Examen Sécurité 06/11/2021

Julien Sibille

Rapport de Julien SIBILLE

Exercice 1

Pour reproduire la faille, il suffit de remplacer index.html dans l'URL par success.html.
Afin de corriger la faille, il faudrait soit demander un mot de passe pour ouvrir la page success, soit vérifier que l'utilisateur en cours à bien les droits nécessaires d'y accéder.

Le mot de passe est : **Pr0t3g3z_V0s_Acc3s_1nd1r3ct**

Exercice 2

On est sur une page de login sans plus d'informations. J'ai donc ouvert le devtools du navigateur et j'ai constaté dans le fichier exo2.js que l'user et le mot de passe étaient stockés en clair. J'ai donc copié ces informations et tenté de me connecter, avec succès.
Pour corriger ce problème il faut tout d'abord ne jamais stocker de mots de passe en clair et non hashés. Et pour finir, il faut que les mots de passe soient stockés côté serveur et non pas côté front.

Le mot de passe est : **N3_p@s_St0ck3r_L3s_M0ts_D3_P@ss3_D@ns_L3_Fr0nt**

Exercice 3

J'ai tout d'abord essayé plusieurs injections différentes de script sans succès. Parmi mes essais, j'ai remarqué qu'une balise img fonctionnait bien. J'ai donc essayé en mettant mon script dans le 'src' mais ça ne fonctionnait pas. Du coup j'ai utilisé dans cette même balise la fonction 'onload' pour générer un alert et cela fonctionne. À chaque rafraichissement de la page, on a bien cette alerte qui s'ouvre systématiquement.
Voici le code utilisé :

```
<img src= »https
://github.githubassets.com/images/modules/site/heroes/universe-2021-logo.png »
onload= »alert('hi') » alt= »hoho » img>
```

Pour remédier à ce problème, il faut échapper les contenus dynamiques et aussi filtrer tout commentaire qui pourrait être du script.

Exercice 4

Sur la page de login, j'ai essayé un user et mot de passe bidon, « admin » « admin » mais connexion refusée. J'ai ensuite analysé l'onglet network et le Response headers. En fouillant un peu je suis tombé sur deux champs X-Psw et X-User qui contiennent les informations de connexion.

Comme pour l'exercice 2, bien que les informations de connexion soient stockées en back, il ne faut jamais les stocker en clair.

Le mot de passe est : **Jc8b&RM52AL**

Exercice 5

Examen Sécurité 06/11/2021

Julien Sibille

En tentant de me connecter, j'ai le message 'wrong user-agent'. J'ai ensuite analysé le response headers du devtools et j'ai constaté un champ « User-agent : toto ».
Après une recherche rapide sur internet, j'ai trouvé comment utiliser un user-agent personnalisé. Il faut aller dans le menu 'trois points' en haut à droite, aller dans 'more tools' puis 'network conditions'. On peut ensuite remplacer le user-agent par défaut par 'toto'. On clique ensuite sur se connecter et cela fonctionne.

Le mot de passe (user-agent) est : **toto**

Exercice 6

Aucune information particulière sur cette page de login si ce n'est que l'on constate que cela fonctionne avec une requête SQL. En saisissant un login et mot de passe au hasard, on peut donc examiner la requête dans Network, Headers puis request payload. Après plusieurs tentatives, en saisissant dans le login et le mot de passe :

```
' OR 'a'='a
```

On arrive à se connecter sans même connaitre le login et le mot de passe.

Pour remédier à ce problème, il faut utiliser des requêtes préparées.

Exercice 7

On a sur cette page html une balise script avec un contenu illisible. En cherchant rapidement sur internet, on trouve des décodeurs de javascript ce qui nous donne :

```
function anonymous( ) { a=prompt('Entrez le mot de
passe');if(a=='toto123lol'){alert('bravo');}else{alert('fail...');} }
```

Pour corriger ce problème, encore une fois il ne faut pas stocker d'informations confidentielles côté front comme un mot de passe.

Le mot de passe est : **toto123lol**