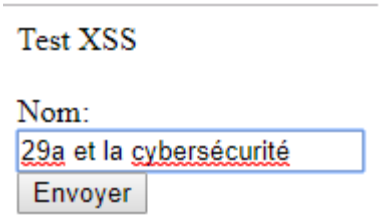
Le *cross-site scripting* (XSS)

Texte de Patrick Roberge – 29A

Dans une application web, il est primordial de valider les caractères autorisés en vue d'éviter l'injection de commandes. Les formulaires effectuant un *submit* sans valider les caractères présentent le risque de diminuer la sécurité de l'application. C'est ce qu'on appelle le *cross-site scripting* (XSS). Plusieurs sites malicieux utilisent ce genre de procédé pour faire des redirections ou du vol d'information ou encore pour effectuer des actions non autorisées.

Voici un code HTML simple contenant un formulaire :

```
<html>
        <head>
        </head>
<body>Test XSS
        <br><br>
         <form action="validateAction.php">Nom:<br>
                <input type="text" name="firstname"><br>
                <input type="submit" value="Envoyer">
        </form>
</body>
</html>
```



Ce code HTML appelle le script PHP *validateAction* ayant en paramètre le *firstname*. À l'exécution, la variable et son contenu sont passés en paramètre au script *validateAction.php*. Visualisons le résultat de la requête et le contenu du code PHP :

Résultat après l'exécution du script :

Ceci est un test

29a et la cybersécurité

Contenu du script :

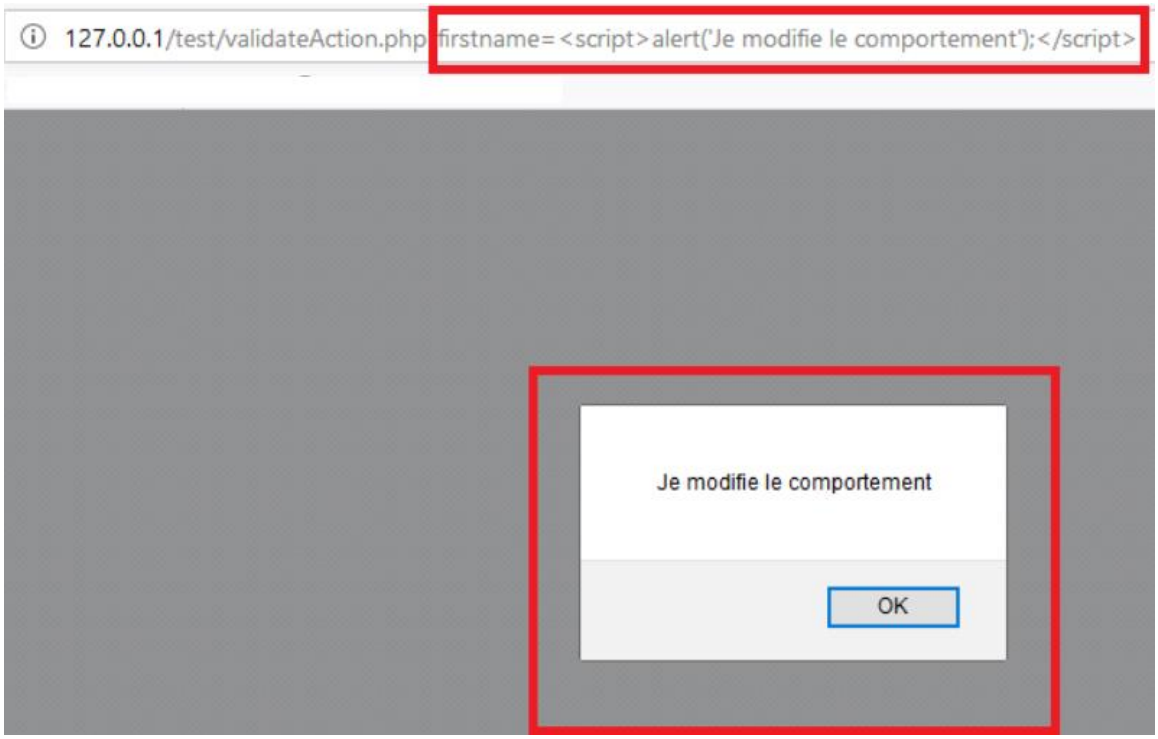
```
<html>
    <body>
    <?php echo "Ceci est un test" . "<br>";
        echo $_GET["firstname"];
    ?>
    </body>
</html>
```

Le paramètre *firstname* contient le contenu de la boîte de texte que nous avons précédemment entré. Jusque-là, tout se déroule bien! Si l'on n'effectue pas de validation, il y a un risque puisqu'il est alors possible d'injecter une ou plusieurs commandes dans la barre d'adresse.

Barre d'adresse sans injection de code :
127.0.0.1/test/validateAction.php?firstname=29a+et+la+cybersécurité

Il est donc possible de modifier les informations dans la barre d'adresse pour changer le comportement de la page web. Par exemple, nous pourrions faire afficher une alerte dans la page sans modifier directement le code source sur le serveur. Modifions la valeur du paramètre *firstname* par la valeur suivante :

<script> alert('Je modifie le comportement'); </script>



En conclusion, évitez les problèmes liés au XSS en effectuant des validations de caractères pouvant causer de tels comportements. Et, dans un monde idéal, il faut également filtrer les variables du côté du serveur pour éviter des problèmes avec le XSS.