实验三

实验环境

本实验使用Linux_x86_64环境，可以自由选择任意Linux 系统LTS运行。建议使用Ubuntu20.04。

如果使用intel芯片Windows/mac 可自行安装虚拟机, 或者使用提供的虚拟机。
如果使用M1芯片MAC 或者其他架构芯片组的Windows/Linux。请先安装qemu，然后使用qemu启动提供的虚拟机。

如果使用自己的系统，请新增一个用户student。并在student用户下完成本次实验。
需要安装`python3` ,`pip`, `gcc/clang`，并自行解决可能出现的编译失败过程中出现的问题，常见为缺少必要链接库，缺少python库等。
如使用提供的虚拟机，使用用自己的电脑按上述方式正确加载并启动虚拟机即，将给定的程序编译运行即可。

```
# M1下qemu使用homebrew下载:
brew install qemu
# 然后编辑使用提供的脚本，删除 -enable-kvm 标志位然后启动即可
./6.858-x86_64-v22.sh
```

实验内容

本次实验分为两个部分：
第一部分：使用fuzz或者其他任意方式寻找代码中存在的buffoverflow（最少三处），并给出对应的POC。
第二部分：挑选出任意一处，并利用给出的ShellCode，写出可以RCE的EXP。

编译运行程序

执行`make`编译程序，然后可以看到会有`zookd-exstack`和`zookd-nxstack` 前者没有开启栈保护，后者有开启栈保护，可作为提高篇内容自行尝试。

为了保证每次运行程序时都有相同的堆栈和内存布局，可以使用`./clean-env.sh ./zookd 8080`来运行程序。

在写完POC之后，可以使用`make check`来检查POC的正确性。
在运行程序之后，可以在浏览器中使用`http://ip_address:8080`打开网站查看程序。

寻找bufferoverflow

可以使用给定的`./exploit-template.py`作为POC模版，也可以使用自己的方式构建POC。
通过使用`make check-crash`来检查POC是否成功使得web服务崩溃。

RCE

可以使用给定的shellcode或者自行构造shellcode，利用发现的bufferoverflow。

要求，在上述poc的基础上进行内存布局，获得一个shell。
提高：修改shellcode，将`execve`系统调用,换为`unlink`系统调用，并删除某一文件。