

Rapport Exercices Sécurité des Applications : Série 6

Thomas Dagier

November, 21, 2020

1 Web server, HTML - Source code (EASY1)

Sur la plateforme Root-me, des challenges sont disponibles pour nous faire découvrir les enjeux de la sécurité Web. Le premier challenge de cette catégorie nous invite à trouver un mot de passe valide. J'ai commencé par rentrer le mot de passe : 1234 pour voir ce qui change sur le site.

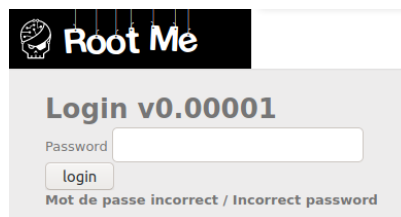


Figure 1: état de la page après avoir rentré le mauvais mot de passe

Au-delà de l'indication du mot de passe incorrect, j'ai remarqué que l'URL avait changé :

`challenge01.root-me.org/web-serveur/ch1/?password=1234`

Figure 2: URL après avoir rentré le mot de passe 1234

J'ai donc essayé de changer l'URL mais sans succès. Comme nous l'avions vu en cours, il est possible d'inspecter la page sur laquelle on se trouve pour y trouver le code HTML (touche F12):

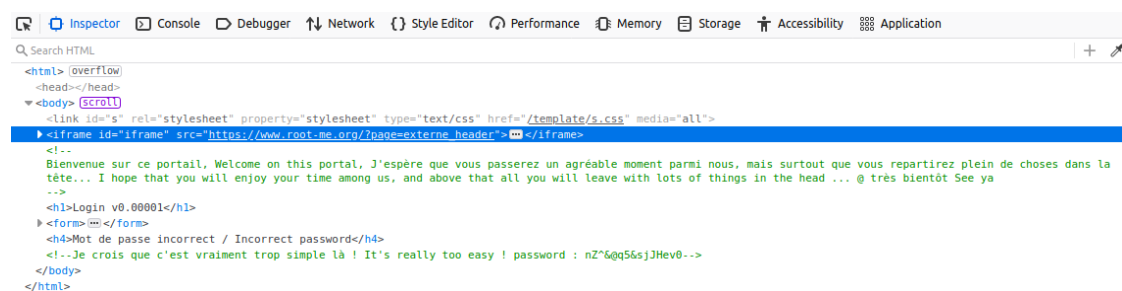
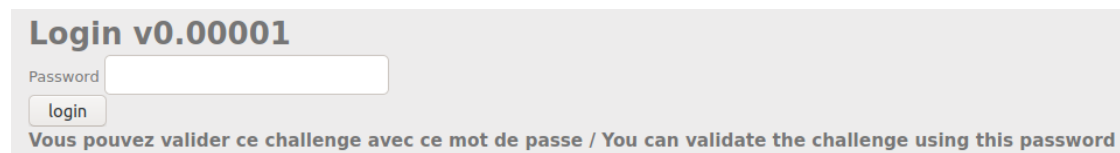


Figure 3: affichage du code source de la page (HTML)

Il se trouve que le mot de passe est indiqué directement dans le code source. On peut donc le copier et le mettre dans la textbox :



The image shows a login interface for a challenge titled "Login v0.00001". It features a "Password" label, a text input field, and a "login" button. Below the input field, a message states: "Vous pouvez valider ce challenge avec ce mot de passe / You can validate the challenge using this password".

Figure 4: entrée du bon mot de passe

Il nous est alors indiqué de rentrer ce mot de passe dans la textbox de la page précédente pour gagner les points du challenge :

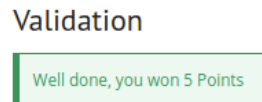


Figure 5: flag de validation

2 Web server, HTTP - Open Redirect (EASY2)

Le but du second exercice est de modifier une redirection. Sur ce challenge, on peut cliquer sur trois boutons qui nous redirigent vers les sites respectifs. Nous devons modifier le code d'un des boutons pour qu'il nous redirige sur un site différent de celui indiqué dans le bouton.

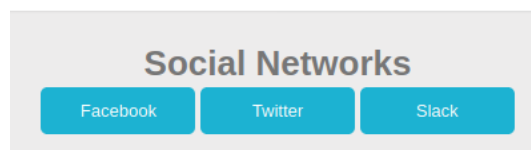


Figure 6: affichage du challenge

Admettons que l'on veuille modifier la redirection du bouton facebook. Pour cela, on peut inspecter la page et on voit quelle est la redirection faite pour Facebook :

```
<a href="?url=https://facebook.com&h=a023cfbf5f1c39bdf8407f28b60cd134">facebook</a>
```

Figure 7: URL redirigeant vers Facebook

J'ai testé de remplacer l'URL par `www.google.com` mais cela n'a pas marché. Après quelques recherches sur internet, j'ai remarqué que je n'avais pas modifié ce qui suit `&h` dans l'URL. Après avoir décrypté la suite de caractères : `a023cfbf5f1c39bdf8407f28b60cd134` j'ai remarqué que le résultat est bien `https://facebook.com`.

Il suffit donc de faire la même manipulation en sens inverse. On doit alors encoder en md5 `https://google.com` :

Md5(https://google.com) = 99999ebcfdb78df077ad2727fd00969f

Figure 8: chaîne md5 de `https://google.com`

On peut alors modifier l'URL directement sur le href :

```
<a href="?url=https://google.com&h=99999ebcfdb78df077ad2727fd00969f">facebook</a>
```

Figure 9: modification de l'URL pour la redirection du bouton Facebook

Dès lors, si on clique sur le bouton Facebook, ce dernier nous redirige vers le site de Google :

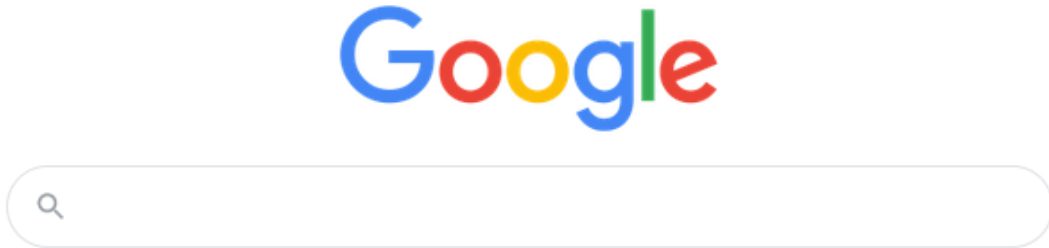


Figure 10: effet de la redirection du bouton Facebook

La redirection est active et on voit apparaitre la mention suivante :



Figure 11: flag de réussite

Avec l'obtention du flag, on peut alors valider l'exercice :

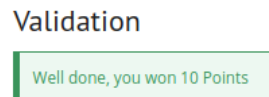


Figure 12: validation du challenge

3 Web server, HTTP - Headers (EASY3)

Dans ce troisième challenge, l'objectif est d'accéder au site en tant qu'administrateur. Sur la page du challenge, on nous indique qu'il faut faire des requêtes HTTP pour pouvoir lire le contenu du header. Sous Linux, un outil qui est déjà installé par défaut est cURL. Il permet de tester la connectivité en fonction des URL et de faire du transfert de données. On peut donc tester la connectivité :

```
thomas@thomas:~$ curl http://challenge01.root-me.org/web-serveur/ch5/  
<html>  
<body><link rel='stylesheet' property='stylesheet' id='s' type='text/css' href='/template/s.css'  
media='all' /><iframe id='iframe' src='https://www.root-me.org/?page=externe_header'></iframe>  
<p>Content is not the only part of an HTTP response!</p>  
</body>  
</html>
```

Figure 13: affichage du contenu de la réponse HTTP

Cependant, une balise `p` nous indique que ce n'est pas tout le contenu d'une réponse suite à une requête HTTP. En cherchant dans le manuel, il est mentionné le paramètre `-verbose` qui permet d'afficher plus de contenu. On peut donc tester cette commande :

```
thomas@thomas:~$ curl --verbose http://challenge01.root-me.org/web-serveur/ch5/
* Trying 2001:bc8:35b0:c166::151:80...
* Connected to challenge01.root-me.org (2001:bc8:35b0:c166::151) port 80 (#0)
> GET /web-serveur/ch5/ HTTP/1.1
> Host: challenge01.root-me.org
> User-Agent: curl/7.71.1
> Accept: */*
>
* Mark bundle as not supporting multiuse
< HTTP/1.1 200 OK
< Server: nginx
< Date: Sat, 21 Nov 2020 15:03:54 GMT
< Content-Type: text/html; charset=UTF-8
< Transfer-Encoding: chunked
< Connection: keep-alive
< Vary: Accept-Encoding
< Header-RootMe-Admin: none
<
<html>
<body><link rel='stylesheet' property='stylesheet' id='s' type='text/css' href='/template/s.css'
' media='all' /><iframe id='iframe' src='https://www.root-me.org/?page=externe_header'></iframe>
>
<p>Content is not the only part of an HTTP response!</p>
</body>
</html>
* Connection #0 to host challenge01.root-me.org left intact
```

Figure 14: affichage du contenu de la réponse complète HTTP

Arrivé là, je m'attendais à trouver le mot de passe qui permet de valider le challenge. Cependant ce n'est pas le cas. Pensant faire fausse route, je suis allé sur le forum de Root-me sur cet exercice. Il y a en effet un paramètre à modifier lors de l'envoi de la requête. Il semble assez évident que ce paramètre soit : `Header-RootMe-Admin: none`.

Toujours dans le manuel de curl, il est indiqué que l'on peut modifier un des paramètres :

```
-H, --header <header/@file>
```

Figure 15: manuel d'utilisation de curl pour le header HTTP

```
Example:
curl -H "X-First-Name: Joe" http://example.com/
```

Figure 16: exemple issu du manuel de curl pour le header

J'ai donc fini par tester la commande : `curl -H "Header-RootMe-Admin: true" http://challenge01.root-me.org/web-serveur/ch5/` :

```
thomas@thomas:~$ curl --header "Header-RootMe-Admin: true" http://challenge01.root-me.org/web-serveur/ch5/
<html>
<body><link rel='stylesheet' property='stylesheet' id='s' type='text/css' href='/template/s.css'
media='all' /><iframe id='iframe' src='https://www.root-me.org/?page=externe_header'></iframe>
<p>Content is not the only part of an HTTP response!</p>
<p>You dit it ! You can validate the challenge with the password HeadersMayBeUseful</p></body>
</html>
```

Figure 17: utilisation de la bonne commande pour trouver le flag

Cette fois-ci le message est différent et on obtient le flag de validation que l'on peut rentrer sur la page du challenge :

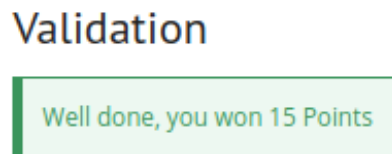


Figure 18: validation du challenge

4 Web server, Directory Traversal (EASY4)

Dans cet exercice, nous avons plusieurs index qui nous permettent de naviguer entre les sections d'une galerie d'images. Le but de l'exercice est de trouver la section cachée de cette galerie.

En basculant entre les galeries, j'ai remarqué que l'URL est modifié :

```
challenge01.root-me.org/web-serveur/ch15/ch15.php?galerie=apps
```

Figure 19: URL après navigation dans les galeries

J'ai donc essayé d'enlever complètement "apps" de l'URL pour voir où j'arrive :



Figure 20: affichage des galleries

On voit sur cette image que toutes les galleries sont présentes mais il y en a aussi une cachée. On peut donc inspecter la page pour connaître son nom et se diriger vers cette galerie :

```

```

Figure 21: affichage du nom de la galerie secrète

On peut donc rentrer ce nom à la place de "apps" supprimé dernièrement de l'URL pour voir ce que contient cette page :

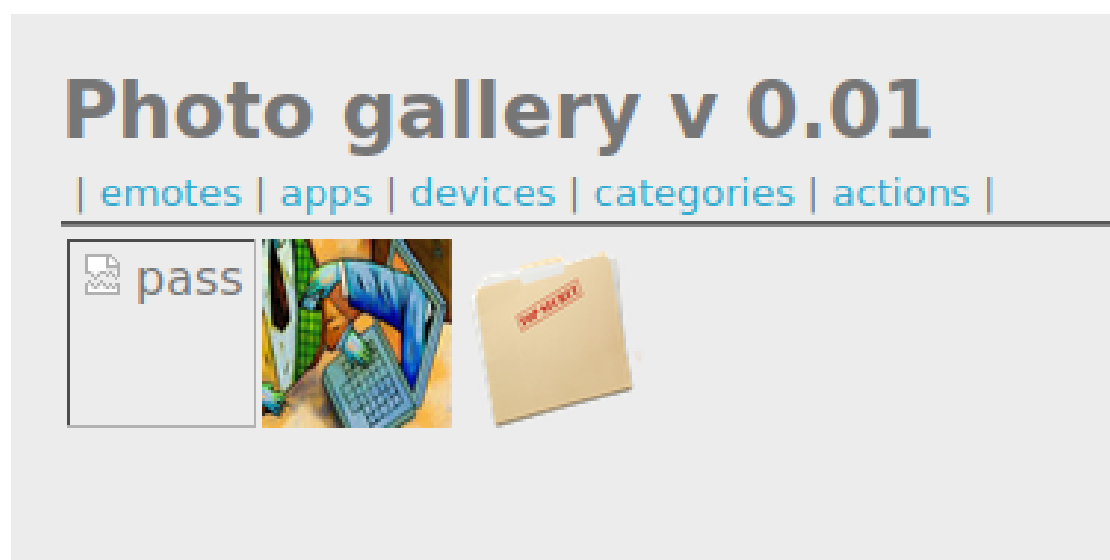


Figure 22: affichage de la galerie secrète

Il y a, sur cette page, un document texte et deux images. Je me doute que le mot de passe se situe dans ce document donc j'essaye de l'ouvrir. Cependant, il m'est impossible de voir son contenu en modifiant l'URL :

```
Warning: opendir(galerie/86hwnX2r/password.txt): failed to open dir: Not a directory in /challenge/web-serveur/ch15/ch15.php on line 34
Warning: readdir() expects parameter 1 to be resource, boolean given in /challenge/web-serveur/ch15/ch15.php on line 36
```

Figure 23: affichage de l'erreur lors de l'ouverture du fichier texte

Après avoir cherché sur internet, ceci semble être dû à l'URL et plus précisément au "?" dans "ch15.php?galerie" qui demanderait l'ouverture d'un dossier et pas d'un fichier. Cela dit, on voit bien que cela est dans le fichier ch15.php, il est donc simple de modifier l'URL pour ouvrir le fichier, ce qui donne :

```
challenge01.root-me.org/web-serveur/ch15/galerie/86hwnX2r/password.txt
```

Figure 24: URL modifié pour afficher le contenu du fichier texte

Ainsi, on peut lire le contenu du fichier texte :

kcb\$!Bx@v4Gs9Ez

Figure 25: affichage du contenu du fichier texte

Une fois le mot de passe récupéré, on peut le rentrer pour valider le challenge :

Validation

Well done, you won 25 Points

Figure 26: validation du challenge

5 Retour sur le travail

Ayant moi-même fait un site de photographie de A à Z en utilisant l'HTML, le CSS, le PHP, le SQL et le JavaScript, j'ai vraiment trouvé ces exercices intéressants. En effet ils m'ont permis de comprendre à quel point il est facile de modifier les informations et accéder à des données sensibles du site. J'ai découvert avec le dernier exercice qu'il était possible d'accéder à la page de login pour la maintenance de mon site, pourtant censée être invisible. Cela m'a beaucoup aidé à trouver où renforcer la sécurité.