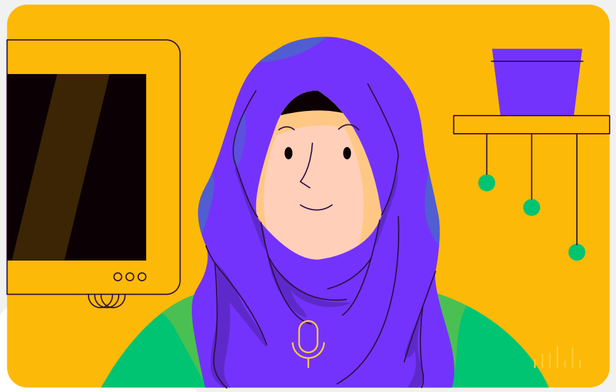
WONDER TECK

Test - SI

Présentation de WonderTeck.

Wonderteck est une entreprise tech qui conçoit et développe des applications destinées au grand public. Notre mission principale est d’identifier les problèmes du quotidien et d’y apporter des solutions digitales simples, efficaces et accessibles. Nous transformons les besoins réels des utilisateurs en applications concrètes, en combinant innovation, design centré utilisateur et technologies modernes.



Aujourd’hui, place à la pratique !

Préparez vos outils, aiguisez vos neurones et amusez-vous tout en montrant vos compétences.

Objectif

API exposée

Vous êtes chargé d'analyser la sécurité d'une API exposée publiquement.
Votre mission est de découvrir, analyser et documenter les vulnérabilités accessibles sans authentification à partir uniquement de l'URL fournie

Informations de départ

- Base URL de l'API : https://api.zpickupgroup.com/
- Contexte : aucune documentation ni compte utilisateur n'est fourni.
- Périmètre autorisé : uniquement l'API mentionnée ci-dessus.
- Interdit : attaques par déni de service, destruction de données, tests hors périmètre.

Tâches

1. Reconnaissance
   - Identifier les endpoints disponibles [via navigation, erreurs, outils comme Postman ou curl].
   - Vérifier la présence éventuelle d'une documentation publique
2. Analyse des endpoints publics
   - Lister les routes accessibles sans authentification.
   - Vérifier si elles exposent des données sensibles .
3. Tests de validation des entrées
   - Essayer des valeurs inattendues dans les paramètres [SQLi, NoSQLi, XSS].
   - Vérifier la présence de failles de mass assignment
4. Sécurité transport & configuration
   - Vérifier si l'API est accessible en HTTP non sécurisé.
   - Identifier les en-têtes de sécurité absents [Strict-Transport-Security, X-Content-Type-Options, etc.].

Tâches

5. Vérification CORS
   - Tester si l'API accepte des origines non autorisées (Access-Control-Allow-Origin: *).
6. Résilience & limitations
   - Tester la présence (ou absence) de rate limiting (ex. flood d'un endpoint public).
7. Gestion des erreurs
   - Vérifier si les messages d'erreur exposent des informations sensibles (stack trace, version de framework, détails SQL).

Livrable attendu

Rapport d’audit (5-8 pages max)

- Liste des endpoints découverts.
- Description claire des vulnérabilités trouvées.
- Pour chaque vulnérabilité :
  - Titre & description
  - Impact potentiel
  - Preuve d’exploitation (PoC) (commande curl ou capture Postman)
  - Correctif recommandé
- Collection Postman ou scripts cURL démontrant les tests effectués.

Merci pour votre participation
et votre sérieux.