GitHub

Neuf bonnes pratiques de sécurité que tous les leaders en conception de logiciels devraient connaître

1 Utilisez un protocole d'authentification centralisé, comme le SSO avec les protocoles LDAP ou SAML pour tous les systèmes de votre organisation.

L'utilisation du SSO pour toutes vos applications minimise le risque de sécurité et permet à votre organisation de garder le contrôle des accès utilisateurs.

Astuces

- Exécutez le SSO sur tous les systèmes qui l'autorisent.
- Ayez une procédure écrite afin d'inclure les nouveaux employés et d'exclure les employés qui quittent l'entreprise.
- Conservez les informations d'audit pour le contrôle d'accès.

2 Activez la MFA (Multi-Factor Authentication) sur tous les systèmes.

L'ajout de l'authentification multi-facteurs offre un niveau supplémentaire de protection et réduit grandement les risques de piratage de comptes utilisateurs.

Astuces

- Auditez tous les systèmes pour documenter leur utilisation de la MFA.
- Demandez une MFA sur tous les systèmes qui l'autorisent.
- Documentez le processus de configuration d'appareils utilisateurs pour la MFA, comme les applications d'authentification (Duo, Google Authy, etc.).

3 Documentez l'utilisation des données et les réglementations d'accès et assurez-vous qu'elles soient faciles à trouver.

Le fait de conserver les réglementations actuelles de sécurité dans un seul document permet à toutes les personnes de votre organisation de les retrouver facilement et de les suivre.

Astuces

- Présentez vos bonnes pratiques de sécurité dans un document facile à trouver pour les nouveaux projets et les projets existants.
- Permettez aux développeurs de maintenir facilement la sécurité en vérifiant régulièrement les configurations de sécurité de vos repositories organisationnels.

4 Cryptez les données en statiques et en transit dès que cela est possible.

Le cryptage des données en transit réduit les risques de fuites de données ou de vol de propriété intellectuelle.

Astuces

- Créez un rapport sur l'état actuel de toutes les données enregistrées (incluant une note signalant si les données sont cryptées ou non).
- Identifiez ce qui doit être crypté, et cryptez les données dans leur ordre d'importance.
- Documentez comment crypter les données et insérez cette référence dans le fichier CONTRIBUTING.md.
- Identifiez les systèmes qui n'utilisent pas de protocoles cryptés comme HTTPS et SSH et activez ces protocoles de cryptage.

5 Intégrez votre équipe de sécurité au développement et faites-les participer aux moments clés du planning et aux réunions de vérification.

Le fait de synchroniser le travail de votre équipe de sécurité et celui de vos développeurs vous permet une meilleure mise en oeuvre de la sécurité et des cycles de développement plus courts.

Astuces

- Informez votre équipe de sécurité de tous les nouveaux de développements afin qu'ils puissent participer aux discussions de lancement et de code.
- Ayez un processus écrit pour rapporter les failles de sécurité.

7 Assurez-vous que tout le monde au sein de votre organisation utilise une base d'identifiants partagée.

Le fait d'enregistrer les identifiants dans une application de base centralisée signifie que vous n'avez pas besoin de les conserver dans votre code source ni dans vos documents.

Conseils

- Utilisez un outil de base d'identifiants partagés (LastPass, 1Password, Dashlane, etc.) pour générer et stocker les mots de passe.
- Protégez cette base partagée avec une MFA pour un niveau supplémentaire de sécurité.

9 Suivez automatiquement les dépendances vulnérables et créez un processus établi pour traiter les alertes de sécurité.

L'analyse du code source à la recherche de vulnérabilités révèle des failles et des faiblesses qui pourraient être exploitées ou entraîner des fuites d'information.

Astuces

- Configurez des alertes de sécurité avec l'API de votre VCS pour suivre et résoudre les issues et les dépendances
- Rechercher automatiquement les vulnérabilités avant la mise en service pour améliorer la qualité du code et révéler les risques de sécurité

6 Scannez le code de votre équipe pour les secrets et les identifiants durant un commit.

Le fait de scanner les identifiants évite automatiquement aux développeurs de s'identifier accidentellement avec des mots de passe ou autres identifiants.

Astuces

- Scannez les identifiants pour tous les commit afin d'éviter de passer les identifiants en version de contrôle ou dans les environnements de production.
- Effectuez régulièrement un scan complet de tous les contenus du répertoire pour trouver tout identifiant existant.

8 Mettez à jour les tokens utilisateurs et les mots de passe de manière régulière et automatique.

Le fait d'alterner vos tokens et mots de passe réduit l'éventualité d'un accès non-autorisé.

Astuces

- Sachez où les mots de passe de tous les systèmes de votre organisation sont stockés.
- Intégrez des règles de mots de passe dans vos documents sur les normes de sécurité et demandez à tout le monde d'utiliser des mots de passe uniques.

Vérifiez votre score

Avez-vous répondu "non" ou "peut-être" à l'une des questions ci-dessus ? Votre équipe a peut-être encore quelques modifications de sécurité à apporter avant d'utiliser des logiciels open source.

Besoin d'aide pour démarrer ?

Parlez à un expert en sécurité de GitHub en contactant :

experts@github.com