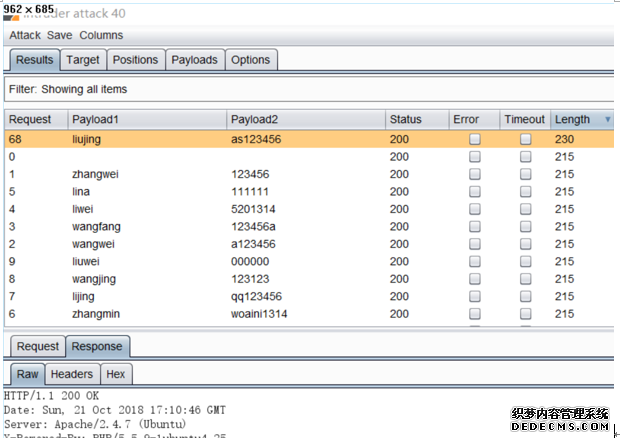
1.考点：身份验证

查看源码得到提示www.zip, 下载解压得到一个username.txt和password.txt, 分别有100个用户名和100个密码，这里在burpsuite里使用Pitchfork模式，然后单线程延时2000毫秒可以爆破成功。

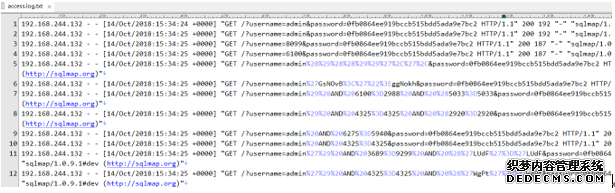


2.考点：访问控制

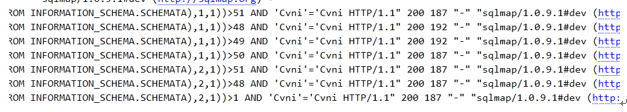
注册账号登录系统后，有个查看个人信息的功能，将uid改为1即可获取flag

3.考点：安全审计

打开页面，提供有一个sqlmap进行注入的访问日志，具体过程是一个盲注的过程。



将所有内容url decode后，分析注入过程，可知页面长度为192的时候所执行的SQL语句查询为true，187的长度时查询为false。



可知逐位注入的最后一次查询时，长度为192时，说明“>”比较为true，真实ascii码值就是该位+1，长度为187时，说明“>”比较为false，真实ascii码值就是该位的值。

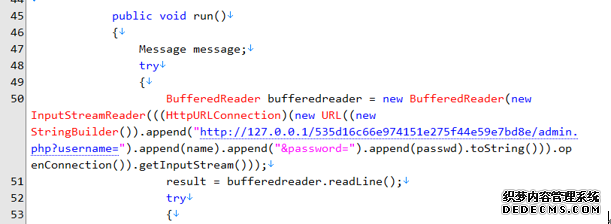
从而可以获取到注入过程中的数据:

Path: /apkdownload/apk.apk

Username: admin

Password: 0fb0864ee919bccb515bdd5ada9e7bc2

下载到apk后，逆向可知存在接口：

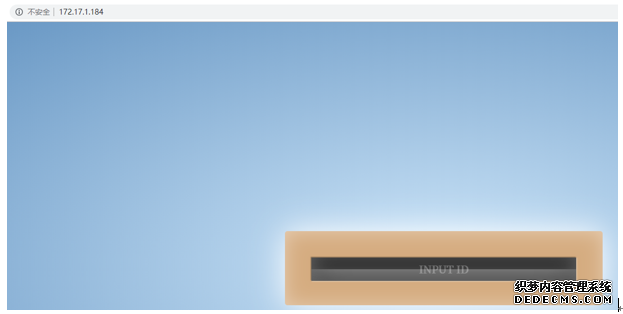


直接访问该接口获取flag



4.考点：通信保密性

页面打开如下：

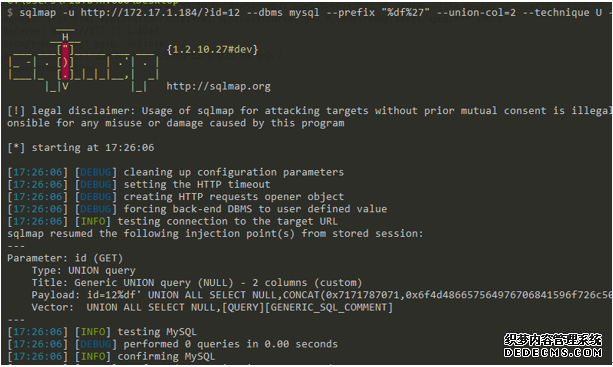


仅有一个输入框，输入id=1和id=2内容不同，其他没有发现有注入，经过多次尝试，发现页面存在宽字节注入，可以直接union注入获取数据：

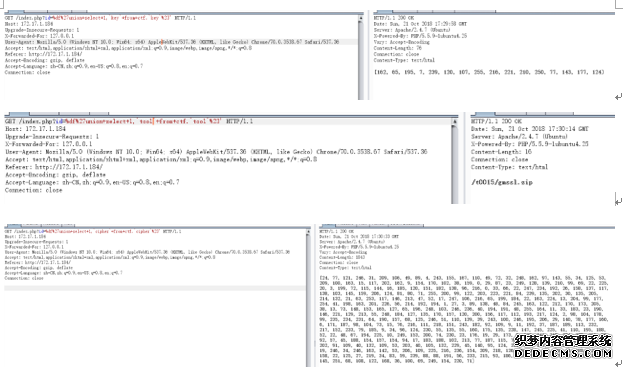


利用sqlmap可以识别出存在注入：

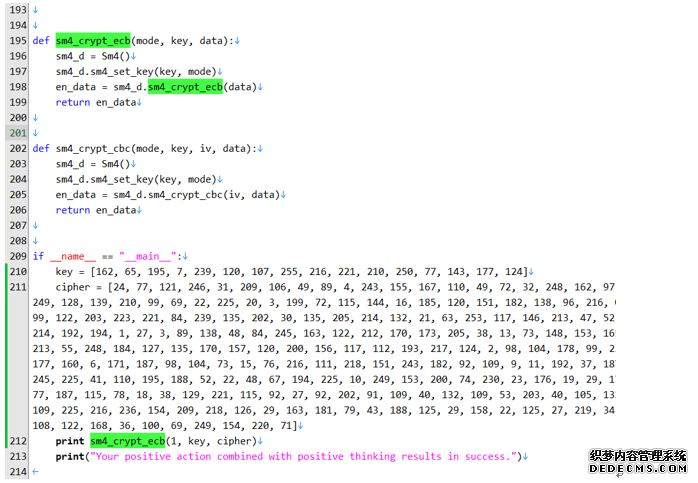
sqlmap -u http://172.17.1.184/?id=12 --dbms mysql --prefix "%df%27" --union-col=2 --technique U -v 3



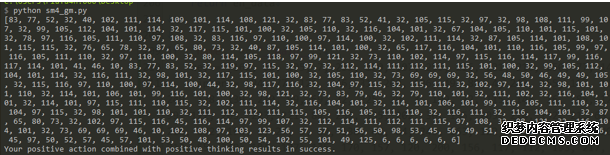
后续可以获取出当前数据库ctf存在3个表，tool、cipher和key。这里sqlmap无法获取数据，直接手工获取数据：



访问http://172.17.1.184/t0015/gmssl.zip下载解压得到一个py文件，内容是sm4国密算法的实现，有了key和cipher，直接在py脚本中添加代码：



运行得到：



5.考点：软件容错

从题目内容得知是weblogic的反序列化漏洞，访问http://172.17.1.205:7001/wls-wsat/CoordinatorPortType11发现页面存在，猜测应该是CVE-2017-10271漏洞

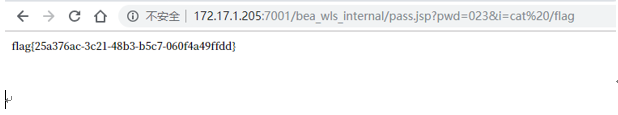
Payload直接写入webshell：



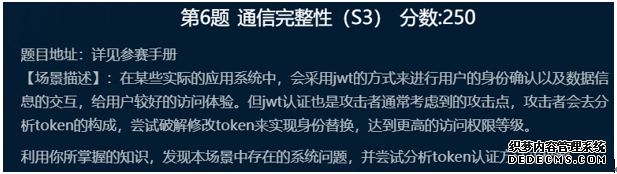
Webshell地址：http://172.17.1.205:7001/bea\_wls\_internal/pass.jsp?pwd=023&i=whoami



获取flag：



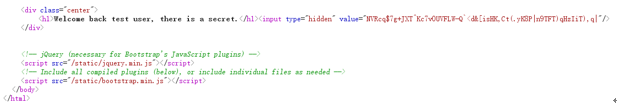
6.考点：通信完整性



使用test/test登录，得到cookie

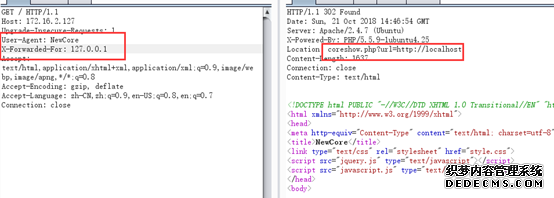
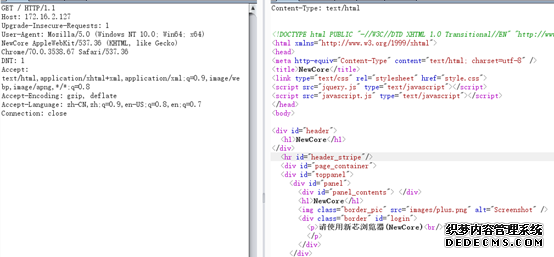


Auth的第一段解base64得到{"alg":"HS512","typ":"JWT"}，第二段解base64得到{"username":"test"}，知道是jwt认证，那么改下username=admin即可，kali里有jwt模块可以计算，其中secret key在页面中

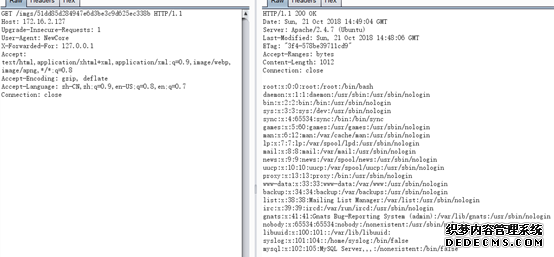
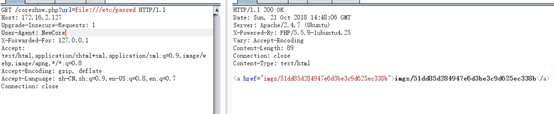


7.考点：资源控制

打开页面提示要NewCore浏览器和本地登录，分别修改UA和XFF头。



这里应该是SSRF漏洞，尝试使用file协议读取本地文件



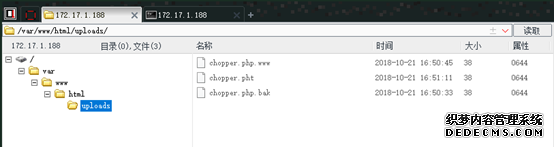
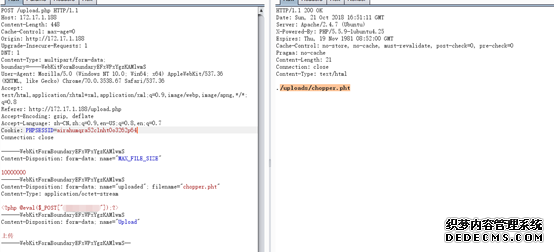
当读取/var/www/html/coreshow.php时，提示要求来自www.newcore.com，于是添加Referer读取coreshow.php，提示源码在core.php中，再读取core.php获取flag.

8.综合题

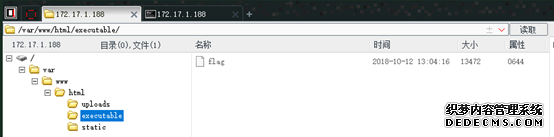
页面打开为一个登录页面login.php，使用万能密码admin’ or ‘a’=’a/1可以登录，登录后为一个上传页面。



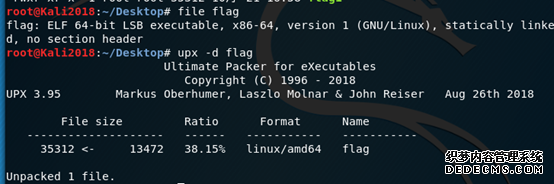
这里的上传文件不会被重命名，尝试解析漏洞没成功，然后用了pht后缀绕过了黑名单限制得到webshell。



目录下有个flag，是个ELF文件，需要进行逆向。



拿回来发现存在upx壳，直接命令行脱掉。



由于时间有限，这题解了一半，没做完。

