Exercice 1 :

Type de faille : Accès indirect non autorisé

Mots de passe : Pr0t3g3z_V0s_Acc3s_1nd1r3ct

Comment reproduire la faille ?

Laisser l'accès à la page success.html sans restriction

Comment trouver la faille ?

Ajouter le nom de la page à la fin de l'URL

Solution pour corriger la faille ?

Ajouter des restrictions pour accéder à la page (session,mots de passe etc...)

Exercice 2 :

Type de faille : Faille de sécurité basique

Mots de passe : N3_p@s_St0ck3r_L3s_M0ts_D3_P@ss3_D@ns_L3_Fr0nt

Comment reproduire la faille ?

Afficher des informations confidentiel directement dans le front

Comment trouver la faille ?

Rechercher des informations dans le front avec l'inspecteur du navigateur.

Ici l'id et le mdp sont visible dans le fichier javascript.

Solution pour corriger la faille ?

Ne pas afficher d'informartions confidentiel directement dans le front

Exercice 3 :

Type de faille : XSS

Comment reproduire la faille ?

Ne pas filtrer les données reçus par un tiers et ne pas échapper les données dynamique afficher

Comment trouver la faille ?

Dans la partie commentaire, j'écris :

```
<iframe/src="data:text/html,<svg onload=alert(1)>">
```

Ce code utilise du javascript directement et étant donné qu'il est stocké, à chaque chargement de la page l'alert se relance.

Solution pour corriger la faille ?

Filter les données reçus avant de les stocker et échapper les données dynamique avant de les afficher.

Exercice 4 :

Type de faille : Faille de sécurité basique

Mots de passe : Jc8b&RM52AL (User : CalvinKim)

Comment reproduire la faille ?

Laisser accessible tout les détails du "Response Headers" avec les id et mdp

Comment trouver la faille ?

Utiliser l'outil d'inspection du navigateur et explorer avec "Network" le "Respons Headers"

Solution pour corriger la faille ?

Limiter les détails donné par le response headers et surtout pas afficher les id et mdp dans les en-têtes http.

Exercice 5 :

Type de faille : faille de sécurité network

Mots de passe : toto

Comment reproduire la faille ?

Utiliser l'agent-user pour se connecter.

Comment trouver la faille ?

On utilise l'inspecteur du navigateur, dans l'onglet network, les « Headers » nous donne pas mal d'informations dont l'user-agent. On remarque ici que l'user-agent est « toto ». On doit donc modifier manuellement celui-ci afin de ce faire passer pour l'user-agent « toto » et ainsi pouvoir se connecter.

Solution pour corriger la faille ?

Ne pas utiliser seulement l'user-agent pour se connecter ou acceder à une page.

Exercice 6 :

Type de faille : Injecton SQL

Comment reproduire la faille ?

Dans le formulaire de connexion, on peut injecter du SQL avec par exemple 'OR 1=1/*' . Cette injection permet de réussir la requêtes et '/*' permet d'ignorer le reste de la requête.

Comment trouver la faille ?

Pour trouver la faille on peut simplement faire un essai d'injection dans le formulaire. On ne peut pas savoir à l'avance si les requêtes sont préparées ou non.

Solution pour corriger la faille ?

Utiliser des requêtes préparées et échapper les élements dynamique