

Protégez les référentiels, réduisez les applications tierces à risque et corrigez les activités d'accès non autorisées "out of the box"

GitHub

Au cours du premier mois, Suridata a détecté et résolu automatiquement plus de **1 110 failles de sécurité, dont 15 % jugées critiques**, avec une surveillance minimale de la part des opérateurs de sécurité, ce qui a considérablement amélioré la posture de sécurité SaaS de GitHub.

Principaux problèmes que nous avons identifiés

Mauvaises configurations

- Branche par défaut sans protection de branche
- Protection de branche sans exigence d'examen des PR (Protection Rules)
- Clé de déploiement du référentiel datant de plus de 180 jours
- Longueur minimale de la stratégie de mot de passe inférieure à 8 caractères
- Validation en deux étapes non appliquée pour les administrateurs

Problèmes de plugins

- Applications tierces inactives connectées mais non utilisées depuis plus de 6 mois
- Applications tierces non vérifiées autorisées par les utilisateurs mais pas par GitHub
- Applications tierces à haut risque utilisées avec des autorisations incluant la lecture et l'écriture

Problèmes d'accès

- Référentiels accessibles au public pour le code source interne de GitHub
- Référentiel accessible à l'extérieur, non vérifié
- Les fichiers GitHub peuvent être validés par un collaborateur externe sans examen

Valeur immédiate

- Il s'agit d'une première analyse typique que nous exécutons pour un nouveau client, et elle représente certaines des failles de sécurité les plus critiques identifiées dès le départ.
- Dans les 24 premières heures, Suridata a identifié les lacunes critiques de mauvaise configuration.
- Dans les 48 heures, Suridata a identifié les applications tierces les plus à risque connectées à GitHub.
- Après la première analyse, toutes les 24 heures, Suridata surveille les changements/modifications et alerte lorsque de nouvelles lacunes sont identifiées.