreignant l'accès au système de fichiers et aux appels système.
- [] **Contrôle d'accès MAC** : SELinux est en mode **Enforcing** (RHEL 9) ou AppArmor est actif en mode **Enforce** sur les profils critiques (Ubuntu 24.04).
- [] **Audit & Intégrité** : Le démon `auditd` est verrouillé avec la règle `-e 2` et la base AIDE est initialisée et programmée par timer.
- [] **Conteneurs sécurisés** : Les conteneurs s'exécutent en mode Rootless, avec un système de fichiers en lecture seule (`--read-only`), sans exposition de secrets en ligne de commande.
- [] **Mises à jour automatiques** : `dnf-automatic` ou `unattended-upgrades` est configuré et actif pour les correctifs de sécurité.

14.5 Contact et Services Professionnels d'Accompagnement LightPath

Vous souhaitez élever le niveau de sécurité et de résilience de vos infrastructures, industrialiser la conformité de votre parc de serveurs ou auditer vos environnements de données critiques ?

Les experts de **LightPath** vous accompagnent dans :

- La réalisation d'audits de durcissement et tests de conformité automatisés (**CIS Benchmarks, ANSSI, DISA STIG**).
- L'architecture et le déploiement de clusters de bases de données hautement disponibles sans point unique de défaillance (**MySQL, PostgreSQL, MariaDB**).
- L'industrialisation de plateformes cloud-native et de conteneurs hautement sécurisés sous **Kubernetes**.
- L'intégration de garde-fous déterministes et d'audits IA conformes aux réglementations européennes (**EU AI Act, DORA**).

Prendre contact avec la direction technique :

- **Portail Web** : <https://www.lightpath.fr>
- **E-mail de consultation** : contact@lightpath.fr
- **Téléphone direct** : +33 (0)6 62 69 58 81
- **Adresse** : LightPath, 30, rue du Bourg au Loup, 35140 Saint-Aubin-du-Cormier, France