LogicPanel সিকিউরিটি ও RESTful API অডিট রিপোর্ট

Repository: LogicDock/LogicPanel
ধরণ: Node.js/Python হোস্টিং কন্ট্রোল প্যানেল (PHP-based API)
অডিট তারিখ: ১ ফেব্রুয়ারি, ২০২৬

📋 সংক্ষিপ্তসার

LogicPanel একটি স্ট্রাকচার্ড PHP-ভিত্তিক কন্ট্রোল প্যানেল যার API ডিজাইন মূলত RESTful নীতির সাথে সঙ্গতিপূর্ণ। সিকিউরিটি দিক থেকে মৌলিক সুরক্ষা ব্যবস্থা (JWT, পাথ ভ্যালিডেশন, রেট লিমিট) বিদ্যমান, তবে কিছু দুর্বলতা (ডিফল্ট সিক্রেট, সীমিত ইনপুট ভ্যালিডেশন) চিহ্নিত করা হয়েছে। API টি প্রোডাকশনে ব্যবহারযোগ্য হলেও ভার্সনিং, পেজিনেশন ও হাইপারমিডিয়া সাপোর্ট যোগ করলে এটি আরও স্কেলেবল ও ডেভেলপার-বান্ধব হবে।

🔍 সিকিউরিটি অডিট বিশ্লেষণ

✅ শক্তিশালী দিকগুলো

১. অথেন্টিকেশন ব্যবস্থা

- JWT + API-Key **সমর্থন**: দুই ধরনের অথেন্টিকেশন পদ্ধতি সমর্থন করে
- **রিফ্রেশ টোকেন**: টোকেন রিফ্রেশ মেকানিজম বিদ্যমান
- **একটিভ অ্যাকাউন্ট চেক**: ব্যবহারকারীর অ্যাকাউন্ট সক্রিয় কিনা তা যাচাই করে

২. অথরাইজেশন কন্ট্রোল

- **মিডলওয়্যার ভিত্তিক**: `AuthMiddleware` ও `MasterAuthMiddleware` দ্বারা রুট প্রোটেকশন
- **রোল-ভিত্তিক অ্যাক্সেস**: বিভিন্ন লেভেলের অ্যাক্সেস কন্ট্রোল

৩. ফাইল সিকিউরিটি

- **পাথ ট্রাভারসাল প্রোটেকশন**: `FileController` -এ `sanitizePath()` ও `isPathSafe()` মেথড
- **ফাইল আপলোড রেস্ট্রিকশন**: `allowedExtensions` কনফিগারেবল, ডিফল্ট ১০MB সীমা

৪. ডাটাবেজ সিকিউরিটি

- ORM **ব্যবহার**: এলোকুয়েন্ট ORM দ্বারা SQL ইনজেকশন ঝুঁকি হ্রাস
- **প্রিপেয়ার্ড স্টেটমেন্ট**: ইমপ্লিসিটভাবে ব্যবহার করা হয়

৫. নেটওয়ার্ক সিকিউরিটি

- CORS **কনফিগারেশন**: `CorsMiddleware` দ্বারা সঠিকভাবে কনফিগারড
- **রেট লিমিটিং**: `RateLimitMiddleware` দ্বারা API আবিউজ প্রতিরোধ

⚠️ উন্নয়নযোগ্য ক্ষেত্র

১. ইনপুট ভ্যালিডেশন

- **অসম্পূর্ণ ভ্যালিডেশন**: কিছু এন্ডপয়েন্টে ব্যাপক ইনপুট ভ্যালিডেশন নেই
- **সীমিত স্যানিটাইজেশন**: কন্ট্রোলারে আংশিক ভ্যালিডেশন (সার্ভিস নামের regex)

২. সিক্রেট ম্যানেজমেন্ট

- **ডিফল্ট সিক্রেট**: `.env.example` -এ ডিফল্ট JWT সিক্রেট ও ডিবি পাসওয়ার্ড
- **স্ট্যাটিক এনক্রিপশন কী**: র‍্যান্ডম জেনারেশন মেকানিজমের অভাব

৩. ক্রস-সাইট স্ক্রিপ্টিং (XSS)

- **আউটপুট এস্কেপিং**: ফ্রন্ট-এন্ডে আউটপুট এস্কেপিং-এর পর্যাপ্ত ব্যবস্থা নেই
- Content Security Policy: CSP হেডার কনফিগারেশন নেই

৪. ডিফল্ট ক্রেডেনশিয়ালস

- **ডিফল্ট ডিবি পাসওয়ার্ড**: `settings.php` -এ ডিফল্ট পাসওয়ার্ড `'578496'`
- **ইনস্টলেশন সিকিউরিটি**: ইনস্টলেশনকালে অটোমেটিক সিক্রেট জেনারেশন নেই

৫. লগিং ও মনিটরিং

- **অসম্পূর্ণ লগিং**: কিছু কন্ট্রোলারে ফাইল-লগিং আছে
- **কেন্দ্রীয় লগিং সিস্টেম**: স্ট্রাকচার্ড লগিং সিস্টেমের অভাব

🔗 RESTful API মূল্যায়ন

✅ RESTful নীতি অনুসরণ

নীতি	অবস্থা	বিবরণ
সেমান্টিক URI	✅ ভালো	রিসোর্স-ভিত্তিক URI (`/services` , `/databases` , `/packages`)
HTTP **মেথড**	✅ বেশিরভাগ ভালো	GET, POST, PUT, DELETE, PATCH সঠিকভাবে ব্যবহার
স্টেটলেসনেস	✅ আছে	JWT/API-Key প্রতি রিকোয়েস্টে
স্ট্যান্ডার্ড স্ট্যাটাস কোড	✅ ভালো	200, 201, 400, 401, 403, 404, 500 সঠিকভাবে ব্যবহৃত
JSON **রেসপন্স**	✅ আছে	সব API `Content-Type: application/json` দেয়

❌ অনুপস্থিত RESTful বৈশিষ্ট্য

বৈশিষ্ট্য	অবস্থা	প্রভাব
API **ভার্সনিং**	❌ নেই	ব্রেকিং চেঞ্জের ঝুঁকি, ক্লায়েন্ট ইন্টিগ্রেশন জটিল
হাইপারমিডিয়া (HATEOAS)	❌ নেই	ডিসকভারিবিলিটি কম, ক্লায়েন্ট স্বাধীনতা হ্রাস
পেজিনেশন	❌ নেই	বড় ডাটাসেটে পারফরম্যান্স ইস্যু, মেমোরি সমস্যা
ফিল্টারিং/সর্টিং	❌ নেই	ডাটা রিট্রিভাল সীমিত, ইউজার এক্সপেরিয়েন্স খারাপ
স্ট্যান্ডার্ড ডকুমেন্টেশন	⚠️ আংশিক	Swagger/OpenAPI ডকুমেন্টেশন নেই

🛡️ সিকিউরিটি সুপারিশসমূহ

উচ্চ প্রাধান্য

১. সিক্রেট ম্যানেজমেন্ট উন্নতি

```php
// ইনস্টলেশন স্ক্রিপ্টে র‍্যান্ডম সিক্রেট জেনারেশন যোগ করুন
$randomSecret = bin2hex(random_bytes(32));
file_put_contents('.env', 'JWT_SECRET=' . $randomSecret . PHP_EOL);
```

২. ডিফল্ট ক্রেডেনশিয়ালস অপসারণ

- `.env.example` থেকে সব ডিফল্ট পাসওয়ার্ড সরান
- ইনস্টলেশনকালে ইউজারকে কাস্টম সিক্রেট সেট করতে বাধ্য করুন

৩. ব্যাপক ইনপুট ভ্যালিডেশন

```php
// সকল কন্ট্রোলারে স্ট্রিক্ট ভ্যালিডেশন ইমপ্লিমেন্ট করুন
public function validateRequest(array $data, array $rules)
{
    $validator = Validator::make($data, $rules);

    if ($validator->fails()) {
        throw new ValidationException($validator);
    }

    return $validator->validated();
}
```

মধ্যম প্রাধান্য

৪. HTTP সিকিউরিটি হেডার যোগ

```php
// SecurityMiddleware তৈরি করুন
class SecurityMiddleware
{
    public function handle($request, Closure $next)
    {
        $response = $next($request);

        $response->headers->set('X-Content-Type-Options', 'nosniff');
        $response->headers->set('X-Frame-Options', 'DENY');
        $response->headers->set('X-XSS-Protection', '1; mode=block');

        return $response;
    }
}
```

৫. লগিং সিস্টেম শক্তিশালীকরণ

- কেন্দ্রীয় লগিং মেকানিজম তৈরি করুন
- সমস্ত অথেন্টিকেশন ও ক্রিটিক্যাল অপারেশন লগ করুন
- লগ ফাইল পারমিশন সীমিত রাখুন (0640)

৬. ফাইল আপলোড সিকিউরিটি

```php
```

```
// আরও রেস্ট্রিক্টিভ এক্সটেনশন লিস্ট
$dangerousExtensions = ['php', 'phtml', 'phar', 'sh', 'bash', 'py', 'pl'];
$allowedExtensions = array_diff($allowedExtensions, $dangerousExtensions);
```

নিম্ন প্রাধান্য

৭. রেগুলার সিকিউরিটি আপডেট

- ডিপেন্ডেন্সি আপডেট রাখুন
- সিকিউরিটি স্ক্যানিং টুল ইন্টিগ্রেট করুন
- কোড রিভিউ প্রক্রিয়া শক্তিশালী করুন

RESTful API উন্নতির সুপারিশ

১. API ভার্সনিং ইমপ্লিমেন্টেশন

```php
// routes/api.php
Route::prefix('v1')->group(function () {
    Route::apiResource('services', ServiceController::class);
    Route::apiResource('databases', DatabaseController::class);
});

Route::prefix('v2')->group(function () {
    // নতুন ভার্সনের API
});
```

২. পেজিনেশন ও ফিল্টারিং যোগ

```php
// BaseController-এ পেজিনেশন মেথড
protected function paginate($query, $request)
{
    $page = $request->get('page', 1);
    $limit = min($request->get('limit', 20), 100);
    $sortBy = $request->get('sort_by', 'id');
    $sortOrder = $request->get('sort_order', 'asc');

    return $query->orderBy($sortBy, $sortOrder)
                 ->paginate($limit, ['*'], 'page', $page);
}
```

৩. হাইপারমিডিয়া (HATEOAS) সাপোর্ট

```php
// রেসপন্সে লিঙ্ক যোগ
protected function addLinks($data, $request, $resourceName)
{
    if (isset($data['data'])) {
        $data['_links'] = [
```

```
            'self' => $request->fullUrl(),
            'next' => $this->getNextPageUrl($data, $request),
            'previous' => $this->getPreviousPageUrl($data, $request)
        ];
    }

    return $data;
}
```

৪. Swagger/OpenAPI ডকুমেন্টেশন

```yaml
# swagger.yaml
openapi: 3.0.0
info:
  title: LogicPanel API
  version: 1.0.0
  description: Node.js/Python Hosting Control Panel API

paths:
  /api/v1/services:
    get:
      summary: Get all services
      responses:
        '200':
          description: Successful response
          content:
            application/json:
              schema:
                type: array
                items:
                  $ref: '#/components/schemas/Service'
```

৫. ইউনিফর্ম এরর হ্যান্ডলিং

```php
// App\Exceptions\Handler.php
public function render($request, Throwable $exception)
{
    if ($request->expectsJson()) {
        return response()->json([
            'error' => [
                'code' => $this->getErrorCode($exception),
                'message' => $exception->getMessage(),
                'details' => config('app.debug') ? $exception->getTrace() : null
            ]
        ], $this->getStatusCode($exception));
    }

    return parent::render($request, $exception);
}
```

৬. API কনটেন্ট নেগোশিয়েশন

```php
// Middleware/ContentNegotiation.php
public function handle($request, Closure $next)
{
    $accept = $request->header('Accept');

    if (strpos($accept, 'application/xml') !== false) {
        $request->headers->set('Accept', 'application/json');
        // XML সাপোর্ট ভবিষ্যতে যোগ করতে পারেন
    }

    return $next($request);
}
```

🚀 অ্যাকশন প্ল্যান

তাত্ক্ষণিক অ্যাকশন (১-২ দিন)

1. ডিফল্ট ডিবি পাসওয়ার্ড পরিবর্তন করুন
2. `.env.example` থেকে সিক্রেট সরান
3. কোম্পোজার ডিপেন্ডেন্সি আপডেট করুন

স্বল্পমেয়াদী অ্যাকশন (১ সপ্তাহ)

1. API ভার্সনিং (`/api/v1/`) ইমপ্লিমেন্ট করুন
2. পেজিনেশন সাপোর্ট যোগ করুন
3. ব্যাপক ইনপুট ভ্যালিডেশন যোগ করুন

মধ্যমেয়াদী অ্যাকশন (২-৪ সপ্তাহ)

1. Swagger/OpenAPI ডকুমেন্টেশন তৈরি করুন
2. HTTP সিকিউরিটি হেডার যোগ করুন
3. কেন্দ্রীয় লগিং সিস্টেম ইমপ্লিমেন্ট করুন

দীর্ঘমেয়াদী অ্যাকশন (১-২ মাস)

1. হাইপারমিডিয়া (HATEOAS) সাপোর্ট যোগ করুন
2. API ক্যাচিং মেকানিজম ইমপ্লিমেন্ট করুন
3. অটোমেটেড সিকিউরিটি টেস্টিং সেটআপ করুন

📊 ঝুঁকি ম্যাট্রিক্স

ঝুঁকি	গুরুত্ব	সম্ভাবনা	প্রভাব	সামগ্রিক রেটিং
ডিফল্ট ক্রেডেনশিয়ালস	উচ্চ	উচ্চ	উচ্চ	🔴 গুরুতর
ইনপুট ভ্যালিডেশন অভাব	উচ্চ	মধ্যম	উচ্চ	🔴 গুরুতর
API ভার্সনিং নেই	মধ্যম	উচ্চ	মধ্যম	🟠 মধ্যম
পেজিনেশন অভাব	মধ্যম	উচ্চ	নিম্ন	🟠 মধ্যম
HATEOAS অভাব	নিম্ন	উচ্চ	নিম্ন	🟢 নিম্ন
ডকুমেন্টেশন সীমিত	নিম্ন	উচ্চ	নিম্ন	🟢 নিম্ন

✅ উপসংহার

LogicPanel এর API ইনফ্রাস্ট্রাকচার একটি শক্তিশালী ভিত্তি প্রদান করে যাতে মৌলিক সিকিউরিটি ও RESTful নীতি বিদ্যমান। তবে প্রোডাকশন রেডিনেসের জন্য কিছু গুরুত্বপূর্ণ উন্নয়ন প্রয়োজন, বিশেষত সিক্রেট ম্যানেজমেন্ট, ইনপুট ভ্যালিডেশন, এবং API ভার্সনিং। প্রস্তাবিত উন্নয়নগুলো ইমপ্লিমেন্ট করলে প্যানেলটি আরও সুরক্ষিত, স্কেলেবল ও ডেভেলপার-বান্ধব হবে।

পরবর্তী পদক্ষেপ:

1. ডিফল্ট সিক্রেট ও ক্রেডেনশিয়ালস সমস্যা সমাধান করুন
2. API ভার্সনিং ও পেজিনেশন যোগ করুন
3. একটি OpenAPI ডকুমেন্টেশন তৈরি করুন

এই রিপোর্টটি কোড রিভিউ ভিত্তিক এবং অটোমেটেড সিকিউরিটি স্ক্যানিং টুলের বিকল্প নয়। প্রোডাকশন ডেপ্লয়মেন্টের পূর্বে সম্পূর্ণ সিকিউরিটি অডিট ও পেনেট্রেশন টেস্টিং করার সুপারিশ করা হচ্ছে।