UE L318 - Semaine 2

Hugo Lignères

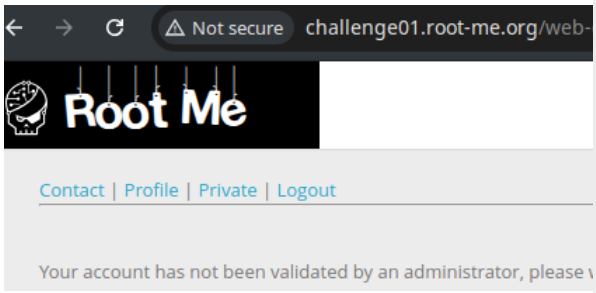
22/02/2025

Table des matières

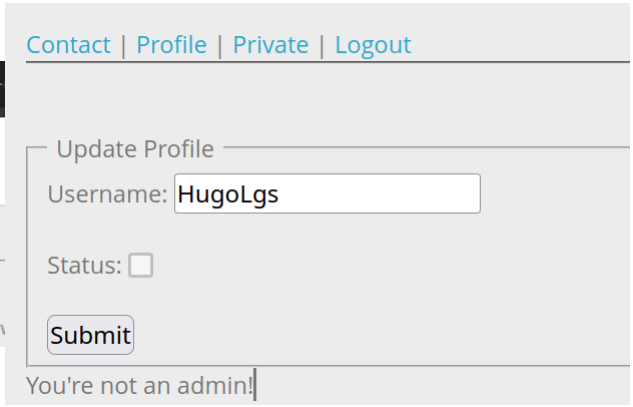
1 Partie 1 : CSRF

1.1 CSRF - 0 protection

J'ai d'abord remarqué que, une fois connecté, dans la section "private", mon compte n'était pas encore validé par un administrateur du site. Également, je ne pouvais pas modifier mon pseudo, car je ne suis pas un admin non plus. Enfin, la checkbox "Status" était désactivé.



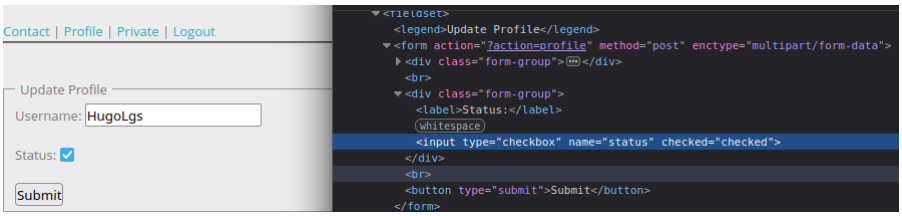
Compte non-validé



Nom d'utilisateur non-modifiable

Premières constatations

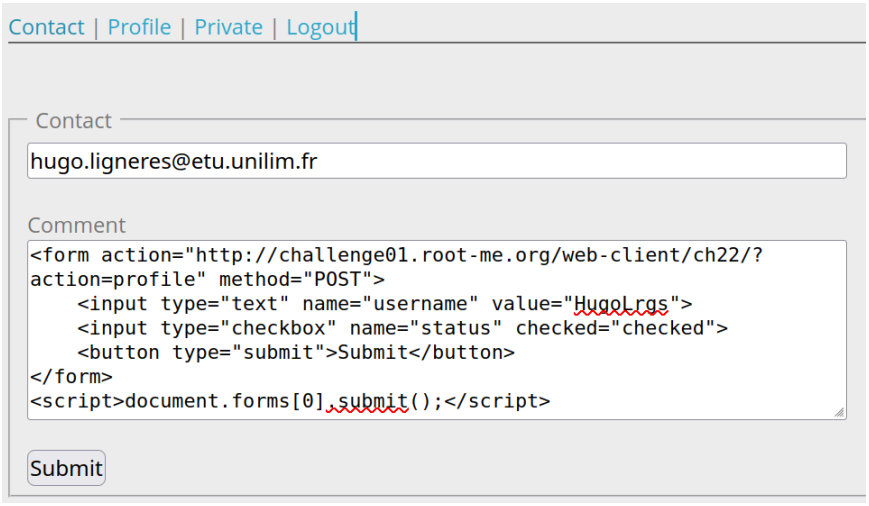
Je me suis donc dit que je devais trouver un moyen de modifier ce formulaire pour "forcer" la validation de mon compte, où accéder à un rôle me permettant de le faire. J'ai d'abord testé depuis l'inspecteur pour voir ce que j'obtenais en modifiant le code du formulaire :



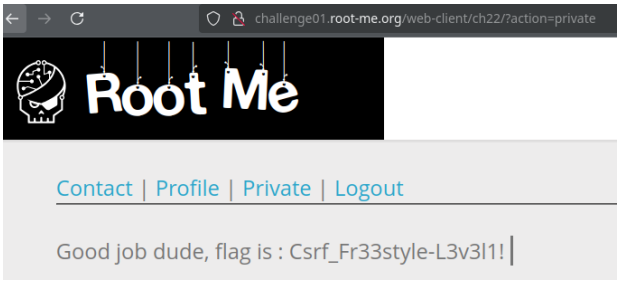
Code test

On peut voir qu'avec le `checked="checked"`, je pouvais modifier le formulaire pour m'approcher de ce que je souhaitais.
Après quelques recherches sur le forum, j'ai compris qu'il était possible de mofidier ce formulaire injectant un code modifié directement dans la page où se situe ce formulaire. Pour cela, j'ai utilisé le formulaire pour contacter un admin du site. Dans la partie du `textarea`, j'ai entré le code suivant :
J'ai pris ce code depuis le code source de la page profil, et je modifie ensuite ce dernier grâce à une injection en entrant le code suivi dans le formulaire de contact
Dans les grandes lignes, ce code envoi une requête POST à l'url de la page pour modifier le nom d'utilisateur, en modifiant la checkbox avec `checked="checked"`. Ceci me permet de modifier le code source de la page ciblée, et d'y modifier le formulaire qui s'y trouve.

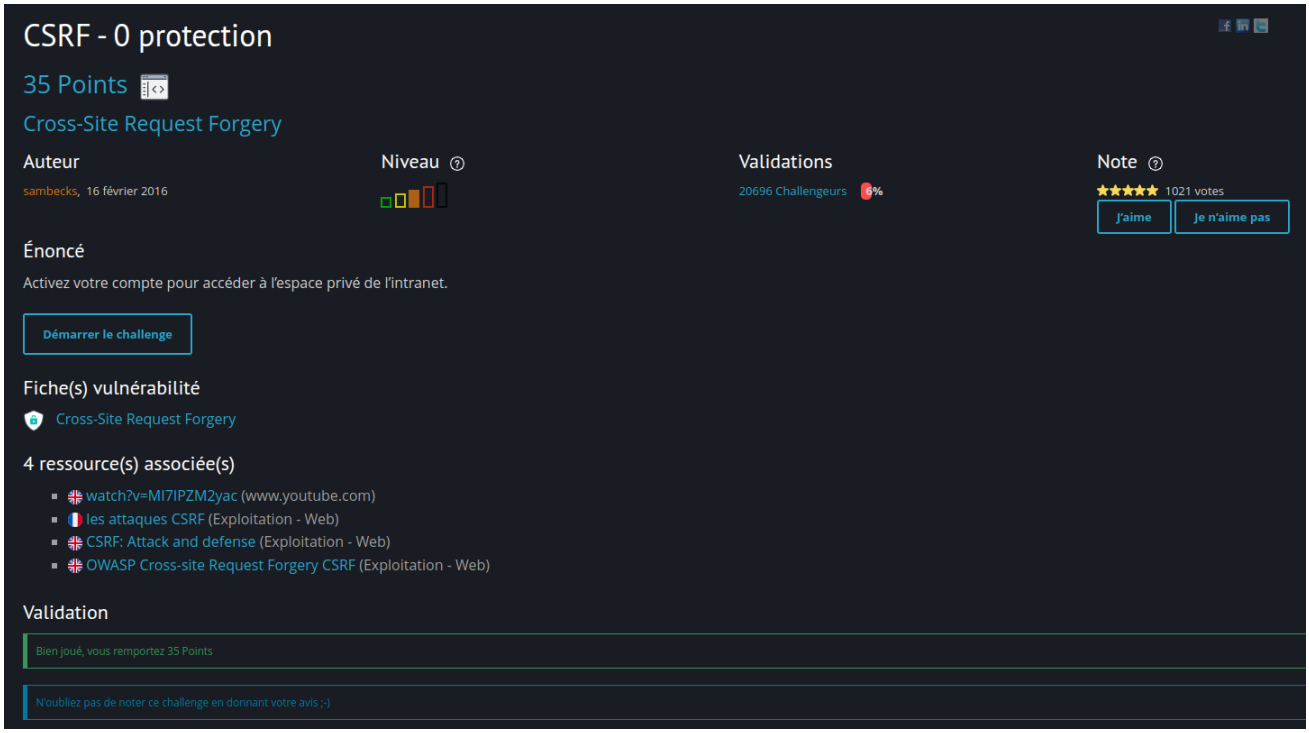
Une fois cette injection réalisée, la page "Private" a été modifiée, et le flag à entrer pour valider le challenge est affiché :
Voici la preuve que le challenge a été validé :



Code à injecter



Flag affiché



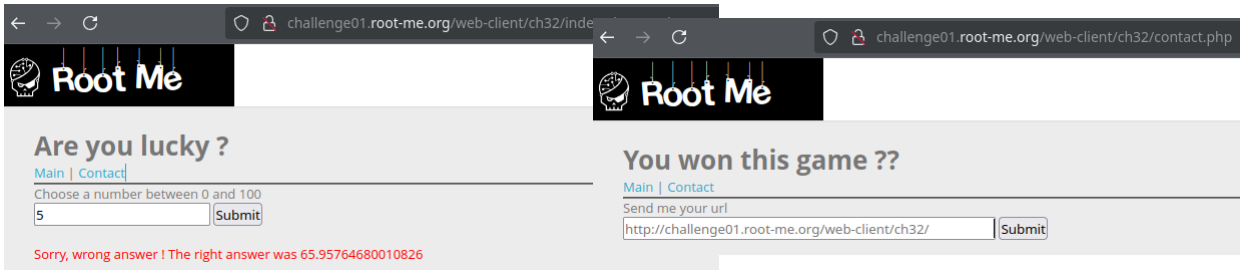
Challenge réussi

1.2 CSRF - contournement de jeton

2 Partie 2 : XSS

2.1 Challenge root-me

J'ai choisi de travailler sur le challenge **XSS DOM Based - Introduction**



Page "Main" Page "Contact"

Les pages du challenge

(présenter ce que font les 2 pages)
J'ai dans un premier temps testé l'injection de code JavaScript très simple dans le formulaire du lucky number, car celui-ci n'imposait pas de format de données à entrer. Après différentes itérations tentées, j'ai obtenu ce code qui produisait l'effet attendu : un message d'alerte qui s'affichait quand on clique sur "Submit" : `test'; alert(1);//`

Injection du code Résultat

Premier test

Maintenant que
Ensuite, webhook

2.2 Les recommandations ANSSI contre les vulnerabilités XSS

Expliquez les recommandations ANSSI contre les vulnerabilités XSS.
Encoder les données entrées, surtout celles entrées du côté client/navigateur d'une application Utiliser textContent ou l'API DOM pour manipuler le DOM, plutôt que des méthodes JavasCript qui injectent du code HTML, comme innerHTML par exemple.
Les pages HTML ne doivent pas contenir de code CSS ou JavaScript