**RAPPORT TP SECURITE**

**Exercice 1 :**

Mot de passe : **Pr0t3g3z\_V0s\_Acc3s\_1nd1r3ct**

Pour accèder à au fichier succes.html il suffit de remplacer dans l’url index.html par la page recherchée. Afin de trouver la faille il a fallu retourner à la racine de l’exercice pour pouvoir voir toutes les pages constituants l’exercice.

Pour corriger la faille il faudrait vérifier si l’utilisateur qui accède à la page a les droits nécessaires pour y accèder.

**Exercice 2 :**

Mot de passe : **N3\_p@s\_St0ck3r\_L3s\_M0ts\_D3\_P@ss3\_D@ns\_L3\_Fr0nt**

Les identifiants de connections sont vérifiés dans le script javascript de la page, il est donc accessible par tous les utilisateurs du site internet. En accédant aux sources clients du site on accède donc aux identifiants attendus.

Pour corriger la faille, la connexion doit être gérer dans le coté back du site internet.

**Exercice 3 :**

Faille XSS, pour afficher un texte à l’arrivée sur la page il faut insérer dans un commentaire ce texte : <img onerror="alert('e')" src="." />

Pour corriger la faille il faudrait échapper les données dynamiques à l’affichage et filtrer les données reçues par le client.

**Exercice 4 :**

Le serveur renvoie les identifiants dans le header de la requête de réponse même si on a entré de mauvais identifiants. Utilisateur : **CalvinKim**, Mot de passe **: Jc8b&RM52AL**

Pour corriger il suffit de ne pas mettre ces identifiants dans le header de la réponse.

**Exercice 5 :**

Le serveur renvoie un user-agent dans le header de la réponse. Il suffit alors de changer son user-agent en toto avant d’envoyer la requête.

Pour corriger il suffit de ne pas renvoyer l’user agent dans le header de la réponse

**Exercice 6 :**

Faille SQL : mettre ‘ OR 1=1 /\* dans le champ utilisateur

Le serveur n’échappe pas les caractères ce qui permet de fermer le paramètre de vérification d’utilisateur, ensuite la vérification OR 1=1 et la vérification du mot de passe est mis en commentaire grâce au /\*

Pour remédier à cette faille il faut échapper les éléments dynamiques et utiliser les requêtes préparées.

**Exercice 7 :**

Mot de passe : toto123lol

Le code source est obfusquée, il suffit d’utiliser un desobfuscator et d’y traduire le code source on y retrouve bien la vérification du mot de pas du côté client.

Pour remédier à cette faille il ne faut pas vérifier le mot de passe du côté du client car tout obfuscation pour être inversée.