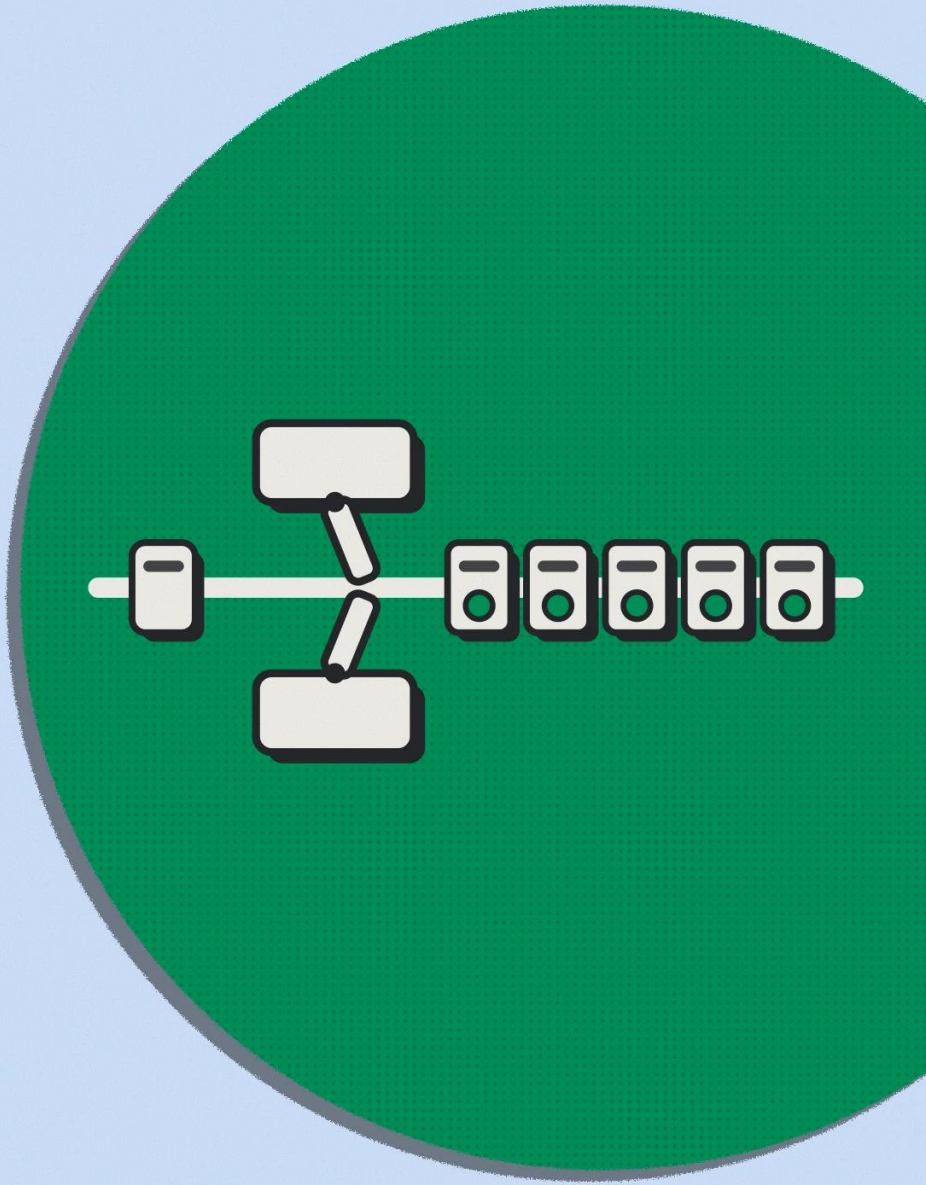
2.10

TEMPS 2 · LE TAMAGOTCHI TROUÉ · SÉANCE 2

La limitation de débit



Thibault Six · IFOSUP Wavre · 2026-2027

? Un programme essaie en masse des mots de passe déjà fuités ailleurs. Combien de comptes ouvre-t-il ?

Des mots de passe fuités ailleurs, essayés en masse, ont ouvert environ 14 000 comptes.

D'AVRIL À SEPTEMBRE 2023

adresse et mot de passe
adresse et mot de passe
adresse et mot de passe

des mots de passe déjà fuités ailleurs

un programme essaie en masse

14 000 comptes

6,9 millions de personnes

la fonction qui montre les parents génétiques d'un client

bourrage d'identifiants : un programme essaie en masse des couples adresse et mot de passe, et garde ceux qui marchent

2,31 millions de livres
l'authentification à deux facteurs n'y était pas obligatoire

LE SITE PEUT

- compter les essais et les limiter
- repérer les tentatives inhabituelles
- demander une seconde preuve d'identité

Un humain essaie trois mots de passe et abandonne ; un programme en essaie des milliers. Ce chapitre installe le premier des trois freins.

? Un code faux reçoit un message d'erreur, et rien d'autre. Combien d'essais Mallory peut-elle faire ?

Rien ne compte les essais, et Mallory peut essayer les dix mille codes de quatre chiffres.

```
routes/web.php | Route::post('codes-cadeau', [GiftCodeController::class, 'store'])->name('gift-codes.store');
```



Le seul portillon est `auth` : toute personne connectée passe, autant de fois qu'elle veut. Trois codes sur dix mille existent.

? Taper dix mille codes à la main prendrait une journée. Comment Mallory les essaie-t-elle tous ?

Une boucle de `fetch` dans la Console essaie les codes un à un et trouve 1996 en 27 minutes.

```
const token = decodeURIComponent(document.cookie.match(/XSRF-TOKEN=([^;]+)/)[1]);
```

une requête POST qui ne porte pas ce jeton reçoit le statut 419

l'onglet Console — extrait

```
for (let i = 0; i < 10000; i++) {   transforme 7 en 0007
    const code = String(i).padStart(4, '0');
    const response = await fetch('/codes-cadeau', {
        method: 'POST',
        headers: { 'X-XSRF-TOKEN': token },
        body: new URLSearchParams({ code }),
```

Code trouvé : 1996

1 997 codes, de 0000 à 1996, en 27 minutes et 41 secondes

environ 72 essais par minute

SOLDE 30,00 €

son solde est passé de 5,00 € à 30,00 €

À VOUS : LA BOUCLE COURTE

5 min

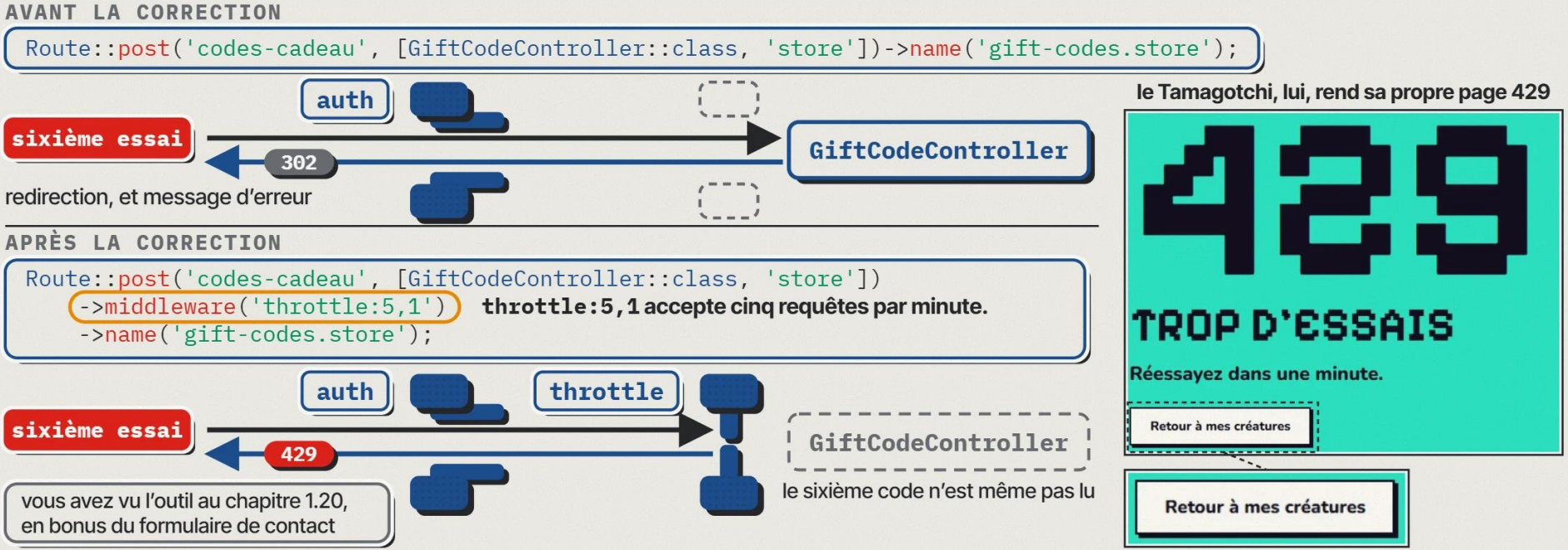
`let i = 1990; i < 2000`

La boucle s'arrête au septième essai

La boucle envoie chaque code comme le formulaire, sans jamais être arrêtée. Sur votre copie, partez de 1990 : sept essais suffisent.

? Au sixième essai dans la minute, qui doit répondre : le contrôleur, ou quelqu'un avant lui ?

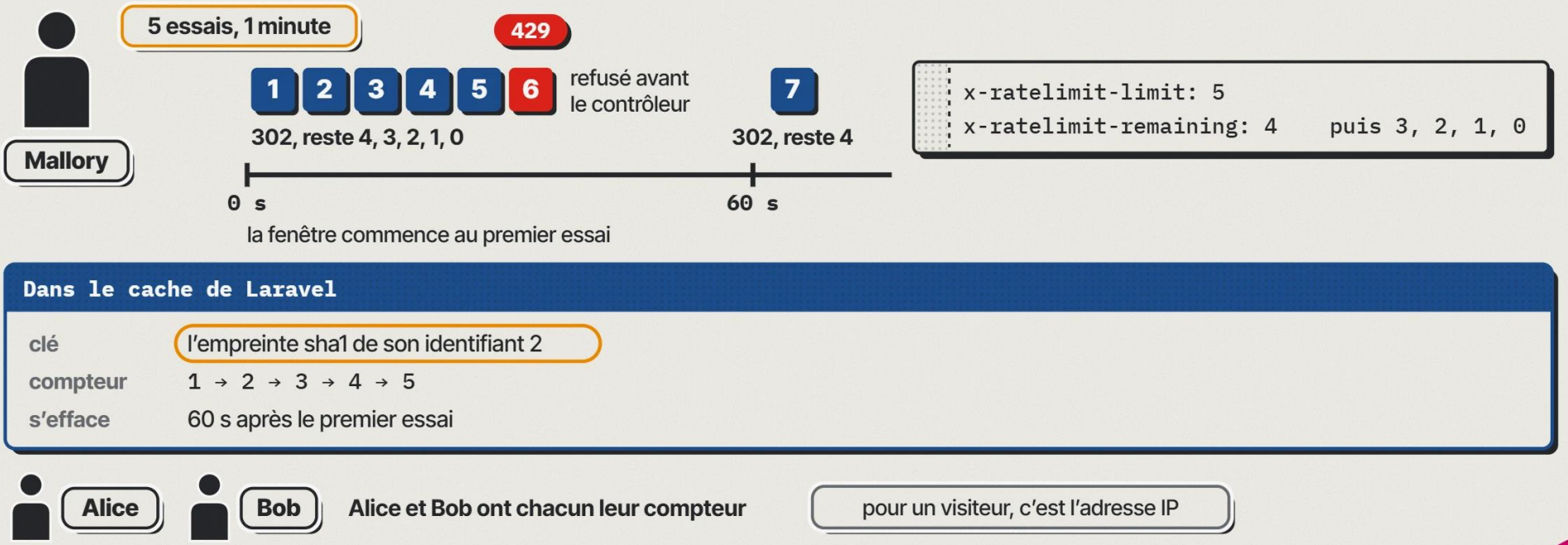
Le middleware `throttle` se pose sur la route : au sixième essai dans la minute, il répond 429.



Même requête, deux trajets. En bas, le portillon `throttle` arrête le sixième essai avant le contrôleur, et Mallory voit la page 429.

? Le middleware throttle compte les essais. Pendant combien de temps, et pour qui ?

Chaque joueur connecté a son propre compteur : cinq requêtes dans une fenêtre d'une minute.



Cinq essais passent, le sixième est refusé jusqu'à la fin de la minute. Le compteur suit le joueur : Alice et Bob gardent le leur.

? Vous n'avez rien écrit pour la page de connexion. Peut-on y essayer mille mots de passe ?

Fortify limite déjà la connexion : cinq essais par minute pour un même e-mail et une même IP.

Fortify déclare ses limites dans `config/fortify.php`

config/fortify.php

```
'limiters' => [
    'login' => 'login',
    'two-factor' => 'two-factor',
    'passkeys' => 'passkeys',
],
```

chaque nom renvoie à un limiteur

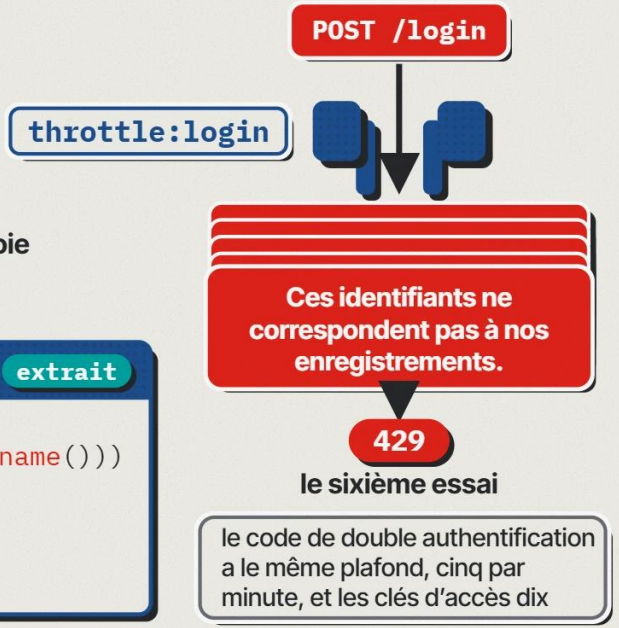
app/Providers/FortifyServiceProvider.php — extrait

```
RateLimiter::for('login', function (Request $request) {
    $throttleKey = Str::transliterate(Str::lower($request->input(Fortify::username()))
        .'|'.$request->ip());

    return Limit::perMinute(5)->by($throttleKey);
});
```

cinq essais par minute

l'adresse e-mail tapée et l'adresse IP



Le nom `login` mène au limiteur de Fortify : cinq essais par minute pour une même adresse e-mail et une même adresse IP.

? Avec cinq essais par minute, combien de temps faut-il à Mallory pour trouver 1996 ?

La limite ralentit l'attaque sans l'empêcher : trouver 1996 passe de 28 minutes à 6 h 40.

Dix mille codes, de 0000 à 9999

Sans limite — **≈ 72 essais par minute**

2 h 20 : tous les codes (calcul)

1996 trouvé après 28 minutes, mesuré sur une copie du jeu

Avec `throttle:5,1` — **5 essais par minute**

33 h 20 : tous les codes

1996 trouvé après **6 h 40**, si la boucle attend chaque fenêtre

La limite suit le compte dix comptes, dix compteurs : 3 h 20 au lieu de 33 h 20

Les deux barres sont à la même échelle. La limite ralentit l'attaque, mais dix comptes ont dix compteurs.

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

Le sixième essai reçoit 429, tous les tests passent, et le solde de Mallory reste à 5,00 €.

1. `php artisan test --filter=Flaw10` → `Tests: 1 passed (11 assertions)`
2. `php artisan test` → `Tests: 89 passed (368 assertions)`
3. `composer test` → `[OK] No errors`

`composer test` passe de bout en bout : Pint, PHPStan, puis les 89 tests

4. `php artisan migrate:fresh --seed`

reconnectez-vous comme Mallory
Collez ensuite la boucle courte, de 1990 à 2000

→ 429



son solde affiche 5,00 €

Toute route qui vérifie un secret ou qui envoie quelque chose reçoit un `throttle`.
Une limite ralentit les essais, et son efficacité dépend de la longueur du secret et des moyens de l'attaquant.

Chaque étape à gauche, sa preuve à droite. La boucle courte, rejouée sur une base neuve, voit 1996 refusé sans être lu.