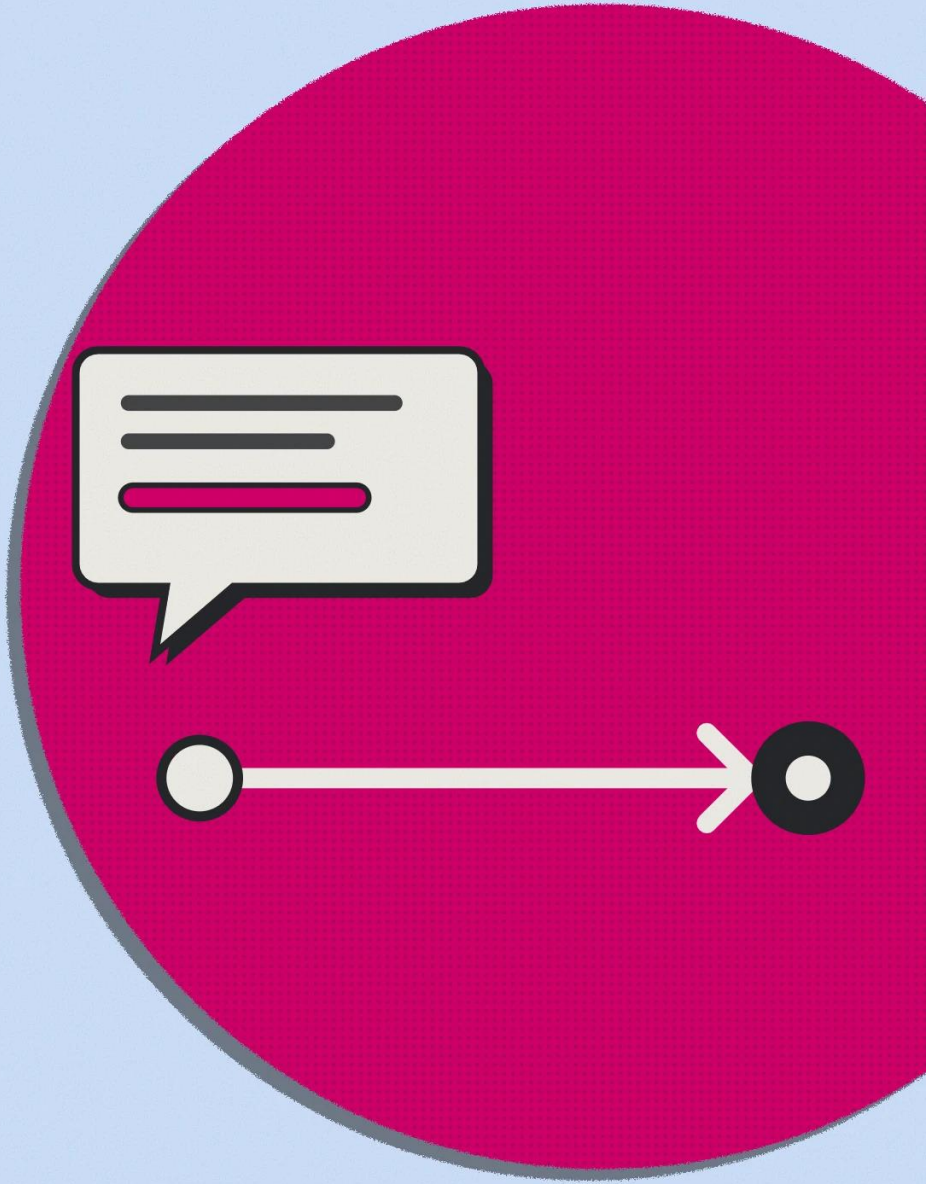
2.05

TEMPS 2 · LE TAMAGOTCHI TROUÉ · SÉANCE 1

Du code dans un message

Thibault Six · IFOSUP Wavre · 2026-2027



? Samy Kamkar glisse du JavaScript dans son profil MySpace. Que se passe-t-il chez ses visiteurs ?

Le code de Samy se recopiait chez chaque visiteur : plus d'un million de comptes sont touchés.

4 octobre 2005

un ver : un code qui se recopie de profil en profil



du JavaScript caché dans le profil

recopie ce code sur le sien sans le savoir

plus d'un million de comptes

en une vingtaine d'heures

MySpace doit couper le site

tout texte écrit par un utilisateur doit être traité comme du texte, jamais comme du code, au moment de l'afficher

Le Tamagotchi affiche aujourd'hui les messages du livre d'or sans cette précaution.

Suivez la ligne : chaque profil visité reçoit le code, qui repart chez les visiteurs suivants. Le chiffre dit la vitesse.

? Sous le livre d'or, la page écrit « Le gras est permis ». Comment un message devient-il gras ?

La page affiche chaque message par `v-html` : la balise `<b>` de Mallory devient du vrai gras.

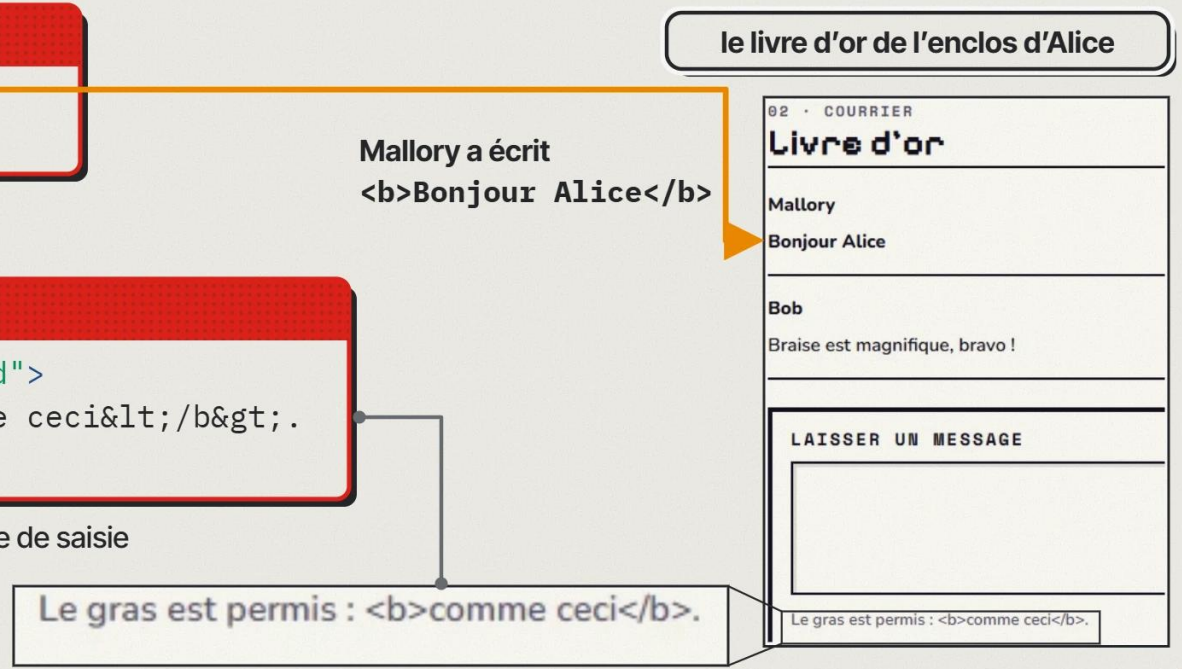
resources/js/pages/enclosures/Show.vue

```
<div v-html="message.body" />
```

`v-html` est une directive de Vue qui insère son contenu comme du HTML

```
<p class="text-sm text-muted-foreground">
    Le gras est permis : &lt;b&gt;comme ceci&lt;/b&gt;.
</p>
```

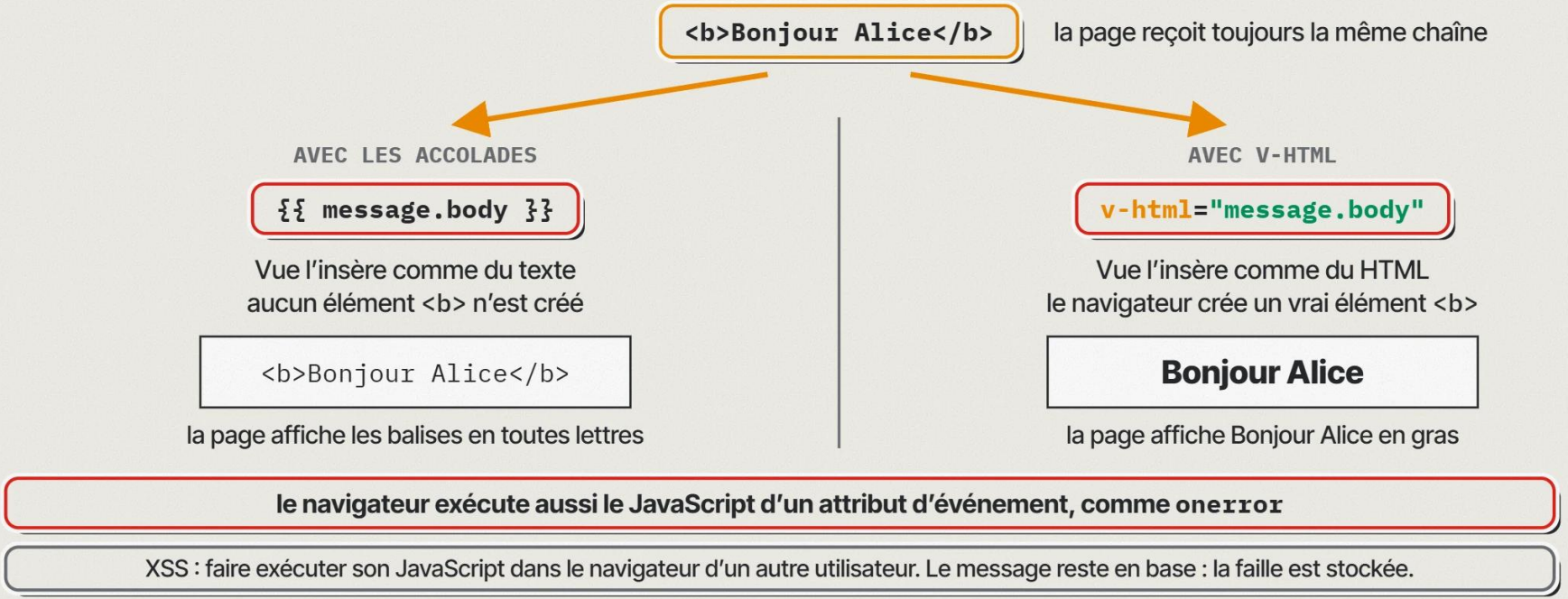
le formulaire du livre d'or l'annonce lui-même, sous la zone de saisie



Suivez la flèche : la ligne de `Show.vue` insère le message comme du HTML, et le navigateur en fait du gras.

? La même chaîne Bonjour Alice passe par les doubles accolades ou par v-html. Que voit-on ?

Les accolades affichent la chaîne en texte ; avec `v-html`, le navigateur en fait des éléments.



Même chaîne, deux écritures. À gauche, des caractères affichés ; à droite, des éléments construits par le navigateur, et le code qu'ils portent.

? Un <script> collé dans le message ne fait rien. Quelle balise Mallory écrit-elle à la place ?

Mallory poste une image sans source : son attribut `onerror` affiche un bandeau rouge chez Alice.

~~<script>~~

`v-html` passe par `innerHTML`, et un `<script>` ajouté par `innerHTML` ne se lance jamais

```
<img src=x onerror="/* du JavaScript ici */">
```

`src=x` ne pointe vers aucune image : le navigateur échoue, et lance `onerror`

Mallory colle le message à bandeau dans le livre d'or d'Alice, `tamagotchi.test/enclos/1`



Envoyer

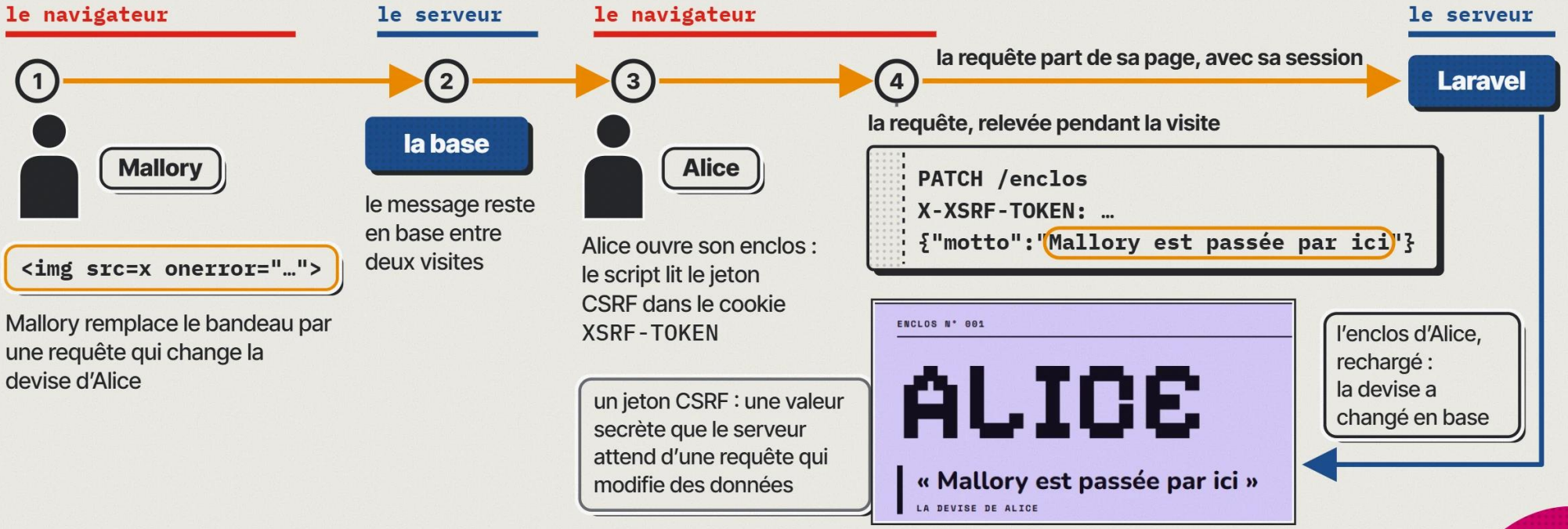
Alice ouvre son enclos : elle n'a rien cliqué



À gauche, ce que Mallory envoie ; à droite, ce qu'Alice voit en ouvrant sa page, sans avoir rien cliqué.

? Un bandeau n'est qu'une démonstration. Qu'est-ce que le même onerror peut lancer au nom d'Alice ?

Le même `onerror` envoie un `PATCH` avec la session d'Alice : sa devise change sans qu'elle clique.



Suivez la ligne : le message dort en base, puis s'exécute chez Alice. Le serveur traite la requête comme une action d'Alice.

? La charge, le script caché dans le message de Mallory, peut-elle agir au nom d'Alice ?

Le script ne peut pas lire le cookie de session d'Alice, mais il agit avec tous ses droits.

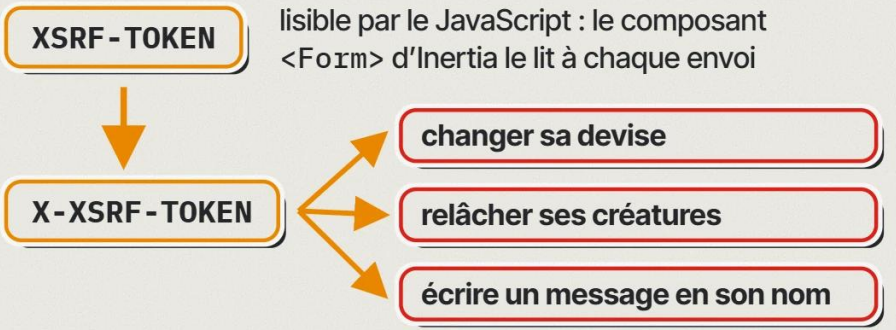
le navigateur d'Alice

CE QUE LE SCRIPT NE FAIT PAS



le JavaScript ne peut pas le lire
mais il part avec chaque requête de la page d'Alice

CE QU'IL FAIT



Le jeton CSRF ne protège pas contre un script qui s'exécute déjà sur la page.

Une faille XSS vaut les droits de celui qui lit le message.

À gauche, la seule porte fermée ; à droite, tout ce que la session ouverte d'Alice autorise, sans qu'elle clique.

? Les doubles accolades affichent du texte. Que faut-il changer dans Show.vue pour s'en servir ?

Dans Show.vue, changez le conteneur v-html du message et supprimez l'invitation au gras.

resources/js/pages/enclosures/Show.vue

```
<div class="message-body">
    <div v-html="message.body" />
</div>
```

Un message est écrit par un visiteur, donc c'est du texte.

~~ ~~
~~Le gras est permis : comme ceci.~~
~~ ~~

l'invitation « Le gras est permis » n'a plus lieu d'être

extrait

le style .message-body prévoyait du HTML libre : que garder ?



quelle balise remplace ce conteneur, avec les accolades ?

CE QUE VOUS DEVEZ OBTENIR

Bonjour Alice

Bonjour Alice

aucun élément n'est créé, et les balises s'affichent en toutes lettres

Prenez dix minutes avant d'ouvrir la correction.

La ligne cerclée est la faille. L'écriture, vous la connaissez depuis la slide 3 : cherchez où et comment l'appliquer.

? Votre portfolio affiche le contenu de vos projets avec v-html. Faut-il le changer aussi ?

`v-html` convient quand vous seul écrivez ; le HTML d'un visiteur se nettoie côté serveur.

qui écrit le HTML affiché ?

un seul auteur : l'administrateur connecté, c'est-à-dire vous

votre portfolio : `v-html` reste acceptable

un visiteur : un commentaire ou un message, montré à d'autres

nettoyez-le côté serveur avant de l'enregistrer : une liste blanche de balises sûres

symfony/html-sanitizer — extrait

```
$config = (new HtmlSanitizerConfig())->allowElement('b')->allowElement('p');
$sanitizer = new HtmlSanitizer($config);

echo $sanitizer->sanitize('<b>Bonjour</b> <img src=x onerror="alert(1)">');
```

`<b>Bonjour</b>`

le gras passe, l'image piégée disparaît

Laravel ne fournit pas de nettoyeur. Le Tamagotchi n'en a pas besoin, puisque sa correction affiche le message comme du texte.

Le choix dépend de qui écrit. Un visiteur qui écrit du HTML montré à d'autres impose le nettoyage, avant l'enregistrement.

? Qu'est-ce qui doit être vrai à la fin ?

À vous : rejouez l'attaque, corrigez `Show.vue`, puis prouvez que le message est devenu du texte.

À FAIRE

1. Connecté comme Mallory, postez le message à bandeau dans l'enclos d'Alice, `tamagotchi.test/enclos/1`.
2. Dans une fenêtre privée, ouvrez l'enclos comme Alice : le bandeau apparaît.
3. Postez le message qui change la devise, puis rechargez deux fois l'enclos d'Alice.
4. Corrigez `enclosures/Show.vue` : cherchez dix minutes avant d'ouvrir la correction.

À VÉRIFIER

```
$ php artisan test --filter=Flaw05
   PASS  Tests\Feature\Flaws\Flaw05XssTest
  ✓ the guest book does not display messages as HTML

  Tests:    1 passed (1 assertions)
```

`php artisan test` : le nombre de tests rouges passe de dix-sept à seize

```
<img src=x onerror="var t=decodeURIComponent(document.cookie.match(/XSRF-TOKEN=([^;]+)/)[1]);fetch('/enclos',{method:'PATCH',headers:{'Content-Type':'application/json','X-XSRF-TOKEN':t},body:JSON.stringify({motto:'Mallory est passée par ici'})})">
```

l'attaque rejouée : le message piégé s'affiche comme du texte, sans rien exécuter

DANS VOTRE CARNET DES FAILLES

Tout contenu écrit par un utilisateur s'affiche comme du texte. Si vous voulez vraiment du HTML, nettoyez-le côté serveur avant de l'enregistrer.

Quatre étapes à gauche, les preuves à droite. Recopiez la règle et notez où elle s'applique dans votre portfolio.