/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\* 程序作者：爱的伊苏 \*/

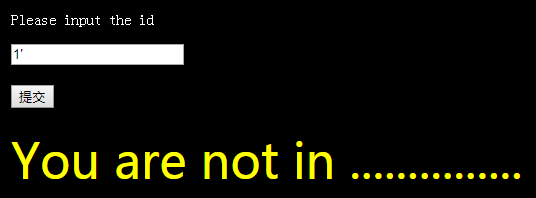
/\* 程序完成时间：2019/1/11 \*/

/\* 有任何问题请联系：740531307@qq.com \*/

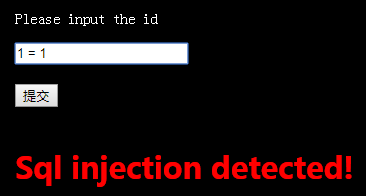
/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

//@永远得意的笑下去





这说明单引号没有被过滤掉



正常是you are in，报错是you are not in，触发waf拦截是sql injection detected

1.提交1或用语句让框内为真，显示You are in。

2.提交语句有错误或语句让框内为假，不报错，但显示You are not in（实质和报错一样）

3.提交某些特殊字符会被过滤并显示sql injection detected,经过各种测试（可用Burp suite进行模糊测试或脚本提交敏感字符）发现过滤的字符有and,空格，+，#，union，逗号，

4.提交语句id=1'or'1'='2，如果网页对or没有任何处理的话，应返回You are in（因为后面的语句错误，相当于只提交id=1），但返回的却是You are not in，说明or被处理了。一般的后台处理逻辑是匹配or、or（不分大小写）、or+空格并替换为空。尝试改变大小写和用oorr代替，发现回显都为You are in，也就是说，后台处理应该是匹配or（小写），并将其替换为空，并且仅仅处理了一次。所以在接下来的语句构造中我们可以用oorr，OR等代替or。

database length: 18

ctf\_sql\_bool\_blind

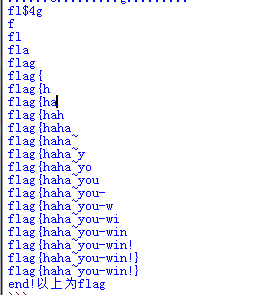
length: 11

fiag@users-

length: 6

fl$4g

flag{haha~you-win!}



其中you win之间的 – 是空格的转义字符，所以在输入的时候以空格代替

参考资料:

基于brupsuite的模糊测试:

http://www.4hou.com/vulnerable/6933.html