Sécuriser et maintenir un portail Portix après audit

Appliquer les contrôles de comptes, fichiers, réseau, sauvegardes et mises à jour depuis les outils de sécurité Portix.

Commencer par les accès

Un audit doit d’abord vérifier les comptes administrateurs, les mots de passe provisoires, la double authentification, les sessions actives et les permissions de chaque rôle.

Contrôler les fichiers

Les zones de téléversement ne doivent jamais exécuter du PHP. Les SVG et les documents actifs nécessitent un filtrage strict, tandis que les archives importées doivent être inspectées avant extraction.

Points de contrôle

- HTTPS imposé et proxies autorisés explicitement ;
- blocage SSRF des adresses privées et locales ;
- filtrage HTML par liste blanche ;
- limitation des connexions et soumissions ;
- journalisation des actions sensibles ;
- sauvegardes protégées et test de restauration ;
- mises à jour signées et vérifiées.

Surveiller sans exposer

Les journaux affichent des adresses masquées et ne doivent pas contenir de mots de passe, de jetons complets ou de clés privées. Les alertes doivent être calculées à partir de l’état réel du serveur.

Préparer la reprise

Avant chaque mise à jour, Portix crée une sauvegarde des fichiers remplacés. En cas d’échec, la restauration automatique ou la page de secours doit permettre de revenir à l’état précédent.

Rythme conseilléContrôler les sessions et alertes régulièrement, puis effectuer un audit complet après chaque changement important de configuration.

Une protection organisée par couches

La sécurité ne repose pas sur une seule option. Portix combine les mots de passe, les sessions, les permissions, la protection des formulaires, la limitation des tentatives et la journalisation. Chaque couche répond à un risque différent : usurpation de compte, action non autorisée, automatisation abusive, fuite de données ou modification non traçable.

Des droits adaptés aux responsabilités

Les fonctions visibles et les routes accessibles doivent correspondre au rôle réel de l’utilisateur. Un membre ne doit pas atteindre l’administration, tandis qu’un responsable métier ne doit voir que les opérations nécessaires à sa mission. Le filtrage des menus améliore la lisibilité, mais le contrôle côté serveur reste indispensable lorsqu’une adresse est saisie manuellement.

Surveiller sans exposer les données

Les journaux et la gestion des sessions doivent fournir assez d’informations pour comprendre un incident sans afficher inutilement des données sensibles. Les adresses réseau peuvent être masquées, les événements importants datés et les sessions révoquées individuellement. Cette approche facilite le diagnostic tout en limitant la diffusion d’informations personnelles.

Une maintenance régulière

Après chaque mise à jour, il faut vérifier les comptes administrateurs, les modules actifs, les permissions, les extensions PHP et les protections des répertoires privés. Une sauvegarde récente, un test de restauration et une revue périodique des journaux complètent le dispositif. La sécurité est un processus continu, pas une opération ponctuelle.

Auditer le comportement réel, pas seulement les fichiers

Un contrôle statique peut confirmer qu’une route est déclarée tout en manquant un défaut d’exécution. La recette Portix complète donc l’analyse des fichiers par des appels HTTP, des sessions authentifiées, des scénarios de paiement et des contrôles de première installation. Cette méthode a permis de détecter des routes présentes dans le code mais absentes du manifeste runtime, ainsi que des incohérences de services.

Transformer les anomalies en tests de régression

Chaque défaut significatif doit conduire à un test dédié : route de paiement, facture payée sans écriture, structure du foyer, synchronisation des services ou contenu de distribution. Le but est d’empêcher une correction de disparaître silencieusement dans une version ultérieure.