# 部署ctf pwn题目的时候遇到的坑+经验总结

原创

N1ch0l4s 于 2021-10-14 07:55:12 发布 376 收藏 2

分类专栏： 工具使用 文章标签： docker ctf部署



本文链接：https://blog.csdn.net/weixin_46521144/article/details/120572274



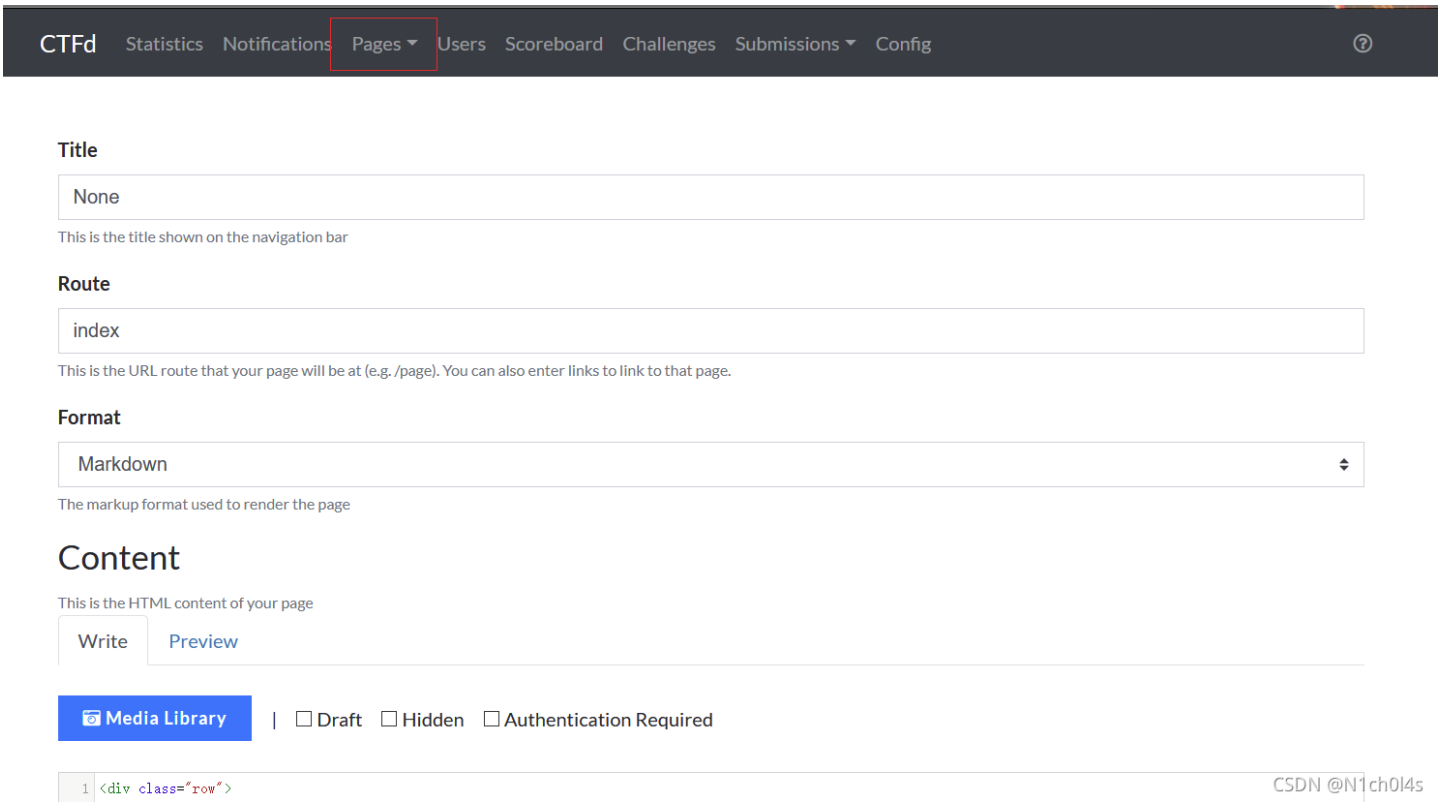


## 前言

还是第一次部署pwn题目，这次一共使用了两个项目一个是ctfd，另一个是ctf-xinetd。重点大概记录一下两个东西咋用。

## ctfd

编辑首页部分 目录栏里面的page 这是看到的首页




注意在放题目的时候要放上连接nc地址，老是忘记。

## xinetd

### 部署题目

- 在bin目录下放好binary和flag（或者flag.txt以及其他的，用户ls之后就能看到这个文件夹的内容）
- 修改ctf.xinetd文件中如下图所示的部分为可执行文件名称，也就是pwn题目的binary文件

```
service ctf
{
    disable = no
    socket_type = stream
    protocol    = tcp
    wait        = no
    user        = root
    type        = UNLISTED
    port        = 9999
    bind        = 0.0.0.0
    server      = /usr/sbin/chroot
    # replace helloworld to your program
    server_args = --userspec=1000:1000 /home/ctf ./solve
    banner_fail = /etc/banner_fail
    # safety options
    per_source  = 10 # the maximum instances of this service per source IP address
    rlimit_cpu  = 20 # the maximum number of CPU seconds that the service may use
    #rlimit_as  = 1024M # the Address Space resource limit for the service
    #access_times = 2:00-9:00 12:00-24:00
}
```



启动docker，使用的命令就是README中的

```
sudo docker build -t "image_name"
sudo docker run -d -p "0.0.0.0:(port):9999" -h "name" --name="name" name
```

测试是否部署成功

```
nc 127.0.0.1 (port)
```

这里注意遇到很坑的地方，第一次部署pwn题可能会碰到，就是一定要设置缓冲区，也就是下面的代码,不然无法输出内容，必须先要用户输入才行。

```
int init()
{
    fflush(stdin);
    fflush(stdout);
    fflush(stderr);
    setvbuf(stdin,0,_IONBF,0);
    setvbuf(stdin,0,_IONBF,0);
    setvbuf(stderr,0,_IONBF,0);
    return alarm(0x14);
}
```

除此以外，不知道为什么加上了这些printf(“aaa\n”)这类的也还输不输出。必须要每次printf之后fflush(stdout)才行

## 有关libc版本

好像docker部署的环境下只能跑最新的libc版本，其中2.23，2.27和2.31分别对应Dockerfile中的顶端ubuntu版本16.04，18.04，20.04。因此要记住最好新出的题目要拿最新的libc，不然后面可能还要重做一遍（萌新不会用别的工具放题目555）

```
FROM ubuntu:16.04

RUN sed -i "s/http:\/\/archive.ubuntu.com/http:\/\/mirrors.tuna.tsinghua.
    apt-get update && apt-get -y dist-upgrade && \
```

```
    apt-get install -y lib32z1 xinetd

RUN useradd -m ctf

WORKDIR /home/ctf

RUN cp -R /lib* /home/ctf && \
    cp -R /usr/lib* /home/ctf

RUN mkdir /home/ctf/dev && \
    mknod /home/ctf/dev/null c 1 3 && \
    mknod /home/ctf/dev/zero c 1 5 && \
    mknod /home/ctf/dev/random c 1 8 && \
    mknod /home/ctf/dev/urandom c 1 9 && \
    chmod 666 /home/ctf/dev/*

RUN mkdir /home/ctf/bin && \
    cp /bin/sh /home/ctf/bin && \
    cp /bin/ls /home/ctf/bin && \
    cp /bin/cat /home/ctf/bin

COPY ./ctf.xinetd /etc/xinetd.d/ctf
COPY ./start.sh /start.sh
RUN echo "Blocked by ctf_xinetd" > /etc/banner_fail

RUN chmod +x /start.sh

COPY ./bin/ /home/ctf/
RUN chown -R root:ctf /home/ctf && \
    chmod -R 750 /home/ctf && \
    chmod 740 /home/ctf/flag

CMD ["/start.sh"]

EXPOSE 9999
```



如果需要部署i386的文件，需要这样写

这里对应的就是系统初始化的命令部分。

```
FROM ubuntu:16.04

RUN apt-get update && apt-get upgrade && \
    dpkg --add-architecture i386 && \
    apt-get update && apt-get upgrade && \
    apt -y install libc6-i386 libc6-dev:i386 libc6-dbg:i386&& \
    apt-get install -y lib32z1 xinetd

RUN useradd -m ctf

WORKDIR /home/ctf

RUN cp -R /lib* /home/ctf && \
    cp -R /usr/lib* /home/ctf

RUN mkdir /home/ctf/dev && \
    mknod /home/ctf/dev/null c 1 3 && \
    mknod /home/ctf/dev/zero c 1 5 && \
    mknod /home/ctf/dev/random c 1 8 && \
    mknod /home/ctf/dev/urandom c 1 9 && \
    chmod 666 /home/ctf/dev/*

RUN mkdir /home/ctf/bin && \
    cp /bin/sh /home/ctf/bin && \
    cp /bin/ls /home/ctf/bin && \
    cp /bin/cat /home/ctf/bin
```

```
COPY ./ctf.xinetd /etc/xinetd.d/ctf
COPY ./start.sh /start.sh
RUN echo "Blocked by ctf_xinetd" > /etc/banner_fail

RUN chmod +x /start.sh

COPY ./bin/ /home/ctf/
RUN chown -R root:ctf /home/ctf && \
    chmod -R 750 /home/ctf && \
    chmod 740 /home/ctf/flag

CMD ["/start.sh"]

EXPOSE 9999
```



## 删除题目

在大多数情况下（这里我几乎每一题要布置四五次...）布置会失败，需要删除image以及容器来重新布置

```
sudo docker ps #查看所有开放端口
sudo docker stop XXXXX #停止XXXXX开放端口
sudo docker container ls -a # 查看所有容器
sudo docker rm XXXXX # 删除当前容器
sudo docker image ls # 查看所有镜像
sudo docker rmi XXXXX # 删除当前镜像
```