OPENCLASSROOMS

Corrigé Activité P1C6

L'approche proposée par le développeur est une fausse bonne idée très répandue. Essayer de deviner et de filtrer tous les caractères malveillants est une course perdue d'avance. Les attaquants connaissent d'innombrables techniques d'encodage (hexadécimal, Unicode, etc.) pour contourner ces filtres basiques et réussir à faire interpréter leurs commandes par la base de données. De plus, bannir certains caractères peut empêcher des utilisateurs légitimes de réaliser des recherches normales (comme chercher un nom d'entreprise contenant une apostrophe).

Pour neutraliser le risque à la racine, vous devez inclure l'exigence technique incontournable suivante dans vos directives de développement :

"Le nettoyage manuel des entrées (Blacklisting/Échappement) est INTERDIT car faillible. Le système DOIT utiliser exclusivement des requêtes paramétrées (Prepared Statements) ou un ORM sécurisé pour toute interaction avec une base de données."

En imposant les requêtes paramétrées, vous obliger le système à séparer la logique de la commande de la donnée de l'utilisateur, ce qui rend l'injection structurellement impossible, quel que soit le texte farfelu ou malveillant saisi dans la barre de recherche.

Ne validez jamais les entrées utilisateur uniquement côté navigateur (frontend). Les contrôles en JavaScript dans l'interface graphique améliorent l'expérience utilisateur, mais un attaquant peut facilement les désactiver ou envoyer ses requêtes directement au serveur de TechNova. Toute validation et paramétrisation doit être exécutée strictement côté backend.