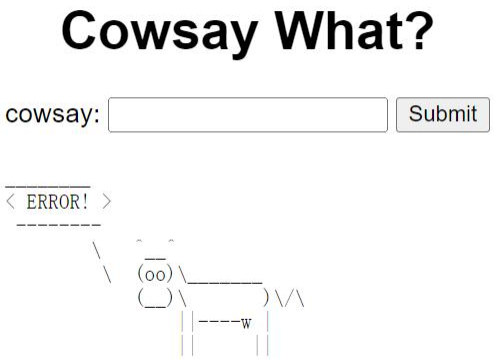
Hgame 2024 week2

-------written by FCowardB

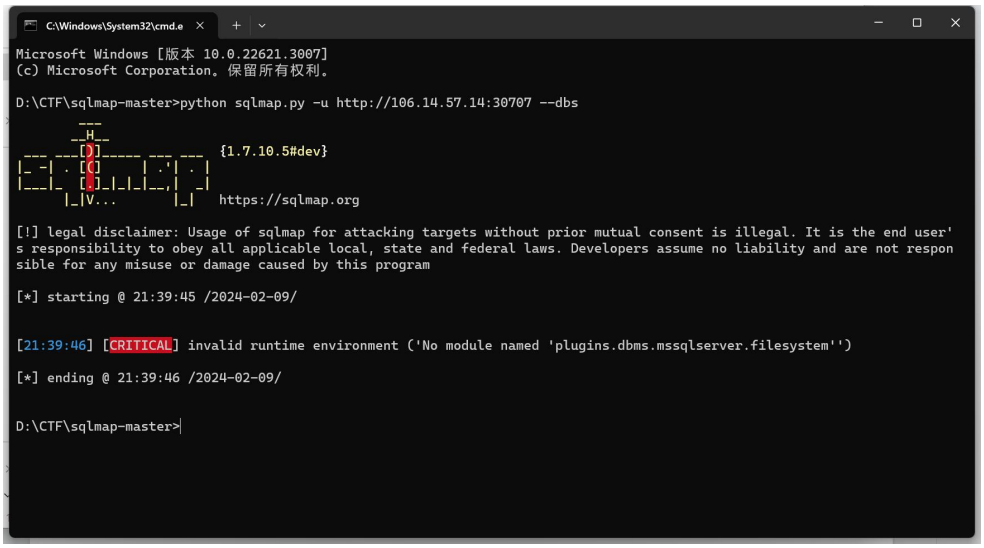
1. what the cow say?

进入环境，只有一个输入框，尝试随便输入东西，就是一头牛说活，说话内容就是输入内容，猜测可能存在注入。

首先考虑 sql 注入，输入 **’**



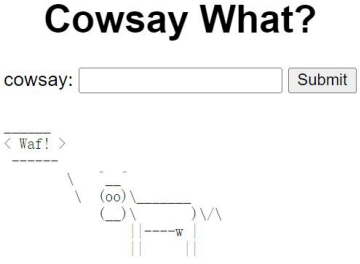
显示 ERROR！　好像有戏，sqlmap 直接怼



怼不出来，应该方向错了

再考虑命令注入漏洞，利用命令连接符 **；等**

11 ；　ls



显示 waf，不太懂，网上搜了一下，waf 是 web 应用防火墙，可以过滤一些符号和关键词，考虑；可能是被过滤了，又尝试了一些其他连接符，|| && | & ，发现都被过滤了，又尝试了 cat flag 这些关键词也被过滤（这两个后面也需要绕过，很简单，只需要把其中一个字母用单引号就行，如 c'a't），卡了很久，最后在学长的提示下搜索 linux 中所有符号的用法（因为无法准确表达我想象的学长所描述的那个字符的功能，所以采用了笨方法）发现$似乎有用，搜了一下具体用法

用法三：

shell中$(( ))、$( )、``与${ }的区别

说明：

${ }这种形式其实与用法一和二是一样的，属于变量替换的范畴，只不过在变量替换中可以加上大括号，也可以不加大括号。

简而言之：$(( ))属于执行计算公式，等价于$[ ]，$( )和` `属于命令替换，${ }属于变量替换

（1）$( )与 ` `(反引号)：返回括号中命令的结果

在bash中，$( )与` `（反引号）都是用来作命令替换的，执行括号或者反引号中的命令。

命令替换与变量替换差不多，都是用来重组命令行的，先完成引号里的命令行，然后将其结果替换出来，再重组成新的命令行

示例：命令：$ echo today is $(date "+%Y-%m-%d")，显示：today is 2014-07-01

注：在操作上，这两者都是达到相应的效果，但是建议使用$( )，理由如下：

1） ` `很容易与"搞混乱，尤其对初学者来说。

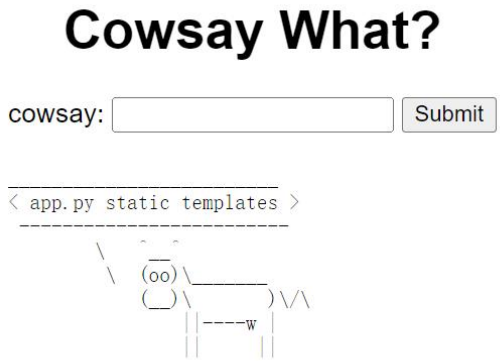
2）在多层次的复合替换中， ` `必须要额外的跳脱处理（反斜线），而$( )比较直观。

最后，$( )的弊端是，并不是所有的类unix系统都支持这种方式，但反引号是肯定支持的。

示例：

```
1# 将cmd1执行结果作为cmd2参数，再将cmd2结果作为cmd3的参数
cmd3 $(cmd2 $(cmd1))
```

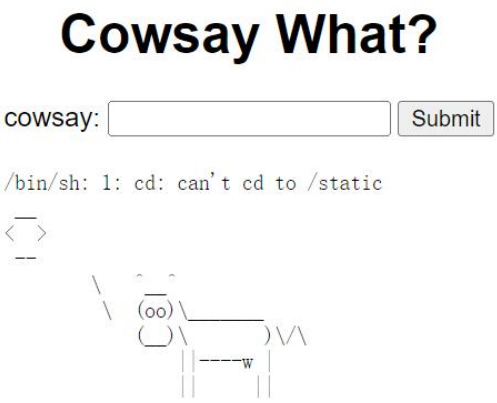
所以直接尝试 **$(ls)** 发现有回显



看到了目录，感觉接近 flag 了，但是我又走错方向，因为我总想着回到根目录，一直尝试 cd / 和 cd .. ，都没有反应，不知道咋回事，又因为显示了 static 和 templates 文件夹，

不懂，搜了一下发现是 springboot 项目创建时默认的两个文件夹，还是尝试一下吧，

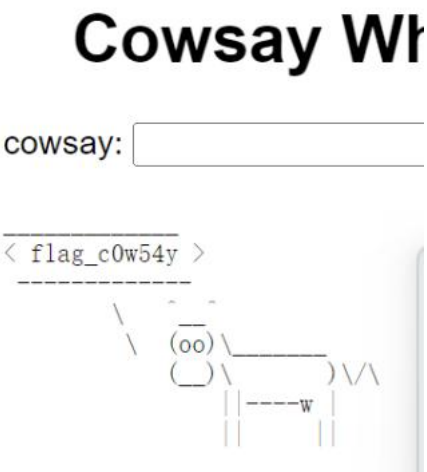
$(cd /static）



不让 cd，感觉有可能是没有权限，又卡了一段时间之后，突然想到 ls 绝对路径可以直接查看目录，

$(ls /)查看根目录

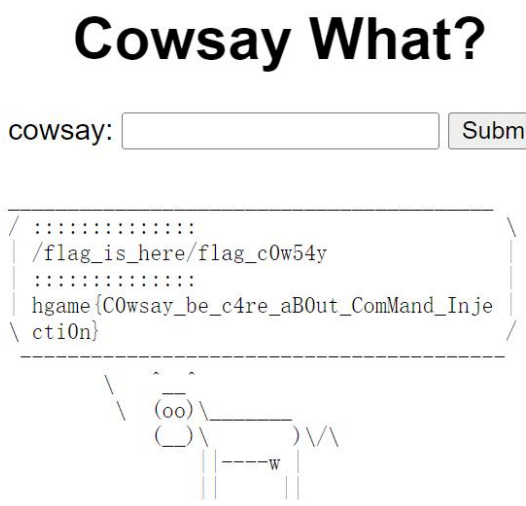
$(c'a't /fl'a'g_is_here)



这时候我又被卡了很久，因为不知什么原因，$(c’a’t /fl’a’g_is_here/fl’a’g_c0w54y)

总是报错，甚至怀疑过 flag_c0w54y 是文件夹，后来想到用 more 代替 cat，就出来了

Payload： **$(more /fl'a'g_is_here/fl'a'g_c0w54y)**



Web 太难了，放弃了，看看其他方向

Misc

Ek1ng_want_gielfriend

第一次做 misc 题，50 分，解出人数很多，应该简单

提示非常明显，用一个工具 wireshark 从 http 流量中提取文件

查查教程，直接怼