

Corrigé Activité P1C7



L'équipe marketing est tombée dans le piège classique du "chemin heureux" : elle n'a pas envisagé qu'une personne malveillante pourrait utiliser la fonctionnalité exactement comme elle est conçue, mais pour nuire.

Voici le scénario de menace métier (Abuse Story) que vous devez exposer :

En tant qu'entreprise concurrente (ou utilisateur malveillant automatisé par un script), je réserve simultanément 100 % de l'inventaire des équipements de TechNova pour les six prochains mois. Étant donné qu'aucun paiement n'est exigé, cela ne me coûte rien. Le résultat est que tous les véritables clients verront le statut "Épuisé", bloquant totalement les ventes réelles de TechNova et causant une perte financière massive et un déni de service métier complet.

Pour corriger ce défaut de conception sans annuler l'idée du marketing, vous devez imposer l'exigence d'architecture suivante :

"Afin de prévenir l'épuisement des ressources partagées et le blocage malveillant des ventes, le système DOIT obligatoirement limiter la fréquence et le volume maximal de réservations possibles par compte utilisateur sur une période donnée. De plus, pour toute réservation dépassant un seuil défini de ressources, le système DOIT exiger une empreinte bancaire préalable ou un dépôt de garantie partiel."

En posant cette exigence, la logique métier devient intrinsèquement robuste. Même si l'attaquant écrit un script parfait pour réserver des équipements, la conception même du système l'arrêtera une fois son quota atteint ou son compte bancaire sollicité.

Ne vous contentez pas de concevoir le "chemin heureux" (Happy Path) lors de l'écriture des spécifications. Imaginez toujours l'utilisateur le plus malveillant ou le plus stupide possible, et concevez l'architecture du système pour limiter l'impact de ses actions.