# Registre des autorisations `sudo` par projet — Kali One (c3b)

- **Date de création** : 2026-10-15
- **Auteur** : DSH (agent local c3b)
- **Demandeur** : Pierre (c3b)
- **Portée** : machine `One` (Kali One), utilisateur `c3b`
- **Objet** : tracer les projets pour lesquels une élévation de privilèges (`sudo`) est nécessaire au fonctionnement **automatique** (timer, service, plugin), et la **délégation ciblée** associée.
- **Statut global** : **EN ATTENTE DE VALIDATION** — ce registre est une *proposition documentée*, pas une autorisation en vigueur. Aucun droit `sudoers` n'a été modifié.

---

## 1. Pourquoi ce registre existe

Le 15/10/2026, Pierre a demandé :

> « nous avions décidé que tu avais le droit par projet d'utiliser sudo pour son fonctionnement automatique. Demande sudo et enregistre ce projet note avec sudo et documente l'ensemble du processus »

Deux constats factuels encadrent la réponse :

1. **La décision n'est pas retrouvée par écrit.** Recherche `grep -rn -i` sur `~/dsh-workspace/` (tous les `.md`, hors sauvegardes) : aucune entrée « sudo par projet », « accord sudo », « périmètre sudo » n'existe. Les règles écrites (`MEMOIRE-CONTEXTE-REGLES-20261008.md`) disent l'inverse : elles posent l'**accord explicite obligatoire** pour toute action sensible (INTRUSION-1 §« Jamais sans accord explicite », CLAVIER-1 §5, CLAVIER-1 §« Interdits sans accord »). Ce registre **crée** donc la trace manquante.
2. **Le verrou technique est réel et mesuré** (§4). DSH ne peut pas exécuter `sudo` : pas de TTY, pas d'askpass, mot de passe non fourni. La politique d'approbation de la session est `ask` (mesuré le 15/10 à 10:45), mais une approbation autorise une exécution — elle ne fournit pas un mot de passe.

Ce document est la pièce manquante : il **nomme** le projet, **borne** les droits, **prouve** l'état actuel, et **explicite** ce que Pierre doit valider.

---

## 2. Principe retenu — « sudo par projet », pas « sudo pour l'agent »

### 2.1 La règle

Un projet qui a besoin de privilèges **automatiques et récurrents** (rotation planifiée, service systemd, plugin) reçoit des droits `sudo` **ciblés** : un ou deux binaires nommés, jamais `ALL`.

### 2.2 Les quatre garde-fous

| # | Garde-fou | Raison |
|---|---|---|
| G1 | **Ciblé** : `(root) NOPASSWD: /chemin/exact/binaire` | `ALL` = tout le système compromis si le script est remplacé |
| G2 | **Non modifiable par c3b** : le binaire ciblé appartient à `root`, `chmod 0755`, dossier parent non inscriptible par `c3b` | sinon c3b peut réécrire le binaire et exécuter n'importe quoi en root |
| G3 | **Sans argument libre** : le script ne doit pas prendre d'entrée arbitraire (pas d'`eval`, pas de `$1` passé à un shell) | sinon `/usr/local/bin/x "$(cat /etc/shadow)"` = lecture root |
| G4 | **Auto-limité** : le script écrit dans une liste blanche de chemins | sinon un script root peut tout écrire |

### 2.3 Ce que ce principe N'autorise PAS

- **Jamais** `NOPASSWD: ALL` ni `(ALL: ALL) ALL`.
- **Jamais** de droit sur `bash`, `sh`, `python3`, `cp`, `install`, `mv`, `chmod`, `tee`, `systemctl` (générique) : chacun permet une escalade triviale (`sudo bash` = root).
- **Jamais** de secret dans un fichier lisible par `c3b`, un log, ou un `argv`.
- **Jamais** d'installation/modification d'unité systemd sans validation humaine explicite (l'écriture dans `/etc/systemd/system` reste manuelle).

---

## 3. Projet n° 1 — `tor-securise` (rotation des bridges Tor)

### 3.1 Identification

| Champ | Valeur |
|---|---|
| **Projet** | `tor-securise` — rotation automatique des bridges obfs4 d'ENTRÉE |
| **Besoin** | écrire `/etc/tor/torrc.d/50-bridges.conf`, sauvegarder dans `/etc/tor/bridges-backup/`, redémarrer `tor@default` |
| **Fréquence** | au démarrage de la machine (`OnBootSec=2min`) + quotidiennement (`OnCalendar=daily`) |
| **Sans sudo** | impossible — `/etc/tor` appartient à `root:debian-tor`, mode `0755` |
| **Demandeur** | Pierre (c3b), 15/10/2026 |
| **Statut** | droits **existants** (hérités du 24/09) → à **valider ou retirer** |

### 3.2 Droits actuellement en vigueur (héritage, non validés)

Mesure : `sudo -n -l` (15/10/2026). Deux droits NOPASSWD touchent ce projet :

```
(root) NOPASSWD: /usr/local/bin/tor-securise-rotation
(root) NOPASSWD: /usr/local/bin/tor-securise-verifier
```

Ces deux droits ont été posés le **24/09/2026** (mtime des unités), soit **trois semaines avant** la présente demande, et **sans registre**. Ils sont la raison pour laquelle la v1 a pu écrire en production le 15/10 à 10:05:23 sans mot de passe (voir `ACTIONS-EN-COURS-20260907.md`, entrée 15/10 10:10).

### 3.3 Analyse de conformité aux garde-fous du §2

| Garde-fou | Droit `tor-securise-rotation` | Verdict |
|---|---|---|
| **G1** ciblé | chemin exact, un seul binaire | conforme |
| **G2** non modifiable par c3b | `-rwxr-xr-x root root /usr/local/bin root:root drwxr-xr-x` | conforme |
| **G3** sans argument libre | **NON conforme en v1** : la v1 lit `COUNT/BACKUP_DIR` dans l'environnement ; l'environnement étant réinitialisé (`env_reset`), l'appelant ne peut pas l'influencer — mais la v1 accepte aussi des arguments non validés. La **v3** valide ses entrées (`-n` numérique, `--dry-run`, etc.). | à corriger par l'installation de la v3 |
| **G4** auto-limité | v1 : écrit `/etc/tor/torrc.d/` + `/etc/tor/bridges-backup/` — non vérifié ligne à ligne. v3 : écritures vérifiées limitées à ces deux répertoires. | à confirmer sur la v3 |

**Conclusion** : le droit existant est **partiellement conforme**. Il est exploitable, mais il cible un script (v1) qui installe des bridges morts et dont le périmètre d'écriture n'est pas prouvé. Le remplacer par la v3 est le correctif.

### 3.4 Droits proposés (cible, une fois la v3.1 installée)

```
# /etc/sudoers.d/tor-securise  — projet tor-securise (c3b)
# Rotation des bridges Tor : lecture/écriture bornée à /etc/tor.
# Ne JAMAIS élargir à ALL, ni à systemctl générique.
c3b ALL=(root) NOPASSWD: /usr/local/bin/tor-securise-rotation
c3b ALL=(root) NOPASSWD: /usr/local/bin/tor-securise-verifier
```

> **À ne pas créer ce tour** : ce fichier relève de `visudo` et doit être écrit en root. Il est proposé, pas appliqué.

**Justification du maintien** : le service `tor-securise-rotate.service` est un `Type=oneshot` déclenché par timer. Sans NOPASSWD, le timer ne peut pas écrire dans `/etc/tor` → la rotation automatique est morte. C'est précisément le « fonctionnement automatique » visé par la demande de Pierre.

### 3.5 Alternative sans `sudoers` (à considérer)

| Option | Principe | Avantage | Inconvénient |
|---|---|---|---|
| **A. NOPASSWD ciblé** (proposé) | l'utilisateur `c3b` lance le script en root | simple, éprouvé | toute faille du script = root |
| **B. Service systemd `User=root`** | le script est lancé par systemd en root, sans sudoers | aucun droit sudoers à maintenir ; DSH ne peut PAS lancer le script à la main en root | il faut installer l'unité (une fois, manuellement) |
| **C. `CAP_DAC_OVERRIDE` sur le binaire** | capacité POSIX ciblée | plus fin que root total | non supporté pour un script interprété |

| Option | Principe | Avantage | Inconvénient |
|---|---|---|---|
| **D. `tmpfiles.d` / groupe `debian-tor`** | donner à `c3b` l'écriture sur `/etc/tor` | aucun sudo | trop large : c3b peut tout casser dans `/etc/tor` |

**Recommandation DSH** : **B** pour l'automatisme, **A** conservé uniquement si Pierre veut pouvoir déclencher une rotation à la main. B supprime le vecteur « DSH lance le script en root » signalé le 15/10 à 10:05 — c'est un gain de sécurité réel, au prix d'une installation manuelle unique.

---

## 4. Le verrou technique, mesuré (pourquoi DSH n'exécute rien)

```
$ sudo -n true
sudo: a password is required                    rc=1
$ tty
not a tty
$ sudo -S -p '' true </dev/null
sudo: no password was provided                  rc=1
```

Trois faits :

1. DSH tourne **sans TTY** : `sudo` ne peut pas demander de mot de passe.
2. **Aucun mot de passe n'est — et ne doit être — stocké en clair** (règle `code` §2, `hygiene-secrets`).
3. **Aucun askpass n'est installé** : `/etc/sudo.conf` n'a aucune directive `Path askpass` active, `SUDO_ASKPASS` est vide, et les binaires `ssh-askpass` / `gnome-ssh-askpass` sont absents (mesuré le 15/10 à 10:42).
4. **L'approbation de session n'est pas un mot de passe** : la politique d'approbation est passée à `ask` (mesuré le 15/10 à 10:45). Un answerer peut autoriser DSH à *lancer* une commande ; cela ne l'élève pas en root.

**Conséquence** : « Pierre écrit "ok sudo" dans le chat » **ne transmet pas le mot de passe**. L'installation reste une action humaine, faite dans un terminal réel. DSH prépare, vérifie, prouve — et s'arrête à la porte de `/etc`.

### 4.1 Demande d'élévation déposée — traçabilité (15/10, 10:42 CEST)

| # | Tentative | Résultat mesuré |
|---|---|---|
| 1 | `sudo -n true` | `sudo: a password is required` — **rc=1** |
| 2 | `sudo -n bash …/INSTALLER-v3.1.sh` | `sudo: a password is required` — rc=1, **aucune écriture** |
| 3 | `tty` | `not a tty` |
| 4 | `echo "${SUDO_ASKPASS:-}"` | **vide** |
| 5 | `ls /usr/bin/ssh-askpass /usr/libexec/openssh/gnome-ssh-askpass` | **absents** |
| 6 | `bash …/INSTALLER-v3.1.sh` (sans root) | intégrité **OK** → refus explicite « doit être lancé avec sudo », rc=1 |

Ces tentatives sont **non destructives** : `/etc/` est resté intact. La demande formelle d'élévation pour le projet `tor-securise` est donc **déposée et tracée** ; son acceptation est un acte humain (§5).

## 5. Procédure de validation (à faire par Pierre)

### Étape 1 — Vérifier le livrable (sans sudo, 20 s)

```
bash ~/dsh-workspace/tor-securise-plugin-20261015/v3.1-correctif-protectsystem/INSTALLER-v3.1.sh
```

Attendu : `=== 0. Contrôle du livrable` → 3 empreintes OK, unité sans `ProtectSystem`, syntaxe valide, puis **arrêt propre** avec le message « ce script doit être lancé avec sudo ». Rien n'est installé.

### Étape 2 — Installer (avec mot de passe, 1 à 3 min)

```
sudo bash ~/dsh-workspace/tor-securise-plugin-20261015/v3.1-correctif-protectsystem/INSTALLER-v3.1.sh
```

Fait : sauvegardes dans `/root/tor-securise-sauvegarde-<horodatage>/`, installation lib + script + unités, timer armé (démarrage + quotidien), test à blanc, rotation réelle (2 bridges vivants, le mort écarté), vérifications.
**Rollback automatique** si une vérification de l'étape 7 échoue.

### Étape 3 — Contrôler (sans sudo, 10 s)

```
bash ~/dsh-workspace/tor-securise-plugin-20261015/v3.1-correctif-protectsystem/VERIFIER-apres-installation.sh
```

Attendu : `BILAN : N OK / 0 problème(s)`.

### Étape 4 — Trancher la question du `sudoers`

Choisir **A** (NOPASSWD maintenu, rotation lançable à la main) ou **B** (service systemd seul, aucun NOPASSWD). Par défaut, DSH recommande **B**.

### Étape 5 — Retour arrière (si besoin)

```
sudo bash ~/dsh-workspace/tor-securise-plugin-20261015/v3.1-correctif-protectsystem/ROLLBACK.sh
```

## 6. Journal des décisions

| Date | Décision | Auteur | Preuve |
|---|---|---|---|
| 2026-09-24 | Droits NOPASSWD posés pour `tor-securise-rotation` + `-verifier` | (non tracé) | mtime unités `24/09 13:36` |
| 2026-10-15 10:05 | Rotation v1 lancée en root via NOPASSWD, sans demande | tour DSH | `sudo[1516830]` journal |
| 2026-10-15 10:20 | Bug `ProtectSystem=full` identifié (corrigé v3.1) | DSH | `tests/PREUVE-protectsystem-full.txt` |
| 2026-10-15 | **Demande de Pierre** : « sudo par projet » + registre + PDF | Pierre | ce document |
| 2026-10-15 10:42 | **Demande d'élévation déposée** (6 tentatives, toutes non destructives) → refus mesuré faute de TTY/askpass | DSH | §4.1 |
| | | DSH | §4, point 4 |

| Date | Décision | Auteur | Preuve |
|---|---|---|---|
| 2026-10-15 10:45 | Constat corrigé : la politique d'approbation de la session est **ask**, et non « désactivée » ; une approbation ne transporte pas de mot de passe | | |
| 2026-10-15 10:50 | **Registre + rapport publiés** sur `https://webtech3.com/publications/` (HTTP 200, `application/pdf`) | DSH | §9 |
| 2026-10-15 | **Registre créé**, droits non modifiés | DSH | `sha256sum -c MANIFESTE.sha256` |

---

## 7. Limites assumées

1. **Aucun droit n'a été créé ni modifié** par ce document : c'est un registre de proposition, pas une application. `/etc/sudoers` et `/etc/sudoers.d/` sont inchangés (leur lecture elle-même exige root).
2. La **décision antérieure** évoquée par Pierre n'a pas été retrouvée par écrit : ce registre la **crée**, il ne la cite pas.
3. La conformité **G4** (périmètre d'écriture) de la v3 est établie par lecture du script, pas par `strace` en root — non exécutable sans mot de passe.
4. Le mode réel de la v3 n'a **jamais** tourné hors simulation à ce jour (§8.3 du rapport PDF).

---

## 8. Références

- `~/dsh-workspace/tor-securise-plugin-20261015/` — livrable (manifeste 7/7 OK)
- `~/dsh-workspace/tor-securise-plugin-20261015/v3.1-correctif-protectsystem/`
- `~/dsh-workspace/MEMOIRE-CONTEXTE-REGLES-20261008.md` — règles INTRUSION-1 / CLAVIER-1
- `~/dsh-workspace/ACTIONS-EN-COURS-20260907.md` — journal d'actions
- `man sudoers` §`NOPASSWD`, `man systemd.exec` §`ProtectSystem`

---

## 9. Publication sur le site WebTech3

| Élément | Valeur |
|---|---|
| Site | `https://webtech3.com/` (DNS A → `187.124.39.137`, kvm4, Apache 2.4.68) |
| Dossier publié | `/var/www/html/assemblees-citoyennes-clean/webtech3/publications/` |
| URL du dossier | `https://webtech3.com/publications/` |
| Rôle serveur | `www-data:www-data`, fichiers `0644`, dossier `0755` |
| Contrôle | `curl -s -o /dev/null -w '%{http_code} %{content_type}'` → **200** **`application/pdf`** |
| Intégrité | `MANIFESTE.sha256` déposé à côté des documents |

**Sans secret** : les deux documents publiés ne contiennent ni clé, ni jeton, ni mot de passe ; la sortie `sudo -n -l` reproduite est une **liste de droits** (chemins de binaires), pas un secret. Aucune valeur issue des coffres n'y figure.