TP Cyber : M83 : Partie 1 : Détails

lundi 12 février 2024 10:01

Prérequis : Installation Linux/Windows - Apache - MySql
NB : Ne pas oublier de garder les programmes avec les versions validées

® Nomenclature et présentation :
   a. TPx : Pour la création
   b. TPFx : Pour l'exploitation de la faille
   c. TPSFx : Pour la suppression de la faille

TP : Fonctionnement du site Web

Objectifs : Vérifier le fonctionnement du site WEB (HTML)

TP 1 : RGPD

- Créer une page HTML avec les règles RGPD

Les étapes :

- Créer une page d'accueil : index.html
- Créer un lien vers la page rgpd.html
- Créer une page RGPD : rgpd.html

Les étapes en détail :

- Créer un répertoire où poser les programmes : /cyber
- Créer un alias : cyber dans httpd.conf
- Créer une page d'accueil : index.html
- Créer une page d'accueil : index.html
- Créer un lien vers la page rgpd.html
- Créer une page RGPD : rgpd.html

TP : Vérification du fonctionnement LAMP/WAMP et des développements PHP

Objectifs : Vérifier le fonctionnement du site WEB (HTML/LAMP ou HTML/WAMP) et créer des programmes en PHP non sécurisés.

TP 2 : Créer un environnement utilisateur pour accéder à une base de données

Les étapes :

- Ajout d'un lien dans index.html
- Créer un Formulaire de création d'un utilisateur
    - Formulaire : Login - MdP (en clair) - Bouton Création : creation_utilisateur.html + programme

- Créer un formulaire de Login Utilisateur
    - Formulaire : Login - MdP : login_utilisateur.html + programme
- Créer un Formulaire de création d'un utilisateur avec validation RGPD
    - Formulaire : Login - MdP (en clair) - Bouton Création + Validation des règles RGPD : creation_utilisateur_rgpd.html + programme
- Suppression de l'utilisateur (User)
    - Après Login - - Bouton Suppression : profil_utilisateur.html + programme

Les étapes en détail :

- Ajout d'un lien dans index.html
- Prérequis : Création de la base de données :

  CREATE DATABASE cyber
- Créer un Formulaire de création d'un utilisateur
    - Formulaire : Login - MdP (en clair) - Bouton Création : creation_utilisateur.html + programme "traitement_creation_utilisateur.php" (svg : TP2)
- Création de la table utilisateurs :

```
CREATE TABLE utilisateurs (
  id INT AUTO_INCREMENT PRIMARY KEY,
  login VARCHAR(50) UNIQUE,
  mot_de_passe VARCHAR(255)
);
```

- Créer un formulaire de Login Utilisateur
    - Formulaire : Login - MdP : login_utilisateur.html + programme "traitement_login_utilisateur.php" (svg TP2)
- Créer un Formulaire de création d'un utilisateur avec validation RGPD
    - Formulaire : Login - MdP (en clair) - Bouton Création + Validation des règles RGPD : creation_utilisateur_rgpd.html + programme "traitement_creation_utilisateur_rgpd.php"
- Suppression de l'utilisateur (User)
    - Après Login - - Bouton Suppression : profil_utilisateur.html + programme "traitement_login_suppression_utilisateur.php" (svg : TP2)
    - Remarques :
        - Création du profil de l'utilisateur : Permettrait d'aller plus loin pour gérer des profils (comme beaucoup de sites). Mais ce n'est pas nécessaire pour ce TD.
        - Ajout de la gestion des sessions avec l'enchainement des programmes :
            1) suppression_utilisateur.html
            2) => traitement_login_suppression_utilisateur.php

3) formulaire_suppression_utilisateur.php

4) => traitement_suppression_utilisateur.php

- Ajout de la gestion des sessions avec le programmes : gestion_sessions.php

1) Prérequis : Création de la table sessions

Les vérifications du TP 2 :

NB : On fera attention à créer des comptes qui identifient le TP afin d'identifier les mots de passe cryptés et non cryptés

- Création des utilisateurs avec les formulaires :
  - usr : cyber_TP2_1 ; mdp : cyber_TP2_1_mdp
  - usr : cyber_TP2_2 ; mdp : cyber_TP2_2_mdp
  - Création des utilisateurs avec le formulaire RGPD :
  - usr : cyber_TP2_3 ; mdp : cyberTP2__3_mdp
- Vérification en base de données :
  - Vérifier que l'utilisateur est bien créé dans la table « utilisateurs » (+champ coché RGPD)
  - Vérifier que le mot de passe de l'utilisateur n'est pas crypté dans la table « utilisateurs »
- Suppression du User : < !> ne supprime rien si on n'est pas capable d'identifier l'utilisateur connecté ou au contraire pourrait supprimer tous les utilisateurs
- Prérequis : Il est nécessaire d'avoir une gestion de la session et de l'utilisateur connecté pour pouvoir supprimer l'utilisateur (à traiter en TP 6 ?)

TP 3 : Créer un Formulaire de création d'un utilisateur avec MdP Crypté

Les étapes (3 steps) :

- Ajout des liens dans index.html
- TP3 : Step0 => Correspond à la création/login sans requête SQL préparé mais avec un mot de passe md5.
- TP3 : Step1 => Correspond à la création/login sans requête SQL préparé mais avec un mot de passe "bcrypt/sha-1".
- TP3_Step2
  - => Correspond à la création/login avec requête SQL préparé, avec un mot de passe "bcrypt" => les failles par injection SQL sont traitées.
  - Formulaire RGPD
- TPF 3 : Suppression : on montre que l'on peut supprimer un compte. **Faille CSRF**

Les vérifications du TP 3 :

- Vérification en base de données :

- Vérifier que l'utilisateur est bien créé dans la table « utilisateurs »
- Vérifier que le mot de passe de l'utilisateur est crypté dans la table « utilisateurs »
- Noter la différence entre md5 et bcrypt/sha-1

TP 4 : Créer un Formulaire de création de la vente d'un véhicule ou autre

Les étapes :

- Ajout des liens dans index.html
- Création de la table vehicule numDeSerie, Marque, Modele, Prix
- Formulaire "Créer un véhicule"
  - Formulaire_creation_vehicule.php
- Formulaire "Supprimer un véhicule"
  - Supprimer_vehicule.php

Les vérifications du TP 4 :

- Vérification en base de données :
  - Vérifier que le véhicule est bien créé dans la table « véhicule »
  - Peut-on identifier propriétaire ?
  - Comment vérifier qu'un utilisateur est connecté ?

Remarques :

- **< !> le véhicule peut être créé et supprimé par un utilisateur connecté ou non connecté ?**
- **N'y a-t-il pas un problème d'intégrité ou de sécurité ?**
- **On continue sur le TP5 puis ensuite sur le TP6**

TP 5 : Tableau listant les modèles vendus

Les étapes (2 steps) :

- Ajout des liens dans index.html
- TP5 : Step1 : Tableau des véhicules sans vérification des utilisateurs
  - Liste de tous les véhicules en vente : Marque, Modele, Prix => Accessible par tous
    - Programme : vehicules_en_vente.php
  - Liste de tous les véhicules en vente avec des liens pour voir => Accessible par tous
    - Programme : vehicules_details.php
- TP5 : Step 2 : Tableau des véhicules avec vérification des utilisateurs
  - Liste de tous les véhicules en vente d'un utilisateur : Marque, Modele, Prix => Accessible par l'utilisateur déclarant

        - Programme : vehicules_utilisateur.php
    - Liste de tous les véhicules en vente d'un utilisateur avec des liens pour voir, modifier, supprimer **(sans contrôle de l'utilisateur)**
        - Programme : vehicules_utilisateur_details.php
        - Programme de suppression sans contrôle utilisateur : supprimer_vehicule.php
    - Liste de tous les véhicules en vente d'un utilisateur avec des liens pour voir, modifier, supprimer **(avec contrôle de l'utilisateur)**
        - Programme de suppression avec contrôle utilisateur : supprimer_vehicule_utilisateur.php

Les vérifications du TP 5 :

- Vérification en base de données :
    - Vérifier que le véhicule est bien créé/supprimé dans la table « véhicule »

Remarques :

- **< !> le véhicule peut être créé et supprimé par un utilisateur connecté ou non connecté ?**
- **On veut montrer qu'avec des informations sur un tableau, on peut lancer des attaques…**

TP 6 : Créer un Formulaire de création d'un utilisateur avec MdP Crypté (Gestion des sessions)

- Vérification en base de données :
    - Vérifier que l'utilisateur est bien créé dans la table « utilisateurs »
    - Vérifier que le mot de passe de l'utilisateur est crypté dans la table « utilisateurs »
    - Vérifier que la session utilisateur est bien créé dans la table « sessions »

Les vérifications du TP 6 :

a) Vérifier si l'utilisateur est connecté
b) Vérifier si l'utilisateur est inscrit dans la table session
c) Comment créer l'utilisateur dans la table session
    a. Lorsqu'il se connecte il crée un enregistrement dans la table session :
    b. Avec $idSession = session_id(); name=login
        i. S'il n'est pas connecté => créer un enregistrement avec $idSession = session_id();
        ii. Si il est déjà connecté vérifier si présent dans la table session

TPF X : Exploitation des Failles

TPF 2 : Exploitation Faille Injection SQL

- **Test avec** Formulaire : Login - MdP (en clair) - Bouton Création : creation_utilisateur.html + programme "traitement_creation_utilisateur.php" (svg : TP2)
- **On ajoute la chaine : ' OR 1=1 OR 1='**

- **Essai avec** traitement_login_utilisateur_TP2_v1.php
- Programme pour tester la faille Injection SQL :
    - traitement_creation_utilisateur*pwd_md5
    - traitement_login_utilisateur*pwd_md5

TPF 3 : Exploitation Faille CSRF(utilisateur)

TPF 5 : Exploitation Faille CSRF (véhicule)

- On modifie les paramètres du programme php.
- On récupère à partir du tableau de ventes des véhicules, l'identifiant. Ensuite, si on est connecté, on passe en paramètre l'id du véhicule ! Le véhicule est supprimé.

TP 5-F : Exploitation Faille XSS (véhicule/création)

- Injection d'un script dans un formulaire (Javascript) dans la partie modèle
- <script type="text/javascript">alert('Bonjour')</script>
- <script type="text/javascript">window.location.replace("http://www.cisco.com");</script>

TPSF X : Suppression des Failles

TPSF 2 => TP3 : Suppression de Faille Injection SQL

- Magic_quotes : Attention peut être dépendant du paramétrage du serveur "php.ini"
- Fontion $login = addslashes($login) => $login = addslashes($login);
- Requête préparée

TPSF 3 => TP6 : Exploitation Faille CSRF : Permettre à une autre personne de supprimer un autre compte

- Correction faite en TP6 : Gestion des sessions

TPSF 5 : Exploitation Faille CSRF : Permettre à une autre personne d'exécuter du code

- A faire sur Vehicule
- Correction : Utilisation de l'autre programme qui contrôle la gestion de session supprimer_vehicule_utilisateur?id=xxx
- Si on ajoutait une contrainte d'intégrité ?

TPSF 5 : Exploitation Faille XSS : Injecte un script à partir d'un autre site (utilisation de JVS)

Correction : Correction avec htmlentities() ou strip_tags() => dépend de ce que l'on veut afficher ou utiliser !

Le texte est converti en empêchant l'exécution du code :

<script type="text/javascript">window.location.replace("http://www.cisco.com");</script>

TPSF 6 : Gestion des sessions

Y a-t-il encore des failles ?

Résume des failles à investiguer :

- **Faille Include :**
- **Mot de passe en clair**
- **Fonction password_hash()**
- **extraire à partir de la console MS , mode terminal , prompt php :**
- **taper echo password_hash('mypassword', PASSWORD_DEFAULT);**
- **password_verify(('mypassword',$2y$10 $j48nLTi6uainwbWTYc46D.I37hHMEUDJDcZEVBnoM3TZauJSjpb2O); => va retourner true**

Nomenclature pages html et programmes php (versions actives)

- Index.html
- Rgpd.html
- Login_utilisateur.html =>traitement_login_utilisateur.php
- Version par TP et par step

Sous-programmes :

- Init.sql : Contient les informations de connexion à la base
- Gestion_sessions.php : Contient les programmes de gestion des sessions utilisateur

Remarques sur le comportement pages html et programmes php

- En TP2 : L'œil du mdp n'apparait plus.
- Est-il nécessaire de contrôler les entrées au niveau du formulaire ?
- Est-il nécessaire de contrôler les entrées des programmes php ?