REVERSE

hello_net

送分题，ida+view string 可直接找到 flag;

dynamic_reverse

基础的，反编译，在 message 函数，直接看到 int64 数组和一个 int16 变量，推测为密文，再看 for 循环，易得简简单单对每一位与 0x72 异或即可（注意大小端）

D: > ChroneDownload > C dynamic_reverse.c > main()

```c
#include <stdio.h>
#include <stdlib.h>

int main() {
    char code[42] = {
        31, 19, 11, 17, 6, 20, 9, 32,
        0x17, 4, 0x17, 0, 1, 0x17, 0x2d, 0x3b,
        6, 0x2d, 0x36, 0xb, 0x1c, 0x13, 0x1f, 0x1b,
        0x11, 0x13, 0x1e, 0x1e, 0xb, 0x2d, 0x3b, 0x21,
        0x2d, 0x21, 0x43, 0x1f, 2, 0x1e, 0x17, 0x53,
        0xf, 0x72
    };
    for (int i = 0;i < 42;i++) {
        code[i] ^= 114;
    }
    puts(code);
    return 1;
}
```

maze

套壳迷宫题，upx 脱壳，反编译后找到地图字符串和起点终点，得到答案 sha256 得到 flag

native_android

c++写的 native 层，用 ida 打开 so 文件；
分析可发现是 tea 加密，且 key 按位异或，算出 key 后解密即可

```c
#include <stdio.h>
#include <stdlib.h>
static int DELTA = 0x61C88647;
void tea_decrypt(unsigned int* result, unsigned int* key) {
    unsigned int l = result[0], r = result[1];
    int sum = 0;
    sum -= DELTA * 32; //32次迭代累加后delta的值
    for (int i = 0; i < 32; i++) {

        r -= (16 * l + *(key + 2)) ^ (l + sum) ^ ((l >> 5) + *(key + 3));

        l -= (16 * r + *key) ^ (r + sum) ^ ((r >> 5) + *(key + 1));
        sum += DELTA;
    }
    result[0] = l;
    result[1] = r;
}
int main() {
    char ans[] = {
        0xF0, 0x2E, 0xD2, 0x73, 4, 0xB6, 0x1A, 0xD3, 0x59,
        0xB2, 0xE3, 0x27, 0xA0, 0xDA, 0x90, 0xD6,
        0xD2, 0x8C, 0x1C, 0x24, 0, 0xCB, 0x7E, 0x80, 0x70,
        0xF7, 0x35, 0xC2, 0x9D, 0x30, 0x3A, 0x3F,
        0x53, 0xb6, 0xf7, 0xbc, 0x29, 0x53, 0x5d, 0xec  //大端
    };
    char buf[100] = {'\0'};
    unsigned char key[] = { 0x40, 0x40, 0x6C, 0x67, 0x76, 0x6A, 0x6F, 0x63, 0x57 ,
                            0x47, 0x6B, 0x7F, 0x3D, 0x7B, 0x6B, 0x2E };
    for (int i = 0; i <= 15; ++i)
        key[i] ^= i;
    for (int i = 0;i <= 39;i += 8) {
        tea_decrypt((unsigned int*)(ans + i), (unsigned int*)key);
    }
    puts(ans);
    return 0;
}
```

mixture

很担心出题人的精神状态（）

观察可得函数主体基本是 go 语言，直接把代码贴到在线编译器，观察发现 java 写的函数里有个 PKCS7Unpadding 函数，删掉无关项，go 语言重写该函数即可（源码找不到了）

hello_exception

c++异常处理，ida 分析到 at 函数发现有 out of range 异常，直接将该永真指令改为 jmp 指令到 catch，反编译即可

```
while ( (unsigned __int8)sub_2A64(&v9, v10) )
{
  v11 = (char *)sub_2AC8(&v9);
  if ( *v11 <= 96 || *v11 > 122 )
  {
    if ( *v11 > 64 && *v11 <= 90 )
      *v11 = (*v11 - 52) % 26 + 65;
  }
  else
  {
    *v11 = (*v11 - 84) % 26 + 97;
  }
  sub_2AA4(&v9);
}
qmemcpy(v14, "L1jf[VpbnnUYg}=lw prK[wyau|i=<?", 31);
for ( i = 0; i <= 30; ++i )
  v14[i] ^= i;
```

```
#include <stdio.h>

int main() {
    char code[] = { "L1jf[VpbnnUYg}=lw prK[wyau|i=<?" };
    for (int i = 0;i <= 30;i++) {
        code[i] ^= i;
    }
    for (int i = 0; i <= 30;i++) {
        if (code[i] <= 96 || code[i] > 122) {
            if (code[i] > 64 && code[i] <= 90) {
                code[i] -= 65;
                if (code[i] >= 13) {
                    code[i] += 52;
                }
                else {
                    code[i] += 78;
                }
            }
        }
        else {
            code[i] -= 97;
            if (code[i] >= 13) {
                code[i] += 84;
            }
            else {
                code[i] += 110;
            }
        }
    }
    puts(code);
    return 0;
```

threads

多线程，ida 打开看 main 函数，查看流程图，发现函数 CreateMuteA 无法直接到达，反编译观察源代码发现有 exception : divide zero, 推断是由该异常处理引发跳转，将主函数的 jmp 指令 nop 掉，可得到核心代码的反汇编，注意到打开了两个新线程，各进行一种加密，交替共进行 92 次，然后进入 check 函数中，92 位字符串解密即可得到 flag.

```
hMutex = CreateMutexA(0i64, 0, "secsome's lovely mutex");
if ( hMutex )
{
  handle1 = CreateThread(0i64, 0i64, (LPTHREAD_START_ROUTINE)StartAddress, 0i64, 0, 0i64);
  if ( handle1 )
  {
    handle2 = CreateThread(0i64, 0i64, (LPTHREAD_START_ROUTINE)StartAddress2, 0i64, 0, 0i64);
    if ( handle2 )
    {
      CloseHandle(handle1);
      CloseHandle(handle2);
      while ( count != -1 )
        ;
      CloseHandle(hMutex);
      check();
      result = 0;
    }
    else
    {
      printf("Failed to start Thread2Proc!\n");
      result = 1;
    }
  }
  else
  {
    printf("Failed to start Thread1Proc!\n");
    result = 1;
  }
}
else
{
  printf("Failed to set-up the mutex!\n");
  result = 1;
}
return result;
}
```

```
int main() {
    char code[93] = { "z`l`gc{Ng.fTwxf{@P_ergeeekeied_l1LcOrzzRy]v" };code[43] = 127;char* v11;v11 = &code[44];strcpy(v11, "u_#NqnfGhSsh8");strcpy(v11 +
    v11[23] = 6;v11[24] = 81;v11[25] = 104;v11[26] = 52;v11[27] = 3;[28] = 83;v11[29] = 122;v11[30] = 45;v11[31] = 114;v11[32] = 51;v11[33] = 12;v11[34]
    char* p = &code[84];strcpy(p, "1dNd6`5m}");

    for (int i = 92;i >= 0;i--) {
        switch (i % 2) {
        case 0:if (code[i] <= 96 || code[i] > 122) {
            if (code[i] > 64 && code[i] <= 90) {
                code[i] -= 65;
                if (code[i] >= 13) {
                    code[i] += 52;
                }
                else {
                    code[i] += 78;
                }
            }
        }
              else {
            code[i] -= 97;
            if (code[i] >= 13) {
                code[i] += 84;
            }
            else {
                code[i] += 110;
            }
        }
              break;
        case 1:code[i] ^= i;
        }
    }
    for (int i = 0;i <= 92;i++) {
        printf("%c", code[i]);
    }
    return 0;
}
```

Crypto

easy_md5

对着源码写个脚本即可

cryp.py

home > mihaii > cryp.py

```python
from pwn import *
import os
import hashlib
import string

p = remote('124.220.41.254', 11111)
p.recvline()
p.recvline()
p.recvline()
p.recvline()
p.recvline()
p.sendline(b'1')
p.sendline(b"Boiser")
p.recvline()
p.recvline()
hp = 999999999
while hp > 0:
    p.recvuntil(b"Calculate Md5 and you can play your ace")
    p.sendline(b'1')
    p.recvuntil(b"calculate Md5 of it: \n")
    code = p.recv(numb = 64)
    ans = hashlib.md5(code).hexdigest().encode()
    p.recvuntil(b"your answer is: \n")
    p.sendline(ans)
    print(p.recvline())
    p.recvuntil(b"Calculate Md5 and you can play your ace")
    p.sendline(b'2')
    p.recvuntil(b"Now the dragon's blood volume is: ")
    p.recvline()
    strhp = p.recvline()
    hp = int(strhp)
    print(hp)
p.recvuntil(b"Calculate Md5 and you can play your ace")
p.sendline(b'3')
p.recvuntil(b"The prince asked me to tell you this:\n")
print(p.recvline())
print(p.recvline())
print(p.recvline())
print(p.recvline())
print(p.recvline())
print(p.recvline())
```

PWN

get_my_number
对太后不敬是吧，舰桥雅座一位（）
sizeof 函数的返回值是 unsigned int，有符号数与无符号数比较时会当作无符号数比较
所以输个-1 就拿到 shell 万事大吉
直接 find -name flag 找到 flag

get_my_float
union 共用一片内存，得到 0.618 的 double 二进制数据，用 pwntools p64 打包上传端口即可

cuora_shell
直接生成二进制文件拖进 ida，发现最后执行的是 call 指令，则直接 sh = asm(pwntools.shellcraft)，注意环境，是 32 位还是 64 位
（脚本找不到了）

MISC

鬼人正邪的挑战
什么车车人厨力放出（）
1500 个小碎片显然是图片拼接题
先用 montage 拼 30x50 的样图，发现相邻色块色差很小，合理怀疑碎片编号不是打乱的
众所周知正邪的逆符是让屏幕上下颠倒的，所以合理推测图片是倒序编号的
然后 replace pioneer 反向编号图片接 montage 30x50，得到这个

marisa 是魔理沙，这很 easy
由魔理沙得到第二排应该是 magamaid?
然后是 flag，观察法直接得到
“我爱金发女巫！！！”
（ps:中间一度用 gaps 拼图，拼了大半天得不到正确结果，后来发现一个碎片内部也是四个碎片乱序排的，想在 6000 碎片下跑，结果内存直接满了）

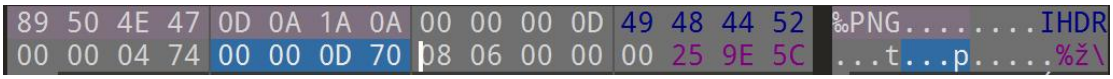
thin_dog
插个耳机听单声道即可

rack your brain
佛又曰 + brainfuck

小紫

大不敬，大不敬（）

图片隐写，010editor 修改 png 的二进制数据进而修改宽高，



同理修改 bmp，不过需要计算，高 = 像素点个数 / 宽，010editor 好心地都标出来了

DWORD biSize	124	Eh	4h	Fg:	Bg:
LONG biWidth	1140	12h	4h	Fg:	Bg:
LONG biHeight	1876	16h	4h	Fg:	Bg:
WORD biPlanes	1	1Ah	2h	Fg:	Bg:
WORD biBitCount	32	1Ch	2h	Fg:	Bg:
DWORD biCompression	3	1Eh	4h	Fg:	Bg:
DWORD biSizeImage	8554560	22h	4h	Fg:	Bg:

new height = (biSizeImage / 4) / 1140 = 1876

修改后打开文件得到 flag

secsome_cfg

cfg:ControlFlowGraph 控制流图

ida 打开汇编代码以流程图打开即可得到含 flag 的图像，观察即可

其中 ($Author) 为典型变量声明，即 secsome

（b 文件太大了，电脑打开卡的头疼）

WEB

no_copy

f12 打开控制台，直接复制元素，替换一下即可得到 flag