Nettoyer les données utilisateur (sanitization)

Faille XSS

Le `cross-site scripting` est un type de faille de sécurité permettant d'injecter du contenu dans une page, provoquant ainsi des actions sur les navigateurs Web d'autres utilisateurs visitant la page.

Petit exemple :

```
<?= $_GET["test"] ?? "" ?>


<form action="" method="GET">
  <input type="text" name="test">
  <button type="submit">Submit</button>
</form>
```

`<?=` est un raccourci pour ouvrir la balise `PHP` et utiliser `echo` ?

`??` permet de ne pas avoir d'erreur si `$_GET["test"]` n'est pas défini en n'affichant rien.

Essayez d'entrer :

```
<script>alert('Hack')</script>
```

Cela exécutera le script dans votre navigateur.

Cette faille ouvre en fait la possibilité à des attaquants de faire exécuter n'importe quel code `HTML` ou `JavaScript` par d'autres visiteurs de votre site.

L'attaquant va mettre du code par exemple dans une image de profil, un commentaire de forum, un poste de blog etc. Si vous ne nettoyez pas suffisamment toutes les données envoyées et sauvegardées pour être affichées à d'autres utilisateurs, vous serez vulnérables à ces attaques.

Elles permettent facilement de voler des données.

Bien sûr, il existe des attaques bien plus complexes et ce n'est qu'un exemple pour montrer l'importance de **nettoyer toutes les données provenant d'utilisateurs**.

La désinfection (`sanitization`)

La désinfection (`sanitization`) consiste à enlever ou traiter tous les caractères provenant d'utilisateurs et pouvant provoquer des failles de sécurité.

`PHP` propose un grande nombre de filtres très utiles pour cette étape. Ils commencent tous par `FILTER_SANITIZE_`.

Pour cette leçon, la table `ASCII` vous sera utile :

ASCII TABLE

Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char
0	0	[NULL]	32	20	[SPACE]	64	40	@	96	60	`
1	1	[START OF HEADING]	33	21	!	65	41	A	97	61	a
2	2	[START OF TEXT]	34	22	"	66	42	B	98	62	b
3	3	[END OF TEXT]	35	23	#	67	43	C	99	63	c
4	4	[END OF TRANSMISSION]	36	24	$	68	44	D	100	64	d
5	5	[ENQUIRY]	37	25	%	69	45	E	101	65	e
6	6	[ACKNOWLEDGE]	38	26	&	70	46	F	102	66	f
7	7	[BELL]	39	27	'	71	47	G	103	67	g
8	8	[BACKSPACE]	40	28	(	72	48	H	104	68	h
9	9	[HORIZONTAL TAB]	41	29	)	73	49	I	105	69	i
10	A	[LINE FEED]	42	2A	*	74	4A	J	106	6A	j
11	B	[VERTICAL TAB]	43	2B	+	75	4B	K	107	6B	k
12	C	[FORM FEED]	44	2C	,	76	4C	L	108	6C	l
13	D	[CARRIAGE RETURN]	45	2D	-	77	4D	M	109	6D	m
14	E	[SHIFT OUT]	46	2E	.	78	4E	N	110	6E	n
15	F	[SHIFT IN]	47	2F	/	79	4F	O	111	6F	o
16	10	[DATA LINK ESCAPE]	48	30	0	80	50	P	112	70	p
17	11	[DEVICE CONTROL 1]	49	31	1	81	51	Q	113	71	q
18	12	[DEVICE CONTROL 2]	50	32	2	82	52	R	114	72	r
19	13	[DEVICE CONTROL 3]	51	33	3	83	53	S	115	73	s
20	14	[DEVICE CONTROL 4]	52	34	4	84	54	T	116	74	t
21	15	[NEGATIVE ACKNOWLEDGE]	53	35	5	85	55	U	117	75	u
22	16	[SYNCHRONOUS IDLE]	54	36	6	86	56	V	118	76	v
23	17	[ENG OF TRANS. BLOCK]	55	37	7	87	57	W	119	77	w
24	18	[CANCEL]	56	38	8	88	58	X	120	78	x
25	19	[END OF MEDIUM]	57	39	9	89	59	Y	121	79	y
26	1A	[SUBSTITUTE]	58	3A	:	90	5A	Z	122	7A	z
27	1B	[ESCAPE]	59	3B	;	91	5B	[	123	7B	{
28	1C	[FILE SEPARATOR]	60	3C	<	92	5C	\	124	7C	|
29	1D	[GROUP SEPARATOR]	61	3D	=	93	5D	]	125	7D	}
30	1E	[RECORD SEPARATOR]	62	3E	>	94	5E	^	126	7E	~
31	1F	[UNIT SEPARATOR]	63	3F	?	95	5F	_	127	7F	[DEL]

Liste des filtres applicables

La liste exhaustive de ces filtres est :

`FILTER_SANITIZE_EMAIL` : ce filtre supprime tous les caractères sauf les lettres, chiffres, et `!#$%&'*+-=?^_`{|}~@.[]`. C'est-à-dire tous les caractères admis dans la spécification des emails. Pas de modificateur.

`FILTER_SANITIZE_ENCODED` : ce filtre permet d'encoder les caractères au format `URL` et de supprimer ou d'encoder les caractères spéciaux (suivant le ou les modificateurs utilisés : `FIL`

TER_FLAG_STRIP_LOW, FILTER_FLAG_STRIP_HIGH, FILTER_FLAG_STRIP_BACKTICK, FILTER_FLAG_ENCODE_LOW, FILTER_FLAG_ENCODE_HIGH).

`FILTER_SANITIZE_MAGIC_QUOTES` : ce filtre permet d'échapper avec des antislashs les guillemets simples et doubles, les antislashs et NUL. Pas de modificateur.

`FILTER_SANITIZE_NUMBER_FLOAT` : ce filtre supprime tous les caractères qui ne sont pas des chiffres saut `+ - . ,` (suivant les modificateurs appliqués : `FILTER_FLAG_ALLOW_FRACTION, FILTER_FLAG_ALLOW_THOUSAND, FILTER_FLAG_ALLOW_SCIENTIFIC`).

`FILTER_SANITIZE_NUMBER_INT` : ce filtre supprime tous les caractères qui ne sont pas des chiffres et `+` et `-`. Pas de modificateur.

`FILTER_SANITIZE_SPECIAL_CHARS` : ce filtre transforme en entité `HTML` les caractères `'"<>&` et les caractères `ASCII` de valeur inférieur à 32 (cf table `ASCII`), et supprime ou encode les autres caractères spéciaux. Les modificateurs applicables sont `FILTER_FLAG_STRIP_LOW, FILTER_FLAG_STRIP_HIGH, FILTER_FLAG_STRIP_BACKTICK, FILTER_FLAG_ENCODE_HIGH`.

`FILTER_SANITIZE_FULL_SPECIAL_CHARS` : ce filtre convertit les caractères spéciaux en entités `HTML`. Ce filtre équivaut à l'application de la fonction `htmlspecialchars()` avec `ENT_QUOTES` activé. Par exemple, `<` est converti en `<`. Le modificateur applicable est `FILTER_FLAG_NO_ENCODE_QUOTES`.

`FILTER_SANITIZE_URL` : ce filtre supprime tous les caractères sauf ceux admis dans les `URL` : les lettres, chiffres et `$-_.+!*'(),{}|\\^~[]`<>#%";/?:@&=`. Pas de modificateur.

`FILTER_UNSAFE_RAW` : ne fait rien par défaut. Les modificateurs applicables sont `FILTER_FLAG_STRIP_LOW, FILTER_FLAG_STRIP_HIGH, FILTER_FLAG_STRIP_BACKTICK, FILTER_FLAG_ENCODE_LOW, FILTER_FLAG_ENCODE_HIGH, FILTER_FLAG_ENCODE_AMP`.

Liste des modificateurs applicables au nettoyage

Les modificateurs, ou `flags`, permettent de modifier le comportement des filtres.

Les `flags` ne sont pas applicables à tous les filtres. Il faut vérifier la liste des modificateurs applicables pour chaque filtre plus haut.

`FILTER_FLAG_STRIP_LOW` : supprime les caractères dont la valeur numérique est inférieure à 32 (cf table `ASCII`).

`FILTER_FLAG_STRIP_HIGH` : supprime les caractères dont la valeur numérique est supérieure à 127 (cf table `ASCII`). C'est-à-dire tous les caractères accentués par exemple.

`FILTER_FLAG_STRIP_BACKTICK` : supprime tous les accents graves `` ` ``.

`FILTER_FLAG_ALLOW_FRACTION` : autorise les points comme séparateurs pour les nombres.

`FILTER_FLAG_ALLOW_THOUSAND` : autorise les virgules comme séparateurs pour les nombres.

`FILTER_FLAG_ALLOW_SCIENTIFIC` : autorise les caractères `e` et `E` pour la notation scientifique des nombres.

`FILTER_FLAG_NO_ENCODE_QUOTES` : les guillemets simples et doubles ne seront pas encodés.

`FILTER_FLAG_ENCODE_LOW` : encode les caractères dont la valeur numérique est inférieure à 32 (cf table `ASCII`).

`FILTER_FLAG_ENCODE_HIGH` : encode les caractères dont la valeur numérique est supérieure à 127 (cf table `ASCII`).

`FILTER_FLAG_ENCODE_AMP` : encode les esperluètes `&`.

Nous verrons les autres filtres applicables à la validation.

Utilisation de `filter_var()`

La fonction `filter_var()` permet de filtrer une variable, passé en premier argument, avec un filtre passé en deuxième argument.

Vous pouvez optionnellement, **préciser un ou plusieurs modificateurs (`flags`) en troisième argument**.

Voici quelques exemples pour filtrer une chaîne de caractères suivant les filtres et les marqueurs :

```
<?php
$var = 'test<>"/ é$€';

echo filter_var($var, FILTER_SANITIZE_SPECIAL_CHARS),  PHP_EOL; //
test&#60;&#62;&#34;/ é$€
echo filter_var($var, FILTER_SANITIZE_FULL_SPECIAL_CHARS),  PHP_EOL; //
test&lt;&gt;&quot;/ &eacute;$&euro;
```

Ne lancez pas le serveur de développement car vous ne verrez pas l'encodage des caractères. En effet, votre navigateur va les décoder et vous ne pourrez voir la différence.

Lancez directement avec `php index.php`.