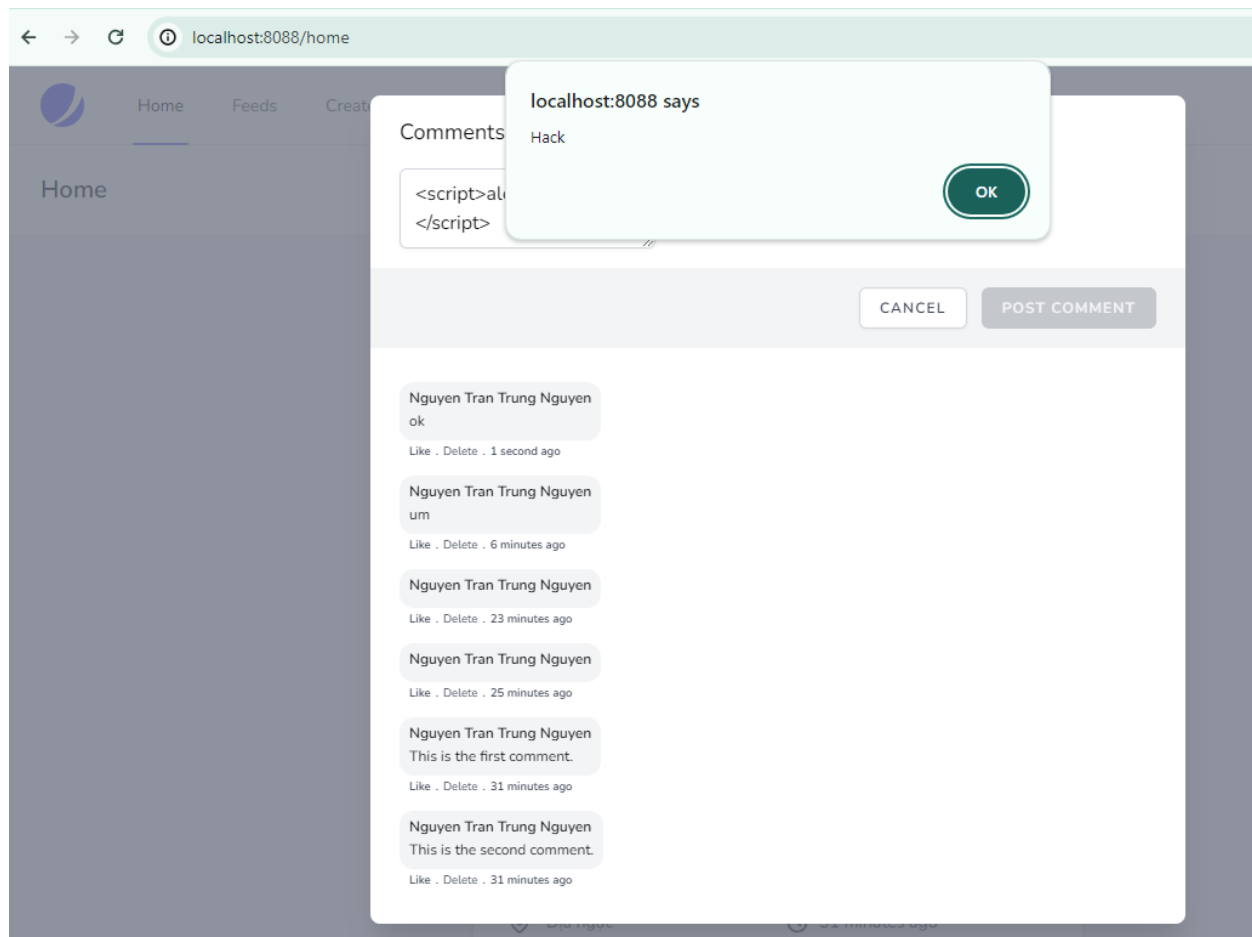


1. XSS Comment in the POST

```
<x-slot name="comments">
@forelse($comments as $comment)
<div class="flex space-x-2 my-3">
  <div class="block">
    <div class="bg-gray-100 w-auto rounded-xl px-2 pb-2">
      <div class="font-medium">
        <a href="#" class="hover:underline text-sm">
          <span class="text-xs font-semibold">{{ $comment->user->name }}</span>
        </a>
      </div>
      <div class="text-xs">
        {!! $comment->comment !!}
      </div>
    </div>
    <div class="flex justify-start items-center text-xs w-full">
      <div class="font-semibold text-gray-700 px-2 flex items-center justify-center space-x-1">
        <a href="#" class="hover:underline">
          <small>Like</small>
        </a>
        <small class="self-center">.</small>
        <button class="" wire:click="deleteComment({{ $post->id }}, {{ $comment->id }})">
          <small>Delete</small>
        </button>
        <small class="self-center">.</small>
        <a href="#" class="hover:underline">
          <small>{{ \Carbon\Carbon::parse($comment->created_at)->diffForHumans() }}</small>
        </a>
      </div>
    </div>
  </div>
</div>
@empty
  No Comments found
@endforelse
</x-slot>
```

Chúng ta có thể thấy dòng “**{!! \$comment->comment !!}**” là dòng gây ra lỗi hổng XSS. Như ta được biết thì trong livewire cho phép chúng ta thực thi các script nếu chúng ta bật nó lên bằng cách thêm **!!** vào.



Ta có thể thấy rằng các tag script không bị filter bởi vì nó đã được gỡ khi mà có !! phía trước

Cách vá

```

35     <x-slot name="comments">
36       @forelse($comments as $comment)
37     <div class="flex space-x-2 my-3">
38       <div class="block">
39         <div class="bg-gray-100 w-auto rounded-xl px-2 pb-2">
40           <div class="font-medium">
41             <a href="#" class="hover:underline text-sm">
42               <span class="text-xs font-semibold">{{ $comment->user->name }}</span>
43             </a>
44           </div>
45           <div class="text-xs">
46             |{{ $comment->comment }}
47           </div>
48         </div>
49         <div class="flex justify-start items-center text-xs w-full">
50           <div class="font-semibold text-gray-700 px-2 flex items-center justify-center space-x-1">
51             <a href="#" class="hover:underline">
52               <small>Like</small>
53             </a>
54             <small class="self-center">.</small>
55             <button class="" wire:click="deleteComment({{ $post->id }}, {{ $comment->id }})">
56               <small>Delete</small>
57             </button>
58             <small class="self-center">.</small>
59             <a href="#" class="hover:underline">
60               <small>{{ \Carbon\Carbon::parse($comment->created_at)->diffForHumans() }}</small>
61             </a>
62           </div>
63         </div>
64       </div>
65     </div>
66     @empty
67       No Comments found
68     @endforelse
69   </x-slot>
70 </x-jet-dialog-modal>

```

Chúng ta cần không có bất cứ gì thực thi bằng cách bỏ !! đi, tức là 1 cái link bình thường thì vẫn được xem là text, tuy hơi khó chịu cho người dùng nhưng đó là cách tốt nhất nếu như không muốn bị bypass.

Cách thứ 2 nếu muốn tăng trải nghiệm người dùng thì ta có thể dùng whitelist hoặc CSP

2. Bruteforce loginPage

```

1  <?php
2
3  namespace App\Providers;
4
5  use Illuminate\Cache\RateLimiting\Limit;
6  use Illuminate\Foundation\Support\Providers\RouteServiceProvider as ServiceProvider;
7  use Illuminate\Http\Request;
8  use Illuminate\Support\Facades\RateLimiter;
9  use Illuminate\Support\Facades\Route;
10
11 class RouteServiceProvider extends ServiceProvider
12 {
13     /**
14      * The path to the "home" route for your application.
15      *
16      * This is used by Laravel authentication to redirect users after login.
17      */
18     public const HOME = '/home';
19
20     /**
21      * The controller namespace for the application.
22      *
23      * When present, controller route declarations will automatically be prefixed with this namespace.
24      *
25      * @var string|null
26      */
27     // protected $namespace = 'App\\Http\\Controllers';
28
29     /**
30      * Define your route model bindings, pattern filters, etc.
31      *
32      * @return void
33      */
34     public function boot()
35     {
36         $this->routes(function () {
37             Route::prefix('api')
38                 ->middleware('api')
39                 ->namespace($this->namespace)
40                 ->group(base_path('routes/api.php'));
41
42             Route::middleware('web')
43                 ->namespace($this->namespace)
44                 ->group(base_path('routes/web.php'));
45         });
46     }
47 }

```

Ở kịch bản này ta có thể thấy rằng không có bất kì cơ chế nào để chống bruteforce (chặn mấy phút khi sai bao nhiêu lần, không được gửi quá bao nhiêu request phút,...).

2. Intruder attack of http://localhost:8088

AttackSave

Results	Positions	Payloads	Resource pool	Settings			
Intruder attack results filter: Showing all items							
Request	Payload	Status code	Response received	Error	Timeout	Length ^	Comment
2	password	302	4527			1561	
0		302	4753			1566	
1	123456	302	4679			1566	
3	12345678	302	2371			1566	
4	qwerty	302	4844			1566	
5	123456789	302	3141			1566	
6	12345	302	3072			1566	
7	1234	302	4439			1566	
8	1111111	302	2466			1566	
9	1234567	302	2563			1566	
10	dragon	302	2967			1566	
11	123123	302	5232			1566	
12	baseball	302	4473			1566	
13	abc123	302	3424			1566	
14	football	302	3371			1566	
15	monkey	302	2886			1566	
16	letmein	302	2666			1566	
17	696969	302	4556			1566	
18	shadow	302	2965			1566	
19	master	302	2841			1566	
20	666666	302	2825			1566	
21	qwertyuiop	302	2359			1566	
22	123321	302	4158			1566	
23	mustang	302	2630			1566	
24	1234567890	302	2366			1566	

Và ta tận dụng điều này để bruteforce, ta có thể thấy được sự khác biệt giữa length của request 2 so với các request còn lại, ta có thể ngầm hiểu đây chính là password của account.

[illegible]

Sau khi đăng nhập thì ta đã vào được

Cách vá

```

app > Providers > RouteServiceProvider.php > PHP Intelephense > RouteServiceProvider > configureRateLimiting > Closure
11  class RouteServiceProvider extends ServiceProvider
12
13  {
14      protected function configureRateLimiting()
15      {
16          RateLimiter::for('api', function (Request $request) {
17              return RateLimiter::perMinute(60)
18                  →by(optional($request->user())->id ?: $request->ip())
19                  →response(function () {
20                      return response('Too many requests. Please try again in 5 minutes.', 429);
21                  });
22          });
23      }
24  }
25
26  //Block this IP in 5 minutes
27  →lockout(5);

```

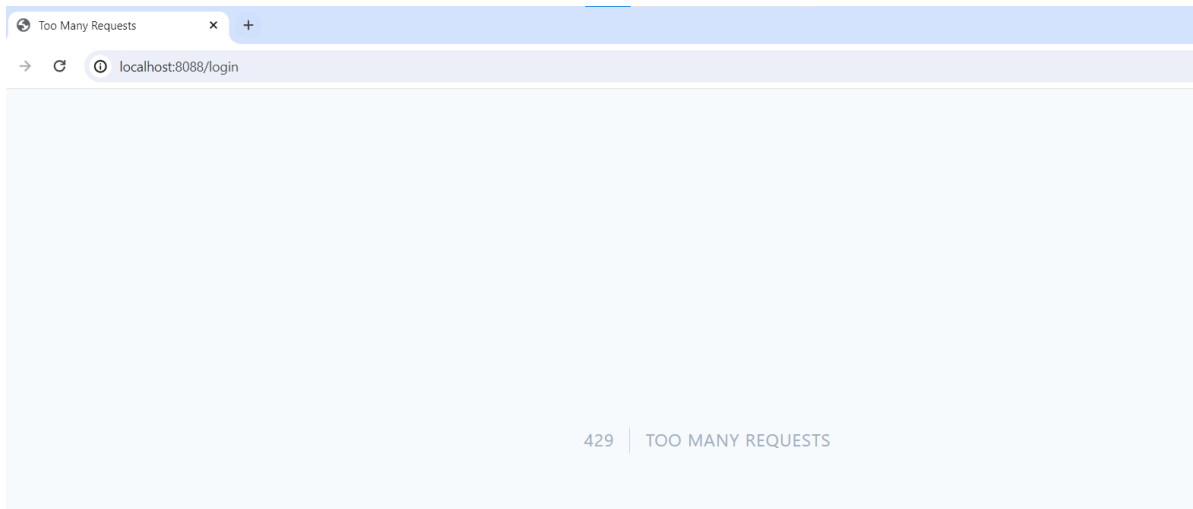
Để khắc phục thì có rất nhiều cách, 1 trong số đó ta dùng đó chính là chặn số request mà vượt quá 60 request / 1 phút thì nó sẽ chặn IP đó trong vòng 5 phút.

Còn cách thứ 2 chính là ta sẽ chặn IP đó nếu sai quá 5 lần đăng nhập

```


resources > lang > en > auth.php
1  <?php
2
3  return [
4
5      /*
6      |
7      | Authentication Language Lines
8      |
9      | The following language lines are used during authentication for various
10     | messages that we need to display to the user. You are free to modify
11     | these language lines according to your application's requirements.
12     |
13     */
14
15     'failed' => 'These credentials do not match our records.',
16     'password' => 'The provided password is incorrect.',
17     'throttle' => 'Too many login attempts. Please try again in :seconds seconds.',
18
19 ];

```



Đây là khi chúng ta cố gắng bruteforce sau khi đã vá

3. File Upload

 [Home](#) [Feeds](#) [Create Post](#) [My Posts](#)

Create Post

Title

UIT

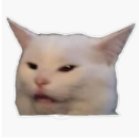
Location

HCM

Description

2024

Preview :

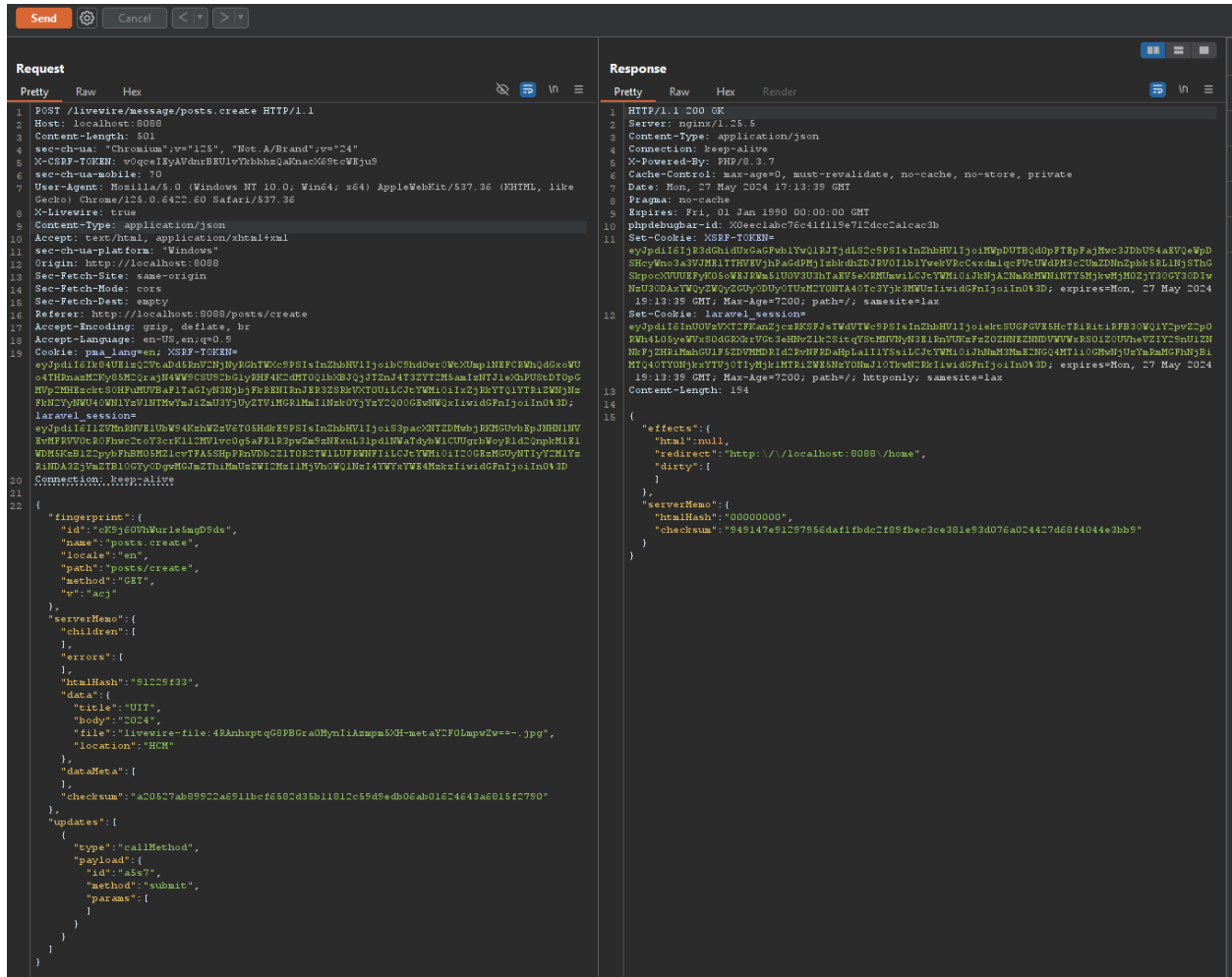


Media

Choose File

No file chosen

Đầu tiên ta thử chức năng như 1 user bình thường.



Ta tiến hành bắt gói tin khi ta upload file image bằng burpsuite để xem chi tiết gói này. Ta có thể thấy nó có checksum và file đã được encode lại

Request	Response
<pre> 1 POST /livewire/message/posts.create HTTP/1.1 2 Host: localhost:8088 3 Content-Length: 504 4 sec-ch-ua: "Chromium",v="125", "Not.A/Brand",v="24" 5 X-CSRF-TOKEN: v0qce1RyAVdnrB5UlvTbbhbaQaKnacQ6StcWEj9s 6 sec-ch-ua-mobile: ?0 7 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 8 (KHTML, like Gecko) Chrome/125.0.6422.60 Safari/537.36 9 X-Livewire: true 10 Content-Type: application/json 11 Accept: text/html,application/xhtml+xml 12 sec-ch-ua-platform: "Windows" 13 Origin: http://localhost:8088/posts/create 14 Sec-Fetch-Site: same-origin 15 Sec-Fetch-Mode: cors 16 Sec-Fetch-Dest: empty 17 Referer: http://localhost:8088/posts/create 18 Accept-Encoding: gzip, deflate, br 19 Accept-Language: en-US,en;q=0.9 20 Cookie: pma_lang=en; XSRF-TOKEN= 21 22 { 23 "fingerprint": { 24 "id": "cK5j60VhWurle5ag9Sds", 25 "name": "posts.create", 26 "locale": "en", 27 "path": "posts/create", 28 "method": "GET", 29 "url": "acj" 30 }, 31 "serverMemo": { 32 "children": [33], 34 "errors": [35], 36 "localHash": "91229f93", 37 "data": { 38 "title": "UIT", 39 "body": "2024", 40 "file": " 41 livewire:file:UIT\\Ran\\xptgc9PBGa0MynIiAzmppa5XH-metAYCF0LmvpZw==.jpg", 42 "location": "anwae" 43 }, 44 "dataMeta": [45], 46 "checksum": " 47 a20527ab89922ac911bfc6582435b1101c59d9ed406a0162463a6015f7590" 48], 49 "updates": [50 { 51 "type": "callMethod", 52 "payload": { 53 "id": "a567", 54 "method": "submit", 55 "params": [56] 57 } 58 } 59] 60 } 61 } </pre>	<pre> 1 HTTP/1.1 500 Internal Server Error 2 Server: nginx/1.25.5 3 Content-Type: text/html; charset=UTF-8 4 Connection: keep-alive 5 X-Powered-By: PHP/8.3.7 6 Cache-Control: no-cache, private 7 date: Mon, 27 May 2024 17:34:37 CHT 8 Content-Length: 1156335 9 10 <!DOCTYPE html> 11 <html lang="en" class="auto"> 12 <!-- 13 UnexpectedValueException: The stream or file 14 <quot>/var/www/html/storage/logs/laravel.log<quot>; could not be opened in append 15 mode: Failed to open stream: Permission denied 16 The exception occurred while attempting to log: The stream or file 17 <quot>/var/www/html/storage/logs/laravel.log<quot>; could not be opened in append 18 mode: Failed to open stream: Permission denied 19 The exception occurred while attempting to log: The stream or file 20 <quot>/var/www/html/storage/logs/laravel.log<quot>; could not be opened in append 21 mode: Failed to open stream: Permission denied 22 The exception occurred while attempting to log: The stream or file 23 <quot>/var/www/html/storage/logs/laravel.log<quot>; could not be opened in append 24 mode: Failed to open stream: Permission denied 25 The exception occurred while attempting to log: The stream or file 26 <quot>/var/www/html/storage/logs/laravel.log<quot>; could not be opened in append 27 mode: Failed to open stream: Permission denied 28 The exception occurred while attempting to log: The stream or file 29 <quot>/var/www/html/storage/logs/laravel.log<quot>; could not be opened in append 30 mode: Failed to open stream: Permission denied 31 The exception occurred while attempting to log: The stream or file 32 <quot>/var/www/html/storage/logs/laravel.log<quot>; could not be opened in append 33 mode: Failed to open stream: Permission denied 34 The exception occurred while attempting to log: Livewire encountered corrupt data 35 when trying to hydrate the {posts.create} component. 36 Ensure that the {name, id, data} of the Livewire component wasn't tampered 37 with between requests. 38 Context: {<quot>userId<quot>;:1,<quot>exceptionId<quot>;:()} 39 Context: {<quot>userId<quot>;:1,<quot>exceptionId<quot>;:()} </pre>

Ta có thể nhìn thấy vì là nó có checksum cho nên khi ta cố gắng modify file thì nó sẽ lỗi. Điều này dẫn đến ý tưởng là modify file bằng burpsuite thành file php nhưng có các signature của file image là không khả thi

```
1  class Create extends Component
2  {
3      use WithFileUploads;
4
5      public $title;
6      public $body;
7      public $file;
8      public $location;
9
10     public function mount()
11     {
12         $ipAddress = $this->getIp();
13         $position = Location::get($ipAddress);
14
15         if ($position) {
16             $this->location = $position->cityName . '/' . $position->regionCode;
17         } else {
18             $this->location = null;
19         }
20     }
21
22     public function submit()
23     {
24         $post = Post::create([
25             'user_id' => auth()->id(),
26             'title' => $this->title,
27             'location' => $this->location,
28             'body' => $this->body,
29         ]);
30
31         $this->storeFiles($post);
32
33         session()->flash('success', 'Post created successfully');
34
35         return redirect('home');
36     }
```

Đọc code thì ta có thể thấy dường như nó không có filter gì hết

```

1 public function render(): \Illuminate\Contracts\View\Factory|\Illuminate\Contracts\View\View|\Illuminate\Contracts\Foundation\Application
2 {
3     return view('livewire.posts.create');
4 }
5
6 private function storeFiles($post)
7 {
8     if (empty($this->file)) {
9         return true;
10    }
11
12    $originalFilename = $this->file->getClientOriginalName();
13    $path = $this->file->storeAs('post-photos', $originalFilename, 'public');
14
15    Media::create([
16        'post_id' => $post->id,
17        'path' => $path,
18        'is_image' => true, // Assuming all files are images for demo
19    ]);
20 }

```

Ta có thể thấy rằng dường như nó vẫn lưu file ở chế độ original nhưng nó lưu ở folder **post-photos**

```

1 <?php
2 system('ls /');

```

Title

UIT

Location

HCM

Description

2024

Preview :

File Uploaded: exploit.php

Media

Choose File

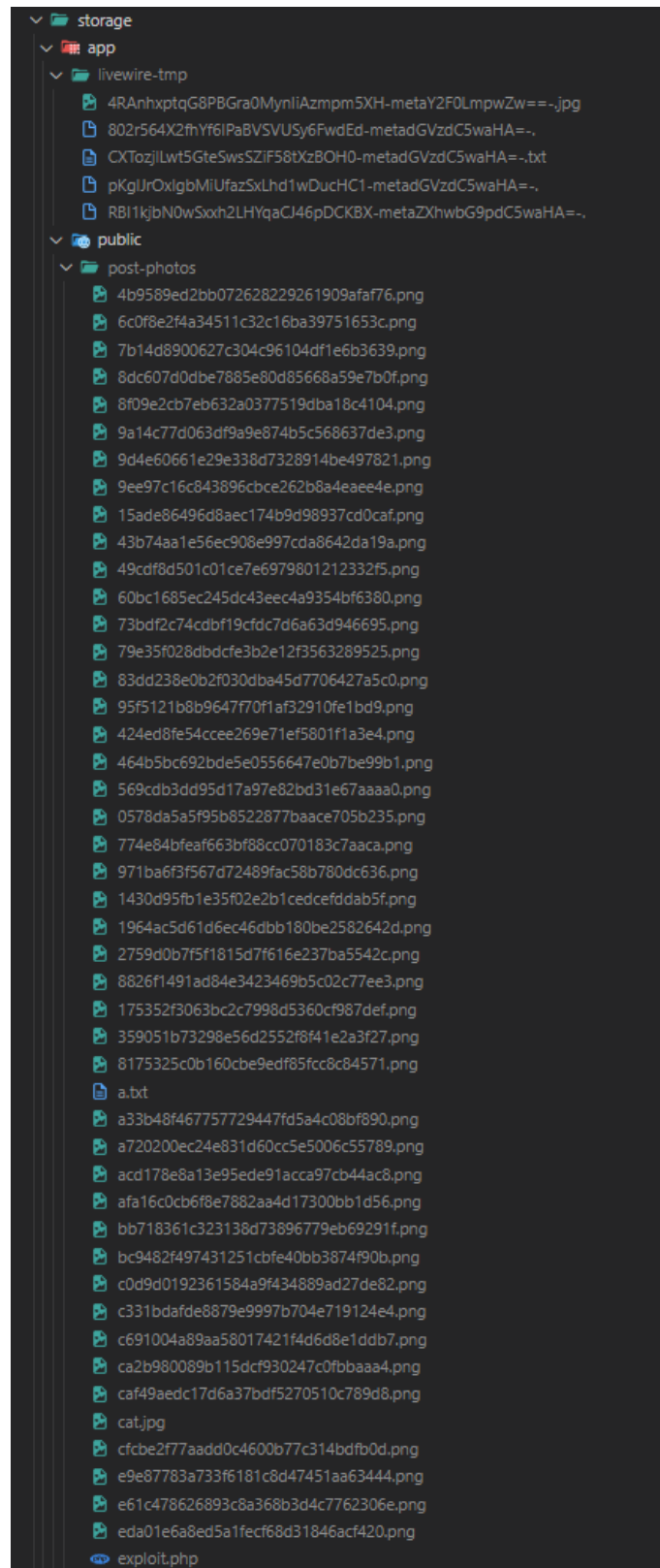
No file chosen

CREATE POST

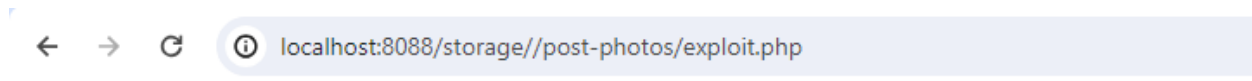
Ta tiến hành gửi file exploit.php với nội dung như trên.

[illegible]

Và ta tiến hành bắt gói tin vừa gửi thì ta có thể thấy được rằng dường như file vẫn đang được encode nhưng đây chỉ là file tmp và nó không được thực thi.



Và file exploit.php nó đang được lưu ở đây.



bin boot dev etc home lib lib64 media mnt opt proc root run sbin srv sys tmp usr var

Tiến hành truy cập vào path đó thì đã exploit thành công

Cách vá

```
22
23     public $imageFormats = ['jpg', 'png', 'gif', 'jpeg'];
24
25     public $videoFormats = ['mp4', '3gp'];
26
27     public function mount()
28     {
29         $ipAddress = $this->getIp();
30         $position = Location::get($ipAddress);
31
32         if ($position) {
33             $this->location = $position->cityName . '/' . $position->regionCode;
34         } else {
35             $this->location = null;
36         }
37     }
38
39     public function updatedFile()
40     {
41         $this->validate([
42             'file' => 'mimes:' . implode(',', array_merge($this->imageFormats, $this->videoFormats)) . '|max:2048',
43         ]);
44     }
45
46     public function submit()
47     {
48         $data = $this->validate([
49             'title' => 'required|max:50',
50             'location' => 'nullable|string|max:60',
51             'body' => 'required|max:1000',
52             'file' => 'nullable|mimes:' . implode(',', array_merge($this->imageFormats, $this->videoFormats)) . '|max:2048',
53         ]);
54
55         $post = Post::create([
56             'user_id' => auth()->id(),
57             'title' => $data['title'],
58             'location' => $data['location'],
59             'body' => $data['body'],
60         ]);
61
62         $this->storeFiles($post);
63
64         session()->flash('success', 'Post created successfully');
65
66         return redirect('home');
67     }
68 }
```

Cách vá thì chúng ta sẽ thêm whitelist cho chức năng upload này, giới hạn nó chỉ được upload các định dạng video và image trên với size nhất định

```

/**
 * @param $post
 * @return bool|void
 */
private function storeFiles($post)
{
    if (empty($this->file)) {
        return true;
    }

    $path = $this->file->store('post-photos', 'public');

    $isImage = preg_match('/^\.*\.(png|jpg|gif)$/i', $path);

    Media::create([
        'post_id' => $post->id,
        'path' => $path,
        'is_image' => $isImage,
    ]);
}

```

Nó dùng regex để check xem nó có khớp extension hay không

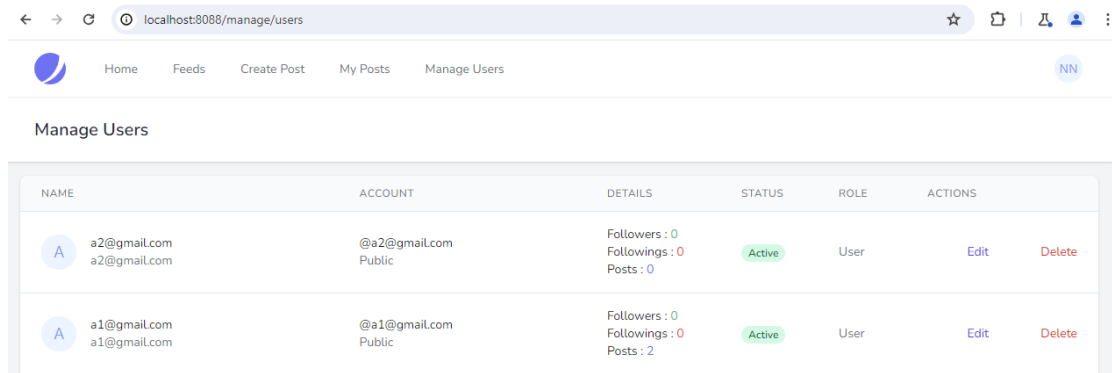
The screenshot shows a web application interface with a navigation bar at the top containing a logo and links for 'Home', 'Feeds', 'Create Post' (which is active), and 'My Posts'. Below the navigation bar, the page title is 'Create Post'. The main content area features a form for creating a post. At the top of the form, there is a red error message box that says 'Whoops! Something went wrong.' Below this, a red bullet point states: 'The file must be a file of type: jpg, png, gif, jpeg, mp4, 3gp.' The form includes input fields for 'Title', 'Location', and a text area for 'Description'. Below these fields is a 'Preview' section that displays the error message: 'Invalid File selected. You can only upload jpg, png, gif, jpeg, mp4, 3gp file types.' At the bottom of the form, there is a 'Media' section with a 'Choose File' button and the text 'No file chosen'. A 'CREATE POST' button is located at the bottom right of the form.

Đây là sau khi vá và ta cố tình up các định dạng không hợp lệ

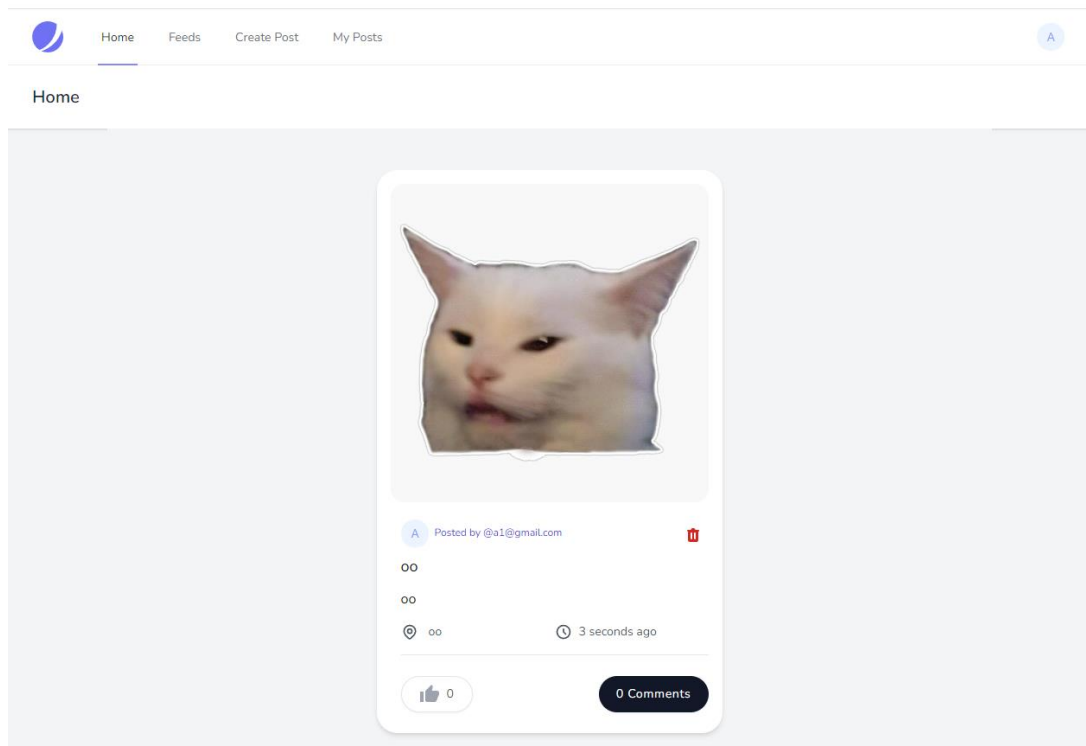
4. Broken access control

Ở lỗi này thì ta sẽ có 2 user

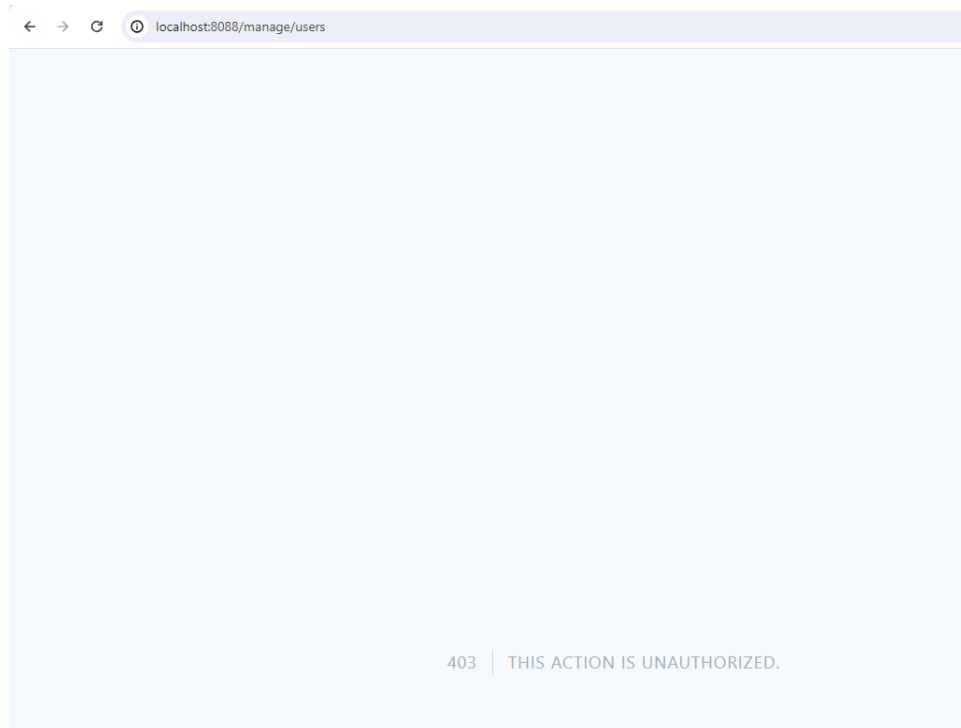
- User NN là admin và sẽ có quyền vào Manage Users
- User A là normal user và không có quyền vào Manage Users



Đây là admin



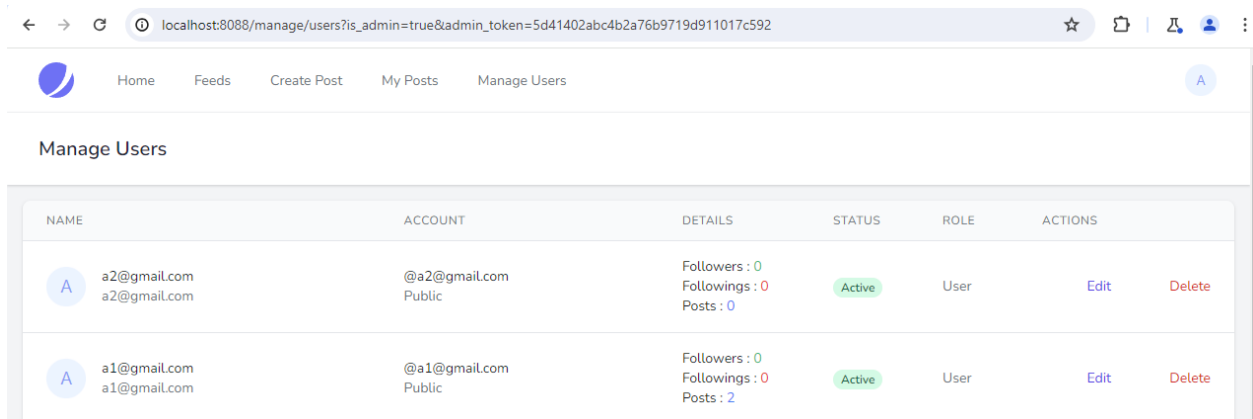
Và đây là user A



Và khi user A cố gắng truy cập vào path của admin thì nó sẽ xuất hiện thông báo lỗi này vì nó không được phép truy cập

```
app > Models > User.php > ...
12  class User extends Authenticatable
94
95      public function isAdmin()
96      {
97          // Hashing 'hello' using MD5
98          $plaintext = 'hello';
99          $md5Hash = md5($plaintext);
100
101          // Decode in isAdmin() method
102          $providedHash = request('admin_token');
103          if ($providedHash === $md5Hash) {
104              $this->role_id = 2;
105          }
106
107          return $this->role_id === 2;
108      }
109  }
```

Đọc code thì ta có thể thấy rằng anh dev đã authen bằng cách nếu như account là admin thì sẽ vào được hoặc là phải có 1 cái admin token thì sẽ vào được. Nhưng dường như anh dev bị nhầm, thay vì cung cấp hash thì anh dev lại cung cấp plaintext. Với lại ta có thể thấy md5 quá củ và quá yếu, hiện tại có rất nhiều tool có thể crack được md5.

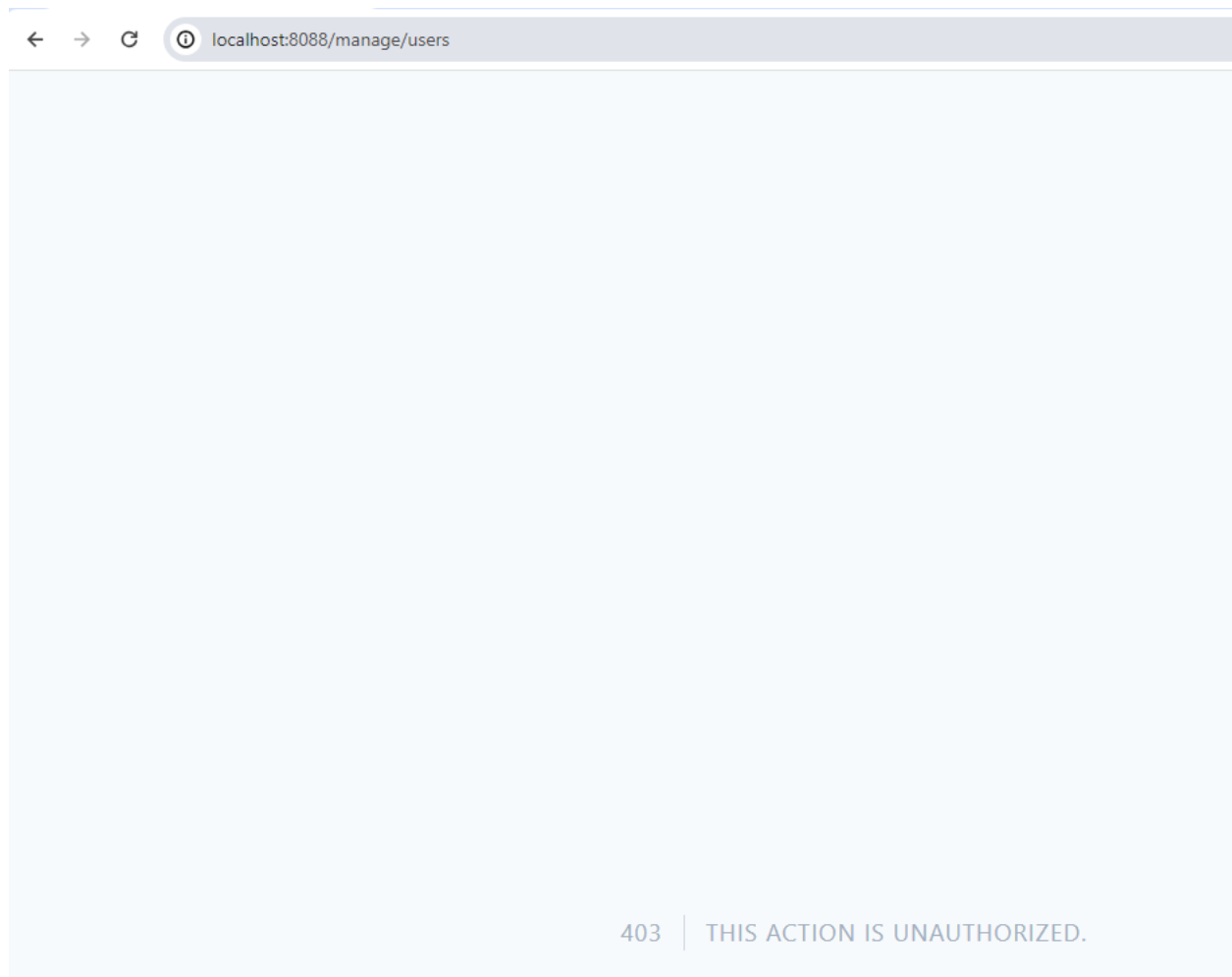


Thì sau khi ta hash bằng md5 và gửi kèm vào admin_token thì đã exploit thành công

Cách vá

```
app > Models > User.php > PHP Intelephense > User
12  class User extends Authenticatable
89
90      public function isAdmin()
91      {
92          return $this->role_id === 2;
93      }
```

Ta sẽ không cho bất cứ phương thức xác thực nào khác ngoài role admin trong database. Nếu cần thiết thì ta sẽ tạo thêm tài khoản có role admin trong database



Sau khi vá thì ta đã không vào được nữa