Audit Sécurité

but informatique |Groupe D1

Equipe laq | SAE S4

Quentin Bouchot | leandre moreau | Axel lhenry

2023

# contexte de l’application

## public visé par l’application (interne, grand public…) et leur niveau de sensibilité aux problèmes de sécurités

Le public visé par l’application est le **grand public**. En effet *TaskStep* est une **ToDoList**, ce qui s’apparente à vouloir toucher n’importe quel type de personnes. Cela se confirme quand on lance la solution, aucun élément dans l’application semble indiquer que celui cible des spécifiquement une catégorie de personne tel que des professionnels.

Le niveau de sensibilité du grand public sur la sécurité d’une ToDoList est relativement faible. Une ToDoList est, d’en l’esprit général, pas sensé demander et utilisé des données très sensibles tel que des coordonnées bancaires.

## les différents types d’utilisateurs (visiteur, utilisateur enregistré, administrateurs…)

Dans cette application Web, censé être multi-utilisateurs, il devrait exister 3 types d’utilisateurs en théorie :

* Les visiteurs : Ils ne peuvent qu’accéder à la page d’inscription/connexion.
* Les utilisateurs enregistrés : Ils ont accès à toutes les fonctionnalités de l’application prévue pour l’utilisateur mais pas à la gestion des comptes.
* Les administrateurs : Ils ont accès à toutes les fonctionnalités de l’application prévue pour l’utilisateur et à la gestion des comptes.

Dans la pratique en revanche :

* La notion de compte utilisateur n’est pas vraiment présente car tous les monde est dans la même session
* Il n’y a que des visiteurs et des utilisateurs
* La notion de visiteurs est partiellement présente car les utilisateurs peuvent décider d’accéder à l’application sans mot de passe.
* Les utilisateurs sont tous administrateurs car ils ont accès à toutes les fonctionnalités et la gestion des comptes

## ensemble des informations manipulées

L’application manipule :

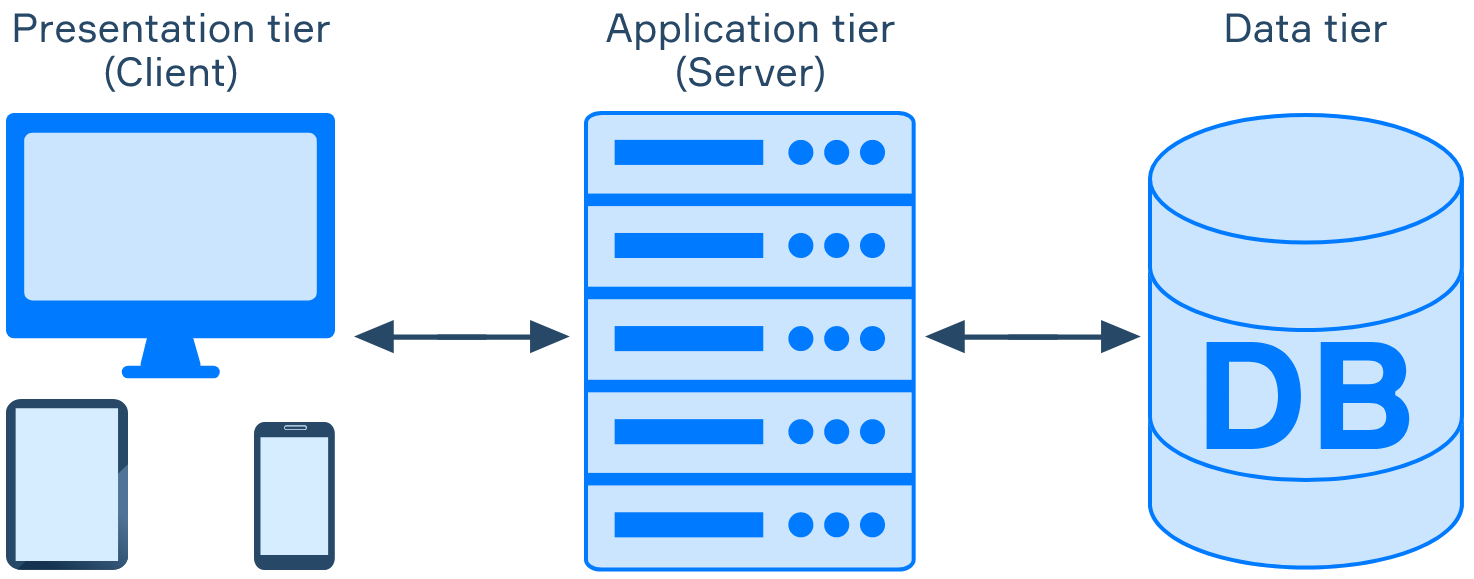
* Tâches
  + Un titre
  + Une date
  + La section
  + Une note (description)
  + Une URL
  + Une information de complétion
  + Un contexte
    - Un titre
  + Un projet
    - Un titre
* Des settings
  + On y stocke un tableau (clé, valeur) où est stockée :
    - Un mot de passe
    - Le salt du mot de passe
    - Un numéro de session
    - Le style CSS actif l’utilisateur
    - La préférence de l’utilisateur vis-à-vis de l’affichage ou non de la fenêtre de conseils sur la page d’accueil.

## ensemble des fonctionnalités

* Voir l’ensemble des Taches créés, faites ou à faire, pour la journée actuelle
* Créer une tache personnalisée
* Validé/ modifier/ supprimer une tâche.
* Ajouté/modifier/supprimer des contextes
* Ajouté/modifier/supprimer des projets
* Voir l’ensemble des tâches créent.
* Rechercher une tâche crée en fonction d’un contexte
* Rechercher une tâche crée en fonction d’un projet
* Changer ses paramètres d’utilisation (style CSS)
* Se connecter
* Changer de mot de passe
* Désactivé le système de connexion
* Désactiver l’affichage des tips
* Accéder à une page d’assistance à non implémenté
* Mettre nos tâches dans un format imprimable
* Supprimer toutes les tâches terminées
* Exécuter le fichier de mise à jour
* Exporter en CSV les données
* Mettre en favori l’application

## les différents composants structurels de l’application

C’est une application client/serveur avec une base de données ayant une architecture à trois niveau (*three-tiers*). Cette base de données hébergé sur PHPMyAdmin.



# degrés de sensibilité

## Pour chaque donnée, déterminer son degré de sensibilités et qui doit pouvoir y avoir accès

* Tâches : Sensible
  + Un titre : Sensible
  + Une date : Sensible
  + La section : Peu sensible
  + Une note (description) : Sensible
  + Une URL : Sensible
  + Une information de complétion : Sensible
  + Un contexte : Sensible
    - Un titre : Sensible
  + Un projet : Sensible
    - Un titre : Sensible
* Des settings
  + On y stocke un tableau (clé, valeur) où est stockée :
    - Un mot de passe : Très sensible
    - Le salt du mot de passe : Peu sensible
    - Un numéro de session : Très sensible
    - Le style CSS actif l’utilisateur : Sensible
    - La préférence de l’utilisateur vis-à-vis de l’affichage ou non de la fenêtre de conseils sur la page d’accueil. : Sensible

Très sensible à Données extrêmement confidentiel qui peuvent permettre d’identifier l’utilisateurs

Sensible Données liée à l’utilisation de l’application par l’utilisateur, les logs sont obligatoires pour accéder à ces informations

Peu sensible Ce n’est pas une donnée qui touche pas à l’utilisateur et qui peuvent être public.

## Pour chaque fonctionnalité, déterminer son degré de sensibilités et qui doit pouvoir y avoir accès

* Voir l’ensemble des Taches créés, faites ou à faire, pour la journée actuelle : Sensible
* Créer une tache personnalisée : Sensible
* Validé/ modifier/ supprimer une tâche : Sensible
* Ajouté/modifier/supprimer des contextes : Sensible
* Ajouté/modifier/supprimer des projets : Sensible
* Voir l’ensemble des tâches créent : Sensible
* Rechercher une tâche crée en fonction d’un contexte : Sensible
* Rechercher une tâche crée en fonction d’un projet : Sensible
* Changer ses paramètres d’utilisation (style CSS) : Sensible
* Se connecter : Très sensible
* Changer de mot de passe : Très sensible
* Désactivé le système de connexion : Très sensible
* Désactiver l’affichage des tips : Sensible
* Accéder à une page d’assistance à non implémenté : Sensible
* Mettre nos tâches dans un format imprimable : Sensible
* Supprimer toutes les tâches terminées : Sensible
* Exécuter le fichier de mise à jour : Sensible
* Exporter en CSV les données : Sensible
* Mettre en favori l’application : Sensible

# les vulnérabilités potentielles

## les protections à réaliser vis-à-vis des données et fonctionnalités

### donnees

* Tâches : Sensible
  + Un titre : Sensible --> L'utilisateur doit être connecté
  + Une date : Sensible --> L'utilisateur doit être connecté
  + La section : Peu sensible --> Données public, rien à faire
  + Une note (description) : Sensible --> L'utilisateur doit être connecté
  + Une URL : Sensible --> L'utilisateur doit être connecté
  + Une information de complétion : Sensible --> L'utilisateur doit être connecté
  + Un contexte : Sensible --> L'utilisateur doit être connecté
    - Un titre : Sensible --> L'utilisateur doit être connecté
  + Un projet : Sensible --> L'utilisateur doit être connecté
    - Un titre : Sensible --> L'utilisateur doit être connecté
* Des settings
  + On y stocke un tableau (clé, valeur) où est stockée :
    - Un mot de passe : Très sensible
      * Hashage du mot de passe avec le salt
      * Vérification du niveau de sécurité
    - Le salt du mot de passe : Peu sensible --> Données public, rien à faire
    - Un numéro de session : Très sensible
      * Interne au serveur et inaccessible par l’utilisateur
      * Génération en random
    - Le style CSS actif l’utilisateur : Sensible --> L'utilisateur doit être connecté
    - La préférence de l’utilisateur vis-à-vis de l’affichage ou non de la fenêtre de conseils sur la page d’accueil : Sensible --> L'utilisateur doit être connecté

### fonctionnalités

* Voir l’ensemble des Taches créés, faites ou à faire, pour la journée actuelle : Sensible --> L'utilisateur doit être connecté
* Créer une tache personnalisée : Sensible --> L'utilisateur doit être connecté
* Validé/ modifier/ supprimer une tâche : Sensible --> L'utilisateur doit être connecté
* Ajouté/modifier/supprimer des contextes : Sensible --> L'utilisateur doit être connecté
* Ajouté/modifier/supprimer des projets : Sensible --> L'utilisateur doit être connecté
* Voir l’ensemble des tâches créent : Sensible --> L'utilisateur doit être connecté
* Rechercher une tâche crée en fonction d’un contexte : Sensible --> L'utilisateur doit être connecté
* Rechercher une tâche crée en fonction d’un projet : Sensible --> L'utilisateur doit être connecté
* Changer ses paramètres d’utilisation (style CSS) : Sensible --> L'utilisateur doit être connecté
* Se connecter : Très sensible --> L'utilisateur doit être connecté
* Changer de mot de passe : Très sensible --> L'utilisateur doit être connecté
* Désactivé le système de connexion : Très sensible --> Seul l’administrateur doit pouvoir faire ça
* Désactiver l’affichage des tips : Sensible --> L'utilisateur doit être connecté
* Accéder à une page d’assistance à non implémenté : Sensible --> L'utilisateur doit être connecté
* Mettre nos tâches dans un format imprimable : Sensible --> L'utilisateur doit être connecté
* Supprimer toutes les tâches terminées : Sensible --> L'utilisateur doit être connecté
* Exécuter le fichier de mise à jour : Sensible --> L'utilisateur doit être connecté
* Exporter en CSV les données : Sensible --> L'utilisateur doit être connecté
* Mettre en favori l’application : Sensible --> L'utilisateur doit être connecté

## l’ensemble des vulnérabilités structurelles

Non-gestion des codes d’erreurs.  
Injections SQL possibles dans les champs d’entrée.  
Obsolescence des technologies utilisées.

Mauvaise gestion des mots de passes

# l’existant

## Pour chaque vulnérabilité potentielle, établir la démarche de sécurité réalisée par l’existant

Le mot de passe est chiffré et déchiffré par la méthode de chiffrement **MD5** (« *Message-Digest Algorithm* ») ainsi qu’avec l’ajout d’un sel (« Salt »). Technologie obsolète !

Une image contenant texte

Description générée automatiquement

Dans l’application existant, on peut choisir de ne pas utiliser de mot de passe ni de session !

Une image contenant texte

Description générée automatiquement

Le niveau de sécurité de l’application est donc mis à néant (même si c’est voulu par l’utilisateur) !!!

Or lors de l’installation de l’application, la session est toujours égale à « 1 », ce qui fait que peut importe l’utilisateur, il n’existe qu’une seule *ToDoList* pour tout le monde et quel est potentiellement accessible par tout le monde si un utilisateur décide de décocher l’utilisation de mot de passe et de session.

²²

Deux problèmes majeurs qui se retrouve entremêlé l’un dans l’autre.

L’injection SQL dans les champs d’entrées est possible.

Le mot de passe transite en claire lors d’un changement de mot passe.

## Pour chaque vulnérabilité potentielle, évaluer la pertinence et la suffisance de la démarche de sécurité réalisée par l’existant

Pour les mots de passes, le chiffrement de ceux-ci est pertinent dans l’idée. En revanche il est totalement impertinent de donner la possibilité aux utilisateurs de pouvoir ne plus utiliser de mot de passe et de session ! Cette fonctionnalité pause un sérieux problème de sécurité pour l’application. De plus cette méthode de chiffrement est insuffisante de nos jours, le MD5 est une technologie obsolète.

L’installation de l’application pose également un problème puisqu’un compte par défaut est créer (et il est impossible d’en créer un nouveau) avec un numéro de session fixe pour tout le monde.

La protection contre les injections SQL est insuffisante.

Recommander des corrections

## Pour chaque vulnérabilité non correctement sécurisée, proposer une démarche de sécurité adaptées pour corriger l’existant

Pour les mots de passes, on peut retirer la fonctionnalité qui permet de désactiver l’utilisation du mot de passe et de session. On peut également implémenter :

* Un nom d’utilisateur et son mot de passe pour dissocier et reconnaitre les utilisateurs entre eux de manière sécurisée.
* Un nom de ToDoList et son mot de passe pour dissocier et reconnaitre les ToDoList entre elles de manière sécurisée.
* Passe le site en HTTPS
* Mettre à jour le code de l’application