AWD平台搭建

转载

weixin_30367873 于 2019-08-10 17:07:00 发布 5089 收藏 28
文章标签： 运维 php python
原文链接：http://www.cnblogs.com/Triangle-security/p/11332223.html

因为之前是被AWD比赛坑过，所以想着自己搭建一下啦。这样方便可以本校和某高校，两个学校的人进行相互交流，共同进步。

搭建前提：

ubuntu16.04镜像，这个最好放到虚拟机里面进行加载。(附:Ubuntu的源可以换成阿里云的，这样的话下载速度就会比较快，这里我是给出别人的博客，我是换成阿里云的源

https://blog.csdn.net/lym152898/article/details/79100507

在虚拟机里面启动好之后，然后再加载docker容器，这个容器可以存放我们比赛时候的ctf环境。启动比赛环境还是挺容易的，就是模板的问题，下面是搭建的步骤。

1.安装docker环境：

正常ubuntu里面是没有docker环境的，这样我们就需要下载一个docker环境。因为是linux系统，所以还是命令行下载。不过我就不演示了，直接放出别人的博客吧。如果博客失效了，大家还可以是百度下ubuntu系统安装docker环境。

https://www.cnblogs.com/jiyang2008/p/9014960.html

2.下一步就是克隆项目

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

3.进入项目

```
sudo cd awd-platform/
```

4.下载镜像，木有镜像等于白搭

```
sudo docker pull zhl2008/web_14
```

5.要以root权限的用户，进入到这个目录里面，如果是root权限的话会是一个 # 而不是一个 $ 。切换用户的指令是 su 用户

6. 启动镜像

```
# python batch.py web_yunnan_simple 3//复制3个web_yunnan_simple的靶机，数值可改
# python start.py ./ 3 //启动三个docker靶机和check服务器、flag_server服务器。数值可改
```

7. 在当前目录下，连接裁判机

```
docker attach check_server
python check.py
```

项目的check.py是有问题的，比如无法正常启动，还有check的也是不怎么规范，所以还是修改一下，这里是宕的别人的。

```
#!/usr/bin/env python
# -*- coding:utf8 -*-
'''

'''
import hashlib
import base64

sleep_time  = 300
debug = True
headers = {"User-Agent":"Mozilla/5.0 (Windows NT 6.1; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/60.0.3112.90 Safari/537.36"}

import time
import httplib
import urllib2
import ssl

my_time = 'AAAA'
__doc__ = 'http(method,host,port,url,data,headers)'
flag_server = '172.17.0.1'
key = '744def038f39652db118a68ab34895dc'
hosts = open('host.lists','r').readlines()
user_id = [host.split(':')[0] for host in hosts]
hosts = [host.split(':')[1] for host in hosts]
port = 80

def http(method,host,port,url,data,headers):
    con=httplib.HTTPConnection(host,port,timeout=2)
    if method=='post' or method=='POST':
        headers['Content-Length']=len(data)
        headers['Content-Type']='application/x-www-form-urlencoded'
        con.request("POST",url,data,headers=headers)
    else:
        headers['Content-Length'] = 0
        con.request("GET",url,headers=headers)
    res = con.getresponse()
    if res.getheader('set-cookie'):
        #headers['Cookie'] = res.getheader('set-cookie')
        pass
    if res.getheader('Location'):
        print "Your 302 direct is: "+res.getheader('Location')
    a = res.read()
    con.close()
```

```
    return a


def https(method,host,port,url,data,headers):
    url = 'https://' + host + ":" + str(port) + url
    req = urllib2.Request(url,data,headers)
    response = urllib2.urlopen(req)
    return response.read()

def get_score():
    res = http('get',flag_server,8080,'/score.php?key=%s'%key,'',headers)
    print res
    user_scores = res.split('|')
    print "**********************************************************************"
    res = ''

    print res
    print "**********************************************************************"
    return user_scores

def write_score(scores):
    scores = '|'.join(scores)
    res = http('get',flag_server,8080,'/score.php?key=%s&write=1&score=%s'%(key,scores),'',headers)
    if res == "success":
        return True
    else:
        print res
        raise ValueError

class check():

    def index_check(self):
        res = http('get',host,port,'/index.php?file=%s'%str(my_time),'',headers)
        if 'perspi' in res:
            return True
        if debug:
            print "[fail!] index_fail"
        return False

def server_check():
    try:
        a = check()
        if not a.index_check():
            return False
        return True
    except Exception,e:
        print e
        return False

game_round = 0
while True:

    scores = get_score()
    scores = []
    print "-------------------------- round %d ------------------------------"%game_round
    for host in hosts:
        print "----------------------------------------------------------------"
        host = host[:-1]
        if server_check():
            print "Host: "+host+" seems ok"
```

```
            print "Host: "+host+" seems ok"
            scores.append("0")
        else:
            print "Host: "+host+" seems down"
            scores.append("-10")
    game_round += 1
    write_score(scores)
    time.sleep(sleep_time)
```

启动check.py之后，环境应该是这样的。至此，环境基本上就已经搭建完成了。后面还有一些东西可以看看

8.事项及规则：

1.靶机端口规则：(假设虚拟机的ip为192.168.1.1 ， 虚拟机要和真实机要在同一个C段，如果是实验室用的话，可以在vm里面设置一个桥接模式)

Team1：192.168.1.1:8801

Team2：192.168.1.1:8802

Team3：192.168.1.1:8803

……

以此类推

2.各个靶机的ssh密码可以在项目的文件夹下的pass.txt文件中，开始比赛时告知各个选手ssh密码。

SSH的端口规则为：(假设服务器ip为192.168.1.1)

Team1：192.168.1.1:2201

Team2：192.168.1.1:2202

Team3：192.168.1.1:2203

……

以此类推

3.提交flag方法：(假设服务器ip为192.168.1.1)

http://192.168.1.1:8080/flag_file.php?token=teamX&flag=xxxx

(teamX中的X为自己队伍号，flag为其他队伍的flag)

4.记分牌：查看实时分数情况，没做到实时刷新一下(假设服务器ip为192.168.1.1)

http://192.168.1.1:8080

5.攻击情况：(假设服务器ip为192.168.1.1)

http://192.168.1.1:8080/result.txt

9.这里有个异常就是可以无限提交flag，先放出别人的

在一次测试中，发现在一轮的五分钟有效时间内一直提交某个对手的正确flag可以无限加分，在审计一波代码后发现，关键点在这里

config.php:

```
<?php

$team_number = 3;
$user_list = [];
$token_list = array();
$ip_list = array();
for ($i=1; $i <= $team_number; $i++) {
    array_push($user_list,'team'.$i);
    $token_list['team'.$i] = $i - 1;
    $ip_list['172.17.0.'.($i+1)] = $i - 1;
}

$key = '744def038f39652db118a68ab34895dc';
$time_file = './time.txt';
$min_time_span = 120;
$record = './score.txt';
```

flag_file.php：

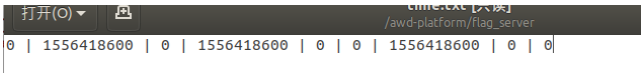
```
require 'config.php';
$now_time = time();
$flag_file = 'xxxxxxxx_flag';

function check_time($attack_uid,$victim_uid){
    global $time_file;
    global $min_time_span;
    global $now_time;
    global $team_number;
    $old_times = explode('|' , file_get_contents($time_file));
    //print $now_time;
    $id = $attack_uid * $team_number + $victim_uid;
    //print $old_times[$id];
    if ($now_time - $old_times[$id] < $min_time_span){
    die("error: submit too quick ". ($min_time_span + $old_times[$id] - $now_time). " seconds left");
    }else{
    return True;
    }
}
```

这边的flag_file.php包含了config.php的配置，即变量$min_time_span和变量$time_file，通过$now_time记录当前时间戳，然后通过与$time_file记录的时间戳节点进行相减，如果符合小于预设的时间差（即一轮多长时间）这一条件则当前时间段无法再次提交flag。

然而 ，$time_file = './time.txt'; 中的time.txt是这样的

```
0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0
```

哈哈 ，为了辨别每一支队伍代表的格子，将其写为0|1|2|3|4.....，然后将变量$old_times输出，经过对比后得出team对应的位置（我这里只找了三个）



此时只需要写一个脚本将五分钟为周期的时间戳更新到time.txt中即可

脚本如下，在启动docker之前五分钟运行，或者后五分钟也可以，改一下脚本内的配置即可，这个脚本是按照某一个时间整点的00 05 10 15 进行记录时间戳的，按自己需要也可以改为其他的

```
#!/usr/bin/env python

#coding:UTF-8

import time

import os


print int(time.time())

Unix_time = int(time.time())

print time.ctime(Unix_time)

while True:

    time_his = []

    time_list = ["00","05","10","15","20","25","30"]

    for i in time_list:

        dt = "2019-04-28 10:"+str(i)+":00"

        time_his.append(dt)

    a = time_his[0]

    b = time_his[1]

    c = time_his[2]

    d = time_his[3]

    e = time_his[4]

    f = time_his[5]

    g = time_his[6]

    time_stamp = [a,b,c,d,e,f,g]

    for k in time_stamp:

            h = open("time.txt", 'w+')

            timeArray = time.strptime(k, "%Y-%m-%d %H:%M:%S")

            timestamp = time.mktime(timeArray)
```

```
            print (int(timestamp))

            data = (int(timestamp))

            separated = '|'

            zero = '0'

            print >>h,(zero),(separated),(data),(separated),(zero),(separated),(data),(separated),(zero),
(separated),(zero),(separated),(data),(separated),(zero),(separated),(zero),

            #           0|data|0|data|0|0|data|0|0

            h.close()

            time.sleep(300)
```

目前这个问题解决了，但是需要一点技巧，就是开启这个脚本的时间要把握好。多调试几次应该就差不多了。

10.感觉这个页面太丑的话，可以换一个页面，这里是夜莫离大佬做的页面

计分板文件拷贝至awd-platform下的flag_server文件夹下。要注意将文件score.txt与result.txt文件权限调至777，这样才能刷新出分值。

还需将scorecard.php文件中的resul变量中的ip地址更改为虚拟机(因为我刚刚已经被坑过了。所以前来补充，不能改成0.0.0.0或者127.0.0.1)的ip地址。

近期更新：这套系统里面不用扫端口，是直接的内网映射到外网。这套系统是找flag的，相当于成功植入webshell之后在受害者的服务器根目录能看到flag，然后把flag进行提交就可以了。每一个容器只有一个flag，并且flag是会变的，这就保证了一个队伍不会被多次攻击的(要是大佬拿到比自己还高的权限，那。。。哭泣泣)。如果想要变动时间，那么就需要check时间和flag刷新时间，因为原版两分钟一次，太快了，所以我把它改为了5分钟。具体修改方法只要将/awd-platform/check_server/gen_flag.py 的time_span 变量设置为5*60即可，也可以改成其他的，同理还有/awd-platform/flag_server/config.php 的 min_time_span变量设置为300、/awd-platform/flag.py 变量time_span设置为5*60。

这套系统里面还有每个docker容器里面连不上数据库的问题，其他基本上都已经解决了，大佬解决的话，给我留言下下，萌新去学习学习。不懂的小伙伴可以留言，我看到后就会回复的。

访问ip:8080/scorecard.php来查看各队得分情况。

计分板源码打包：https://pan.baidu.com/s/18KlIeluaTtm-kT3KuXHseQ

提取码：cvdn

11.玩累了关闭环境的命令

```
python stop_clean.py
```

12.参考链接

https://www.heibai.org/post/1468.html

https://blog.csdn.net/huanghelouzi/article/details/90204325

https://mp.weixin.qq.com/s?__biz=MzU1MzE3Njg2Mw==&mid=2247486325&idx=1&sn=96c04f3609a04260eabdd187fc7c38b1&chksm=fbf79105cc8018131579a

现在搭建后，还是有点问题的，数据库连接不上。比较尴尬啊这个下去之后解决了，我再写出来

----------再次更新----------

每套题里面都有writeup，里面有一套easycms猜测是其他像类似的cms也能搭建出来。选手好像不能连接本地的数据库，但是能注入，暂时还不太清楚是怎么回事。看writeup里面好像能用curl执行得到flag的命令。

----------2019-08-14再次补充----------

看着有那么多人看我博客，还是挺喜欢的，最近又发现了一个AWD平台，有时间的话，我会搭建一下，虽然这个平台功能还能用，但是吧，有个功能好像没实现就是不知道是谁攻击谁，也不知道哪个队伍分数变化最快，这就比较尴尬了。所以另寻他路，就再搭建一个试试吧。另外要搭建的，还是学长(大佬)告诉我的，哇塞，那个平台看着是真的好呀。就是学长说还有一点小bug，唉，有点可惜，不管饿了，到时候搭建个试试。对了，另外的平台是在github上面的，在github上面一搜就有了。

因为搭建的时候也是遇到很多坑，希望后来的人可以少走这些坑。所以坚持不水文