搭建CTF-AWD训练平台

原创

下载

```
git clone https://github.com/zhl2008/awd-platform
```

项目大小是429.03M，下载了整整一个小时

项目说明

AWD线下环境启动说明

> by Hence Zhang @Lancet

比赛的环境介绍：

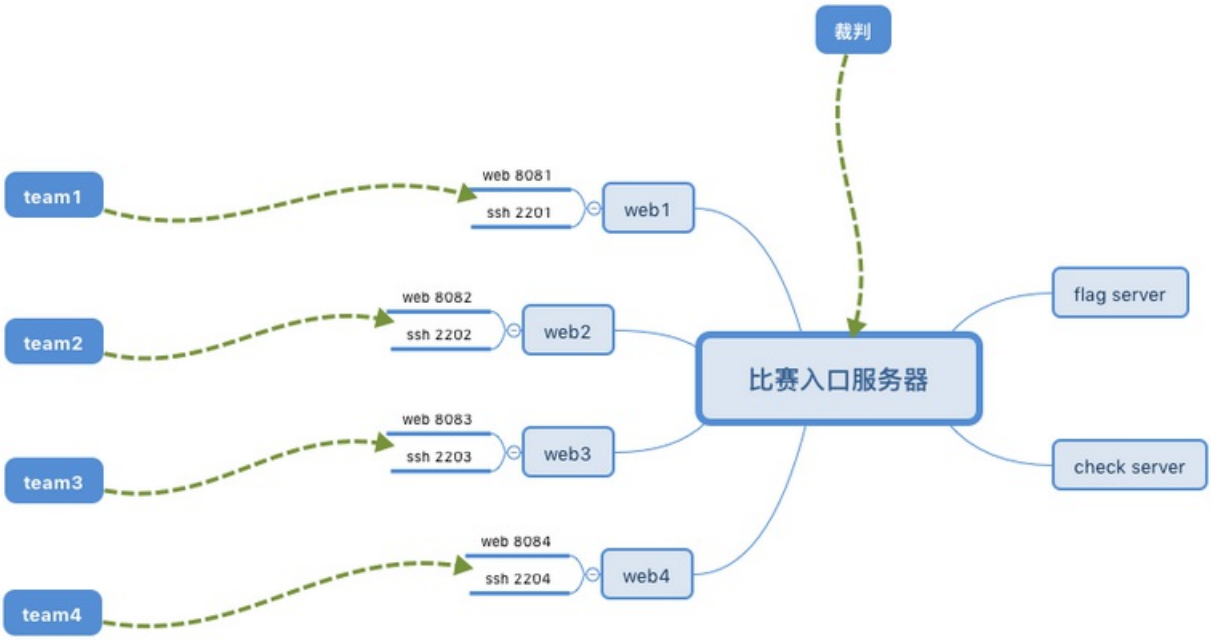
服务器全部以docker形式部署在同一台虚拟机上。

Check_server:
服务检查服务器，用于判定选手维护的服务是否可用，如果不可用，则会扣除相应的分数。不开启任何端口。需要与flag服务器通信。

Flag_server:
选手提交flag的服务器，并存储选手的分数。开启80端口。

Web_server:
选手连接的服务器，选手需要对其进行维护，并尝试攻击其他队伍的机器。通常开启80端口，22端口，并将端口映射到主机。

比赛逻辑拓扑：



比赛入口服务器
比赛启动
1.根据当前队伍数量copy所有的队伍的比赛文件夹:

```
python batch.py web_dir team_number
```

> for example: python batch.py web_server 5

2.启动比赛：

```
python start.py ./ team_number
```

> for example: python start.py ./ 5

3.启动check脚本：

```
docker attach check_server
```

```
python check.py
```

比赛参数

Flag 提交： 172.17.0.6:80/flag_file.php?token=teamx&flag=xxxx (x为你们的队伍号)
比赛规则(新）：

1.每个队伍分配到一个docker主机，给定ctf用户权限，通过制定的端口和密码进行连接；

2.每台docker主机上运行一个web服务或者其他的服务，需要选手保证其可用性，并尝试审计代码，攻击其他队伍；

3.比赛开始后，前30分钟，选手维护各自的主机，在这个阶段，所有的攻击和服务不可用不影响分数；

4.选手可以通过使用漏洞获取其他队伍的服务器的权限，读取他人服务器上的flag并提交到指定的flag服务器：

http://flag服务器IP:端口/fflag_file.php?token=队伍token&flag=获取到的flag 来获得相应的分数。

例如：flag server地址为8.8.8.8，端口为8080，队伍token为team1，flag为40ed892b93997142e46124516d0f5ac0，则请求http://8.8.8.8:8080/fflag_file.php?token=team1&flag=40ed892b93997142e46124516d0f5ac0来获得相应分数。

每次成功攻击可获得2分，被攻击者扣除2分；有效攻击两分钟一轮；

5.选手需要保证己方服务的可用性，每次服务不可用，扣除1分,服务可用，加1分；服务检测两分钟一轮；

6.选手可以从flag服务器上获取所有的攻击情况以及当前的分数：

攻击情况url地址：http://flag服务器IP:端口/result.txt

得分情况地址：http://flag服务器IP:端口/score.txt

7.不允许使用任何形式的DOS攻击
比赛规则(旧）：

1.每个队伍分配到一个docker主机，给定ctf用户权限，通过制定的端口和密码进行连接；

2.每台docker主机上运行一个web服务或者其他的服务，需要选手保证其可用性，并尝试审计代码，攻击其他队伍；

3.比赛开始后，前30分钟，选手维护各自的主机，在这个阶段，所有的攻击和服务不可用不影响分数；

4.选手可以通过使用漏洞获取其他队伍的服务器的权限，并在他人服务器上请求如下地址：

http://flag服务器IP:端口/flag.php?token=队伍token来获得相应的分数。

例如：flag server地址为8.8.8.8，端口为8080，队伍token为team1，则请求http://8.8.8.8:8080/flag.php?token=team1来获得相应分数。

每次成功攻击可获得2分，被攻击者扣除2分；有效攻击两分钟一轮；

5.选手需要保证己方服务的可用性，每次服务不可用，扣除1分,服务可用，加1分；服务检测两分钟一轮；

6.选手可以从flag服务器上获取所有的攻击情况以及当前的分数：

攻击情况url地址：http://flag服务器IP:端口/result.txt

得分情况地址：http://flag服务器IP:端口/score.txt

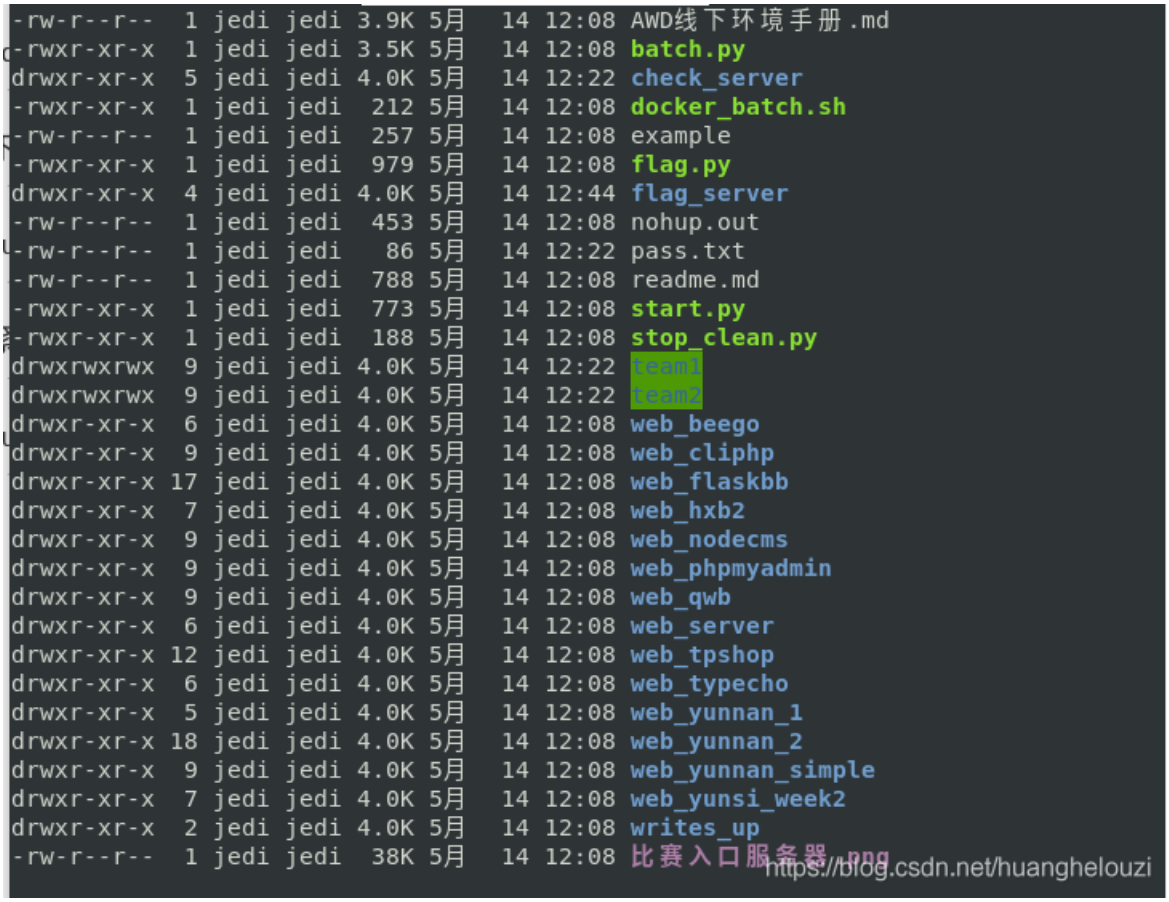
7.不允许使用任何形式的DOS攻击

配置

下载成功之后进入项目目录

```
cd awd-platform
```

项目目录，其中的web_xxxx为awd赛题



下载docker镜像

```
sudo docker pull zhl2008/web_14.04
```

紧接着就是修改docker镜像的名称（不改名称的话请去修改代码）

```
sudo docker tag zhl2008/web_14.04 web_14.04
```

然后就可以按照文档里面的说明进行操作

1.根据当前队伍数量copy所有的队伍的比赛文件夹:

```shell
python batch.py web_dir team_number
```

> for example: python batch.py web_server 5

2.启动比赛:

```shell
python start.py ./ team_number
```

> for example: python start.py ./ 5

3.启动check脚本:

```shell
docker attach check_server
```

```shell
python check.py
```


例如：我创建两个队伍，使用的赛题是`web_yunnan_simple`

```
python batch.py web_yunnan_simple 2
```

启动比赛

```
python start.py ./ 2
```

check模块有问题的，不能正常使用，需要的可以自行依据赛题环境修改check代码，规则大概是文件存在check通过，不存在check失败扣分。

```
$ sudo python start.py ./ 2
ERROR: ld.so: object 'libproxychains.so.3' from LD_PRELOAD cannot be preloaded
cannot open shared object file): ignored.
599fa8789806f7049ace8d33084b8686324ee29f6247cfbbc74496b08546dc99
[*] start docker team1
e14540fd120524afe0ac27ae15bf219e58d0d34d9f71e8b3db19ff25f24e3d6d
[*] start docker team2
e47397d2dd51985f47ead4cff07759592a306f0e7e18a180d67365dbab9ee2f4
[*] start docker check_server
41bd1774b19d44eb7f8be4e82dde467594b9e182aa812c2e439e06b60f3ebd9c
[*] start docker flag_server
```


然后就会在项目根目录下会生成以队伍名称命名的目录

```
drwxrwxrwx  9 jedi jedi 4.0K 5月   14 12:22 team1
drwxrwxrwx  9 jedi jedi 4.0K 5月   14 12:22 team2
```

靶机端口映射规则

```
team1 - 8801
team2 - 8802
...
```

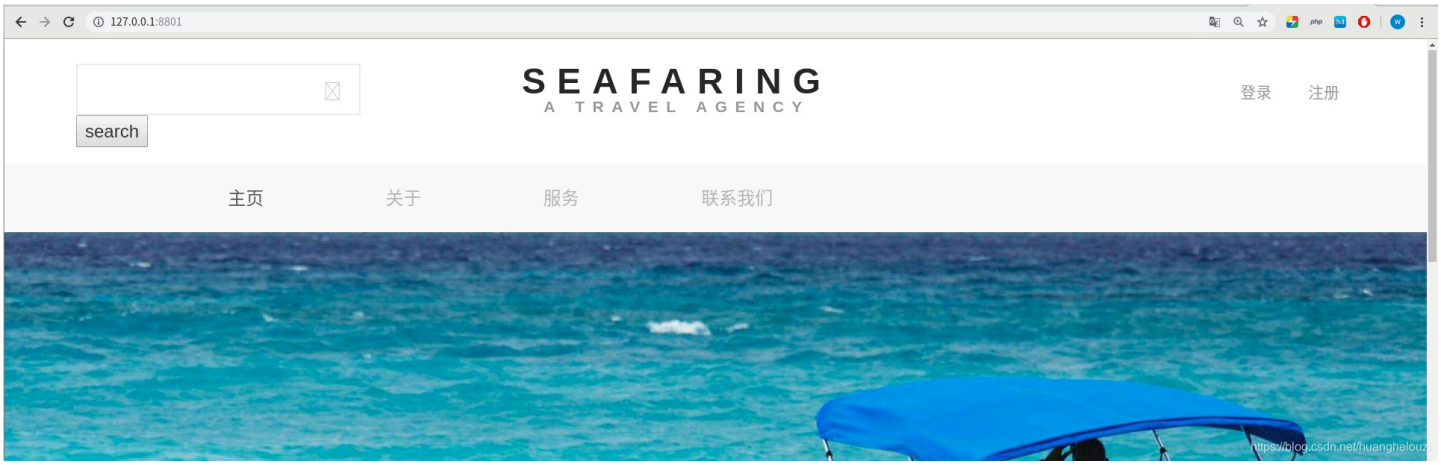
ssh端口映射规则

```
team1 - 2201
team2 - 2201
...
```

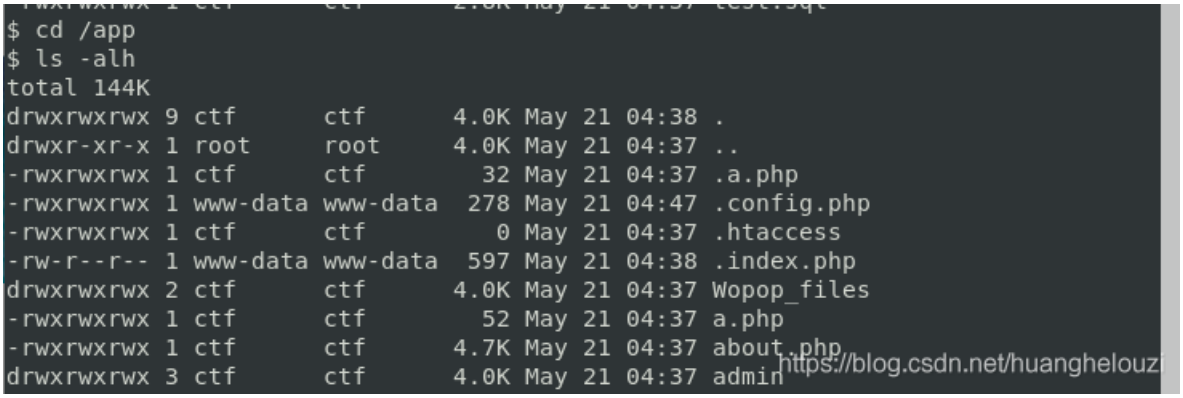
ssh连接密码 `awd-platform/pass.txt`

```
team1:ctf:6f1704cbabd4e013bcdbd979665f2a9b
team2:ctf:ab0253740d7974fefb474f2f7e2da2c9
```

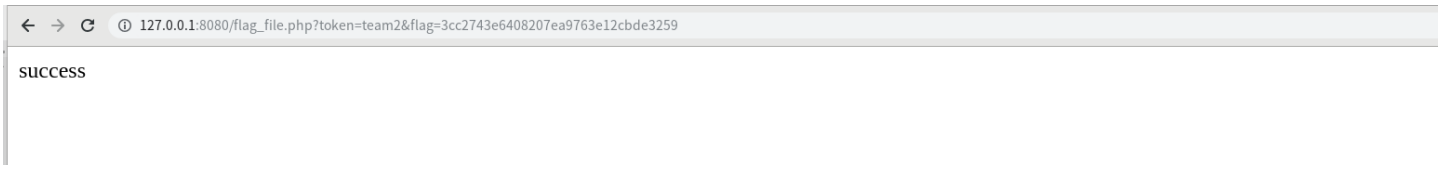
至此，赛题环境全部搭建起来了，下面是效果演示：
web界面



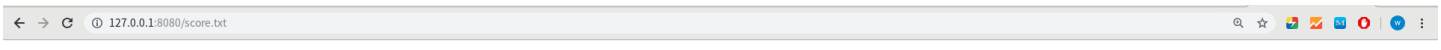
ssh 登陆



提交flag



初始分数榜（如果提交flag之后分数没有变，请修改score.txt的权限），有点low，可以自己写一个排行榜。



比较懒直接使用dalao@夜莫离的模板直接修改了。

比赛或者练习结束之后，需要清除docker环境，可以使用下面的命令（注：初始项目执行会直接删除所有的CONTAINER，需要自己修改代码，血的教训）

```
sudo python stop_clean.py
```

踩坑有点多，完