python123第四周_百度杯十月第四周WriteUp

weixin_39735166 于 2020-11-30 23:20:34 发布 40 收藏
文章标签： python123第四周
本帖最后由 一口盐汽水 于 2016-11-2 11:21 编辑

0x00 misc

1、泄露的数据md5解密即可

2、考眼力

凯撒密码

3、flag格式

直接比较题干中的flag即可

0x01 Web

1、try

查看源代码有个注释 reset.php

http://e24890cb606846708221a491cd15d6a2e983c17bacc24676.game.ichunqiu.com/level.php?name=guest' or '1'='1

存在注入，可以使用OPTIONS请求突破

这里直接使用sqlmap来跑

图片1.png (59.48 KB, 下载次数: 68)

2016-11-2 11:00 上传

user表有2个用户guest member，token表为空

dump数据发现guest为空 就拿member来测试

reset.php 源代码有提示

图片2.png (1.92 KB, 下载次数: 65)

2016-11-2 11:01 上传

这里去搜了下类似的格式

在php官网上找到能生成类似结果的加密函数

图片3.png (1.37 KB, 下载次数: 52)

2016-11-2 11:01 上传

顿折腾发现第二个参数为6rounds=66 能生成一样的hash

图片4.png (3.54 KB, 下载次数: 63)

2016-11-2 11:01 上传

图片5.png (854 Bytes, 下载次数: 56)

2016-11-2 11:02 上传

reset.php 账号填member 尝试重置 reset_do.php

然后通过注入查询token

发现hash会变 并且格式和提示的一样 这里通过没法爆破发现只要reset_do.php提交的token一旦错误就会清空数据表

本地通过python脚本生成一份hash表 (这里不要全生成需要花费太多时间)

图片6.png (36.73 KB, 下载次数: 60)

2016-11-2 11:03 上传

思路：

通过reset.php重置再通过注入获取token 接着把token放到hash表中查找 如果有对应的明文则是token 否则继续重置 然后重复

脚本如下：

图片7.png (40.82 KB, 下载次数: 69)

2016-11-2 11:03 上传

接着挂机去跑就行了 如果2000次没跑出来重复跑 如果生成的hash表数据足够多几率就越高

图片8.png (29.56 KB, 下载次数: 38)

2016-11-2 11:04 上传

然后拿到对应的hash去hash表中找明文

去/reset_do.php 重置新密码

图片9.png (6.94 KB, 下载次数: 57)

2016-11-2 11:04 上传

去login.php登陆返回Get_Fl3g_e165421110ba03099a1c0393373c5b43.php源代码注释提示 备份文件 Get_Fl3g_e165421110ba03099a1c0393373c5b43.txt

图片10.png (11.18 KB, 下载次数: 69)

2016-11-2 11:05 上传

这里一个循环$_GET $_POST $_COOKIE 注册变量如果key 为_SESSION会跳过但是这里$_SERVER没有进行检查key最终检查$_SESSION[admin]为yes这回打印flag利用没有检查$_SERVER的key 可以通过上面的get post cookie来注册个二维数组来跳过前面的key检查 然后到循环$_SERVER再进行一次注册给$_SESSION来赋值[PHP] 纯文本查看 复制代码Get_Fl3g_e165421110ba03099a1c0393373c5b43.php?_SERVER[_SESSION][admin]=yes

或者post过去也可以

图片11.png (28.7 KB, 下载次数: 44)

2016-11-2 11:06 上传

2、Hash

访问题目发现一个超链接

[PHP] 纯文本查看 复制代码
http://72db6205d8c9474995e81100930ecc07c22aadb2fa8e4c23.game.ichunqiu.com/index.php?key=123&hash=f9109d5f83921a551cf859f853afe7bb

hash解密是md5 kkkkkk01123 感觉后3位是key 尝试kkkkkk01admin md5后049f601185c0846faac45065a834b1c5http://72db6205d8c9474995e81100930ecc07c22aadb2fa8e4c23.game.ikey=admin&hash=049f601185c0846faac45065a834b1c5提示next step is Gu3ss_m3_h2h2.php 访问

图片12.png (12.49 KB, 下载次数: 45)

2016-11-2 11:09 上传

接受一个var参数base64解码然后正则匹配如果匹配到程序结束执行 否则就进行反序列化可以发现是一个文件读取类 但是会在执行__destruct之前会调用__wakeup来修改掉file变量这里利用一个php的bug简单来说就是当序列化字符串中表示对象属性个数的值大于真实的属性个数时会跳过__wakeup的执行 然后绕过正则的话我们可以在对象长度前加一个 + 号http://www.milw0rm.cn/Article/exploit/20160915/vulzone-26.html然后生成一下payload

图片13.png (56.27 KB, 下载次数: 71)

2016-11-2 11:09 上传

[AppleScript] 纯文本查看 复制代码
TzorNDoiRGVtbyI6NTp7czoxMDoiAERlbW8AZmlsZSI7czoxNjoiZjE1Z18xc19oZXJlLnBocCI7fQ==

图片14.png (6.87 KB, 下载次数: 74)

2016-11-2 11:10 上传

接受一个参数val 然后eval参数是进行变量赋值 这里payload是${phpinfo()} 或者${${phpinfo()}}可以执行phpinfo 改成一句话${@eval($_POST[0])} 然后post突破WAF执行代码[PHP] 纯文本查看 复制代码 http://72db6205d8c9474995e81100930ecc07c22aadb2fa8e4c23.game.ichunqiu.com/f15g_1s_here.php?val=${@eval($_POST[0])}

图片15.png (59.76 KB, 下载次数: 49)

2016-11-2 11:11 上传

通过反引号(实际调用shell_exec)执行命令获取flag

图片16.png (27.82 KB, 下载次数: 61)

2016-11-2 11:11 上传

图片17.png (31.28 KB, 下载次数: 61)

2016-11-2 11:11 上传

3、Nothing

[PHP] 纯文本查看 复制代码 http://f7bda9a07e5e4ea79ab455b17f40b4f93c86166b84884e4d.game.ichunqiu.com/

页面打开显示ICHUNQIU BACKDOOR 没发现什么地方 感觉应该是个后门有一个docker镜像地址 下下来导入 [Bash shell] 纯文本查看 复制代码docker load < backdoor.tar

图片18.png (9.29 KB, 下载次数: 70)

2016-11-2 11:13 上传

图片19.png (1.94 KB, 下载次数: 59)

2016-11-2 11:13 上传

然后翻web目录 .info.php发现是个phpinfo猜测一般套路 是写了个扩展后门 grep backdoor 发现是个扩展，之后

查看php.ini

图片20.png (19.43 KB, 下载次数: 69)

2016-11-2 11:14 上传

图片21.png (59.48 KB, 下载次数: 47)

2016-11-2 11:14 上传

图片22.png (2.75 KB, 下载次数: 66)

2016-11-2 11:15 上传

下载下来拖到IDA里头使用F5大法挨个查看

图片23.png (42.32 KB, 下载次数: 58)

2016-11-2 11:15 上传

初步判断是POST接受个参数 然后进行eval

猜测了下是不是string参数

图片24.png (51.24 KB, 下载次数: 53)

2016-11-2 11:17 上传

然后老套路执行命令去web目录读取flag.php

图片25.png (25.62 KB, 下载次数: 57)

2016-11-2 11:17 上传

图片26.png (32.03 KB, 下载次数: 65)

2016-11-2 11:18 上传