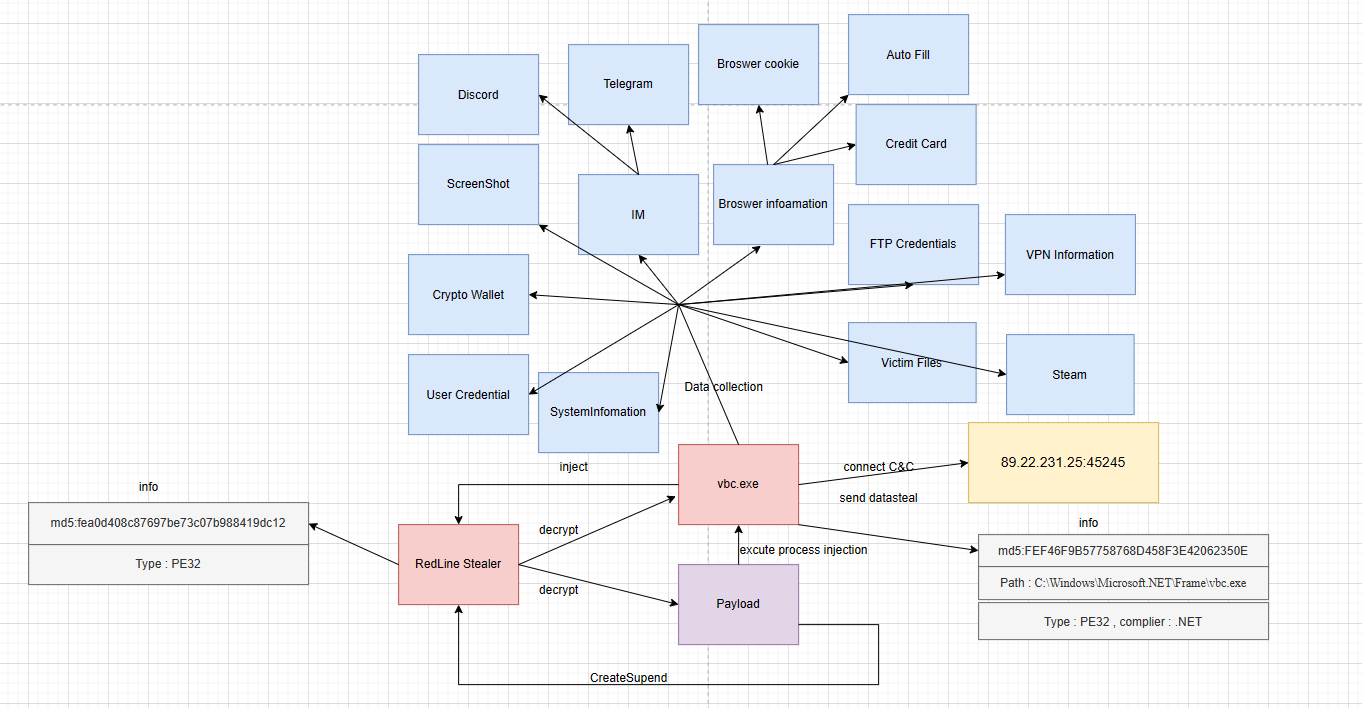
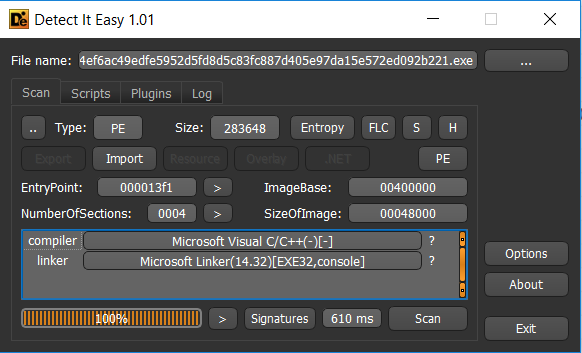
**Phân tích ReadLineStealer**

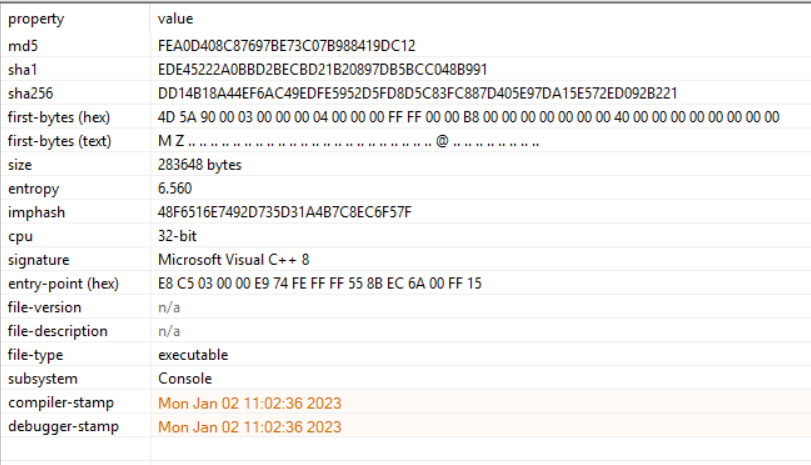
**1 Thông tin chung**

****

DelectIteasy:



Pestudio:

****

**2 MITRE ATT&CK™ Techniques Detection**

Obfuscated Files or Information [Obfuscated Files or Information, Technique T1027 - Enterprise | MITRE ATT&CK®](https://attack.mitre.org/techniques/T1027/)

Shared Modules [Shared Modules, Technique T1129 - Enterprise | MITRE ATT&CK®](https://attack.mitre.org/techniques/T1129/)

Account Discovery [Account Discovery, Technique T1087 - Enterprise | MITRE ATT&CK®](https://attack.mitre.org/techniques/T1087/)

Process Discovery [Process Discovery, Technique T1057 - Enterprise | MITRE ATT&CK®](https://attack.mitre.org/techniques/T1057/)

Screen Capture [Screen Capture, Technique T1113 - Enterprise | MITRE ATT&CK®](https://attack.mitre.org/techniques/T1113/)

Data from Local System [Data from Local System, Technique T1005 - Enterprise | MITRE ATT&CK®](https://attack.mitre.org/techniques/T1005/)

Archive Collected Data [Archive Collected Data, Technique T1560 - Enterprise | MITRE ATT&CK®](https://attack.mitre.org/techniques/T1560/)

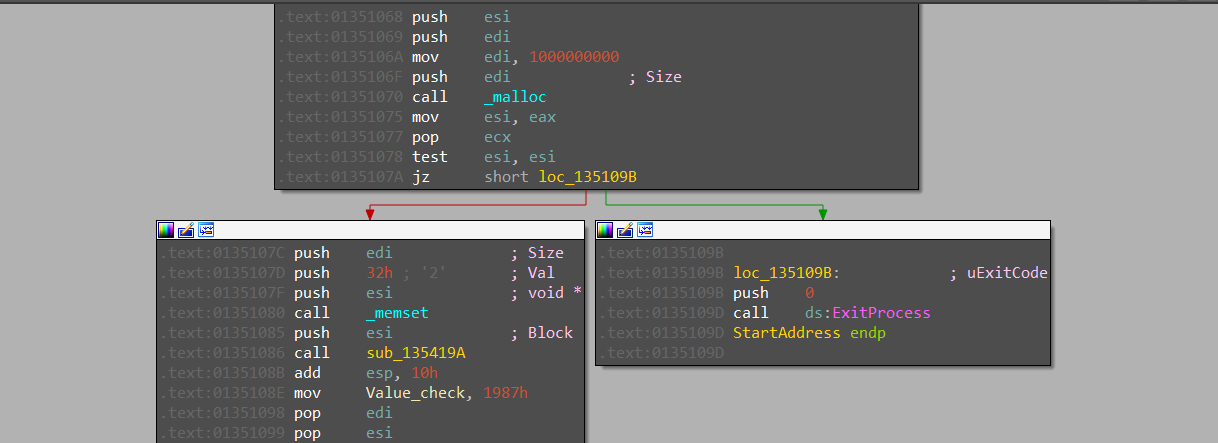
**3 Phân tích chi tiết**

**3.1 Giai đoạn 1**

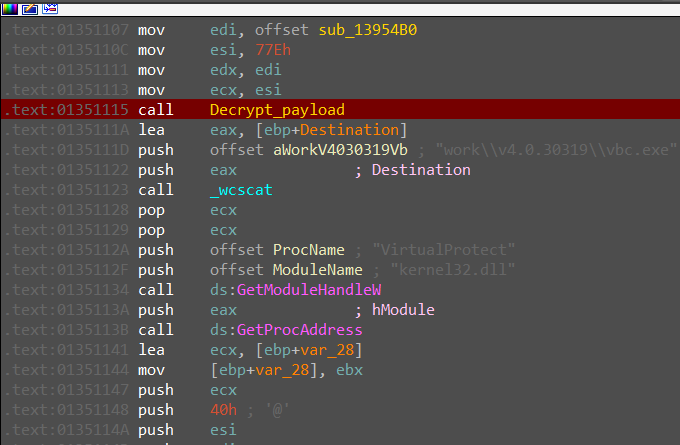
****

Ban đầu mã độc sẽ nối 2 chuỗi thành 1 chuỗi là “C:\Windows\Microsoft.NET\Frame” cho biến Destination . Sau đó mã độc sẽ CreateThread và sleep 3s . Nó sẽ check giá trị Value kia nếu không bằng 6535 thì nó sẽ kết thúc

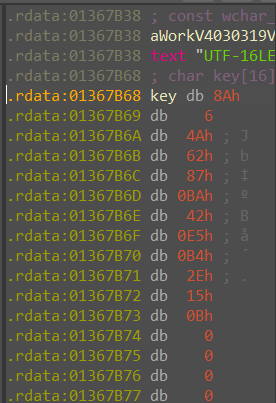
Đi vào hàm StartAddress chúng ta sẽ thấy nó call malloc là 1 hàm cấp phát bộ nhớ động ở đây nó cấp phát khoảng 1 tỷ byte . Nếu thành công thì nó sẽ call memset và hàm sub\_135419A để giải phóng bộ nhớ và sau khi làm xong thì nó sẽ gán giá trị Value\_check là 6535 . Đây có thể coi là 1 cơ chế antiVM vì thông thường các máy ảo cấp phát 1 tỷ byte trong 3s là điều khó có thể xảy ra



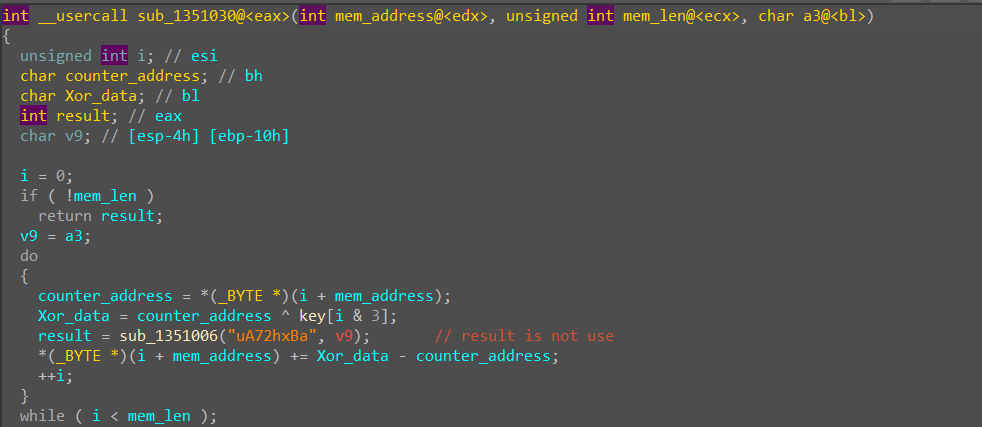
Tiếp theo nó sẽ call hàm Decrypt\_payload và truyền vào offset 13954B0



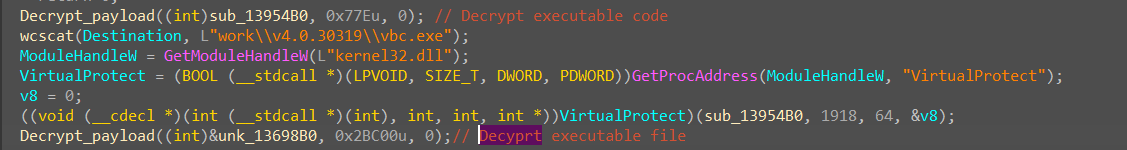
Ở hàm Decrypt\_payload nó sẽ giải mã bằng 1 hàm xor cơ bản với key là 1 mảng byte đã được setup từ trước



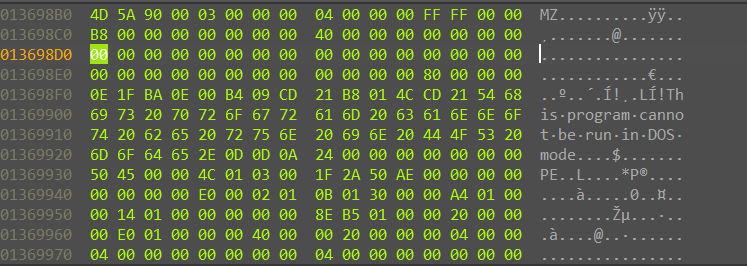
Ở đây nó làm rối việc phân tích bằng cách sử dụng 1 hàm sub\_1351006 và không sử dụng kết quả trả về



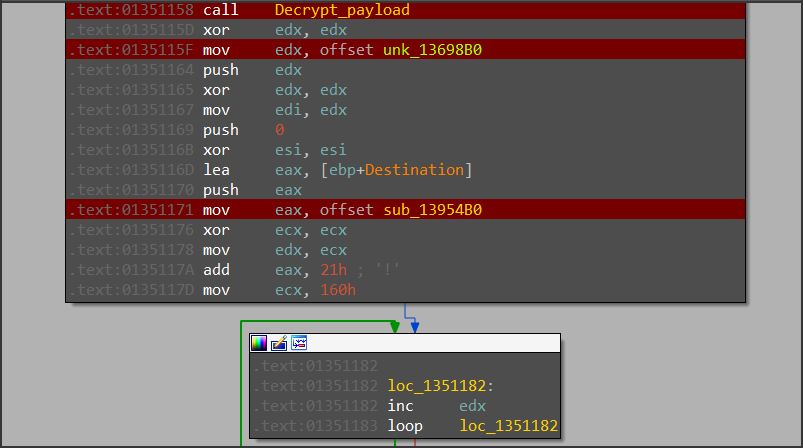
Chúng ta có thể thấy nó thực thi hàm Decrypt\_payload 2 lần . Lần đầu nhằm decrypt payload để thực thi . Lần thứ 2 nhằm decrypt ra 1 file thực thi .



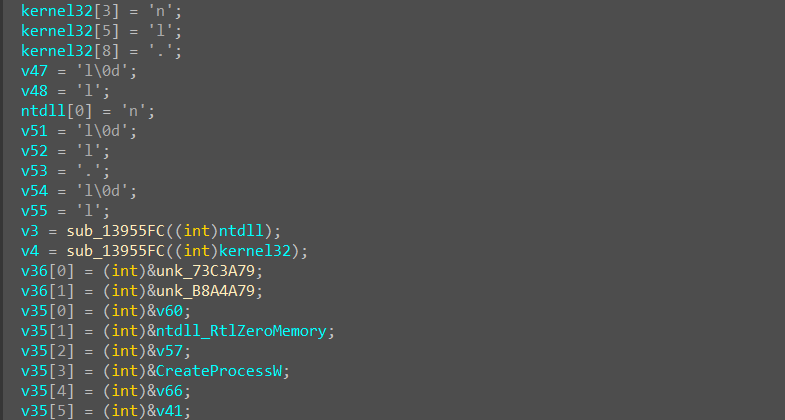
Sau khi thực thi đoạn decrypt thứ 2 thì nó sẽ tạo ra 1 file PE nhằm mục đích thực thi file này dưới cái tên gọi là vbc.exe . Đây là 1 file chuẩn trong Windows\Microsoft.NET nhằm mục đích giả mạo process chuẩn



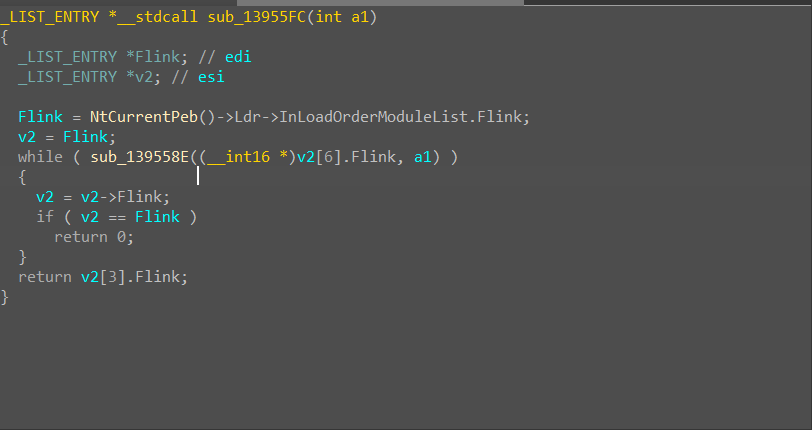
Sau khi giải mã payload thực thi file và file thì nó push địa chỉ lên stack nhằm mục đích sau khi kết thúc hàm main thì sẽ thực thi payload



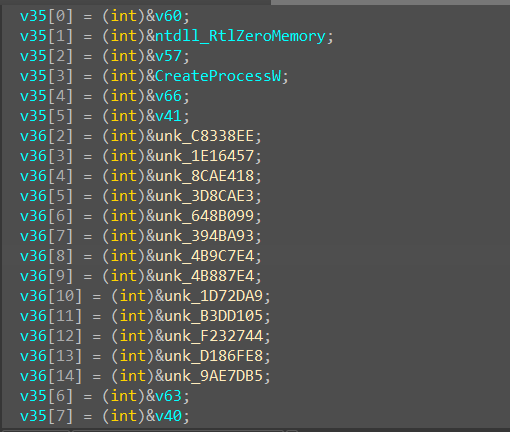
Sau khi nhảy đến payload thực thi nó sẽ call tới sub\_13955FC nhằm mục đích get address của 2 dll là kernel32.dll và ntdll.dll để nhằm mục đích import các API nó cần để thực thi



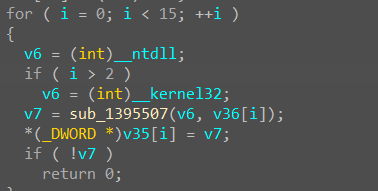
Nó sử dụng PEB structure để call



Nó sẽ gán các offset chính là các giá trị hash vào mảng nhằm mục đích call đến các API cần thiết



Hàm sub\_1395507 chính là hàm tính toán từ các giá trị hash để call tới offset chứa API cần thiết .



Chúng ta có thể nhìn thấy nó lặp 15 lần ứng với nó sẽ import 15 API

kernel32\_ResumeThreadStub

kernel32\_TerminateProcessStub

kernel32\_VirtualAllocStub

kernel32\_SetThreadContextStub

kernel32\_ReadProcessMemoryStub

kernel32\_GetThreadContextStub

kernel32\_CreateProcessWStub

kernel32\_VirtualProtectExStub

kernel32\_VirtualFreeStub

kernel32\_WriteProcessMemoryStub

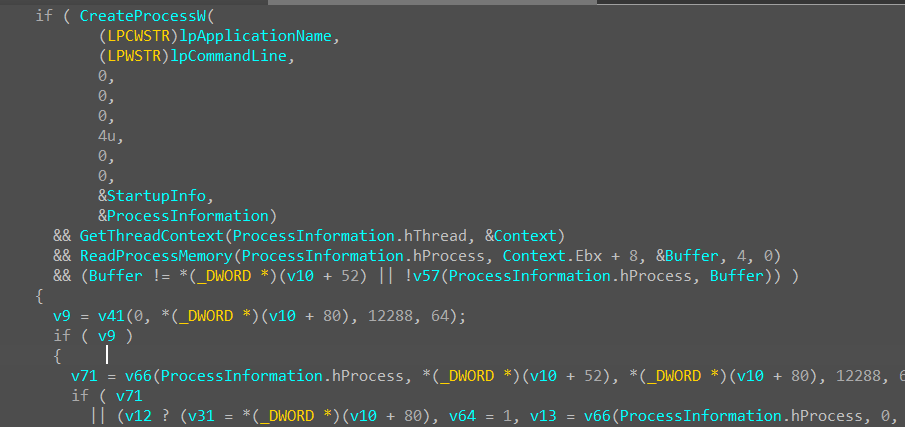
kernel32\_VirtualAllocExStub

kernel32\_CloseHandle

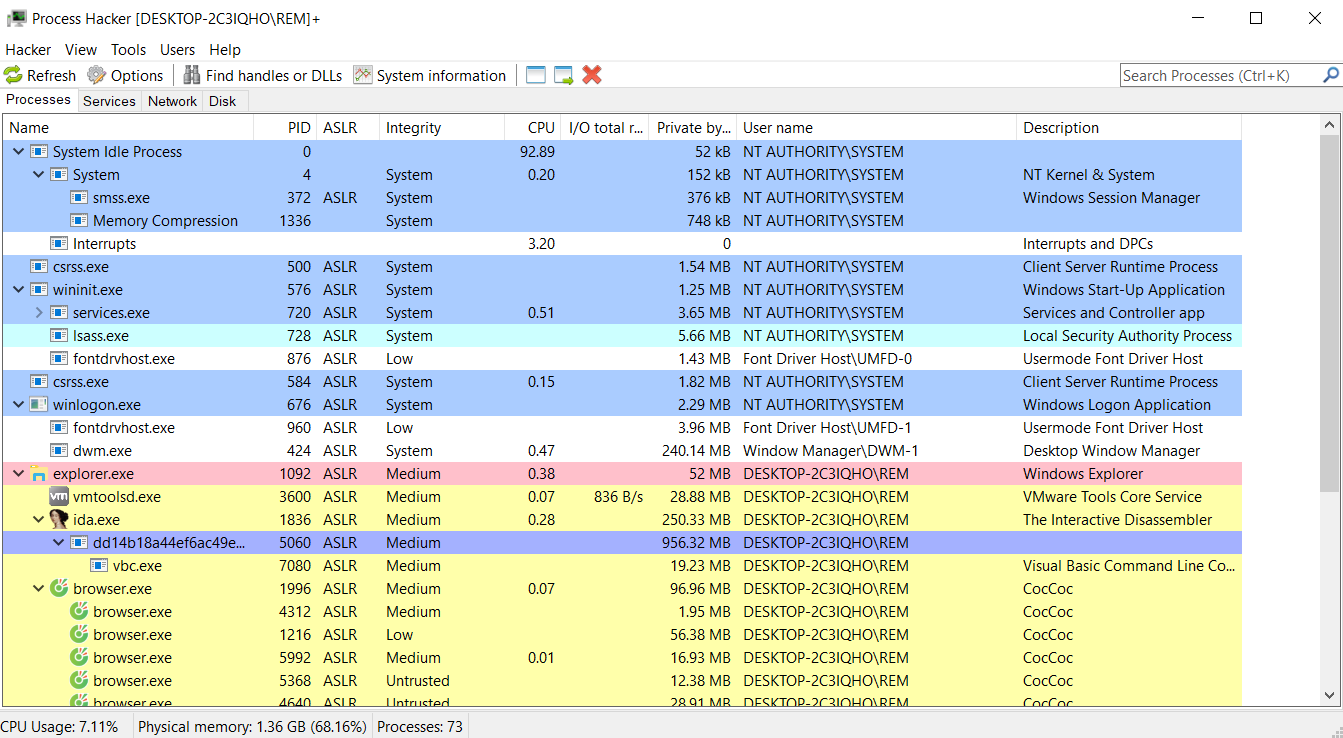
ntdll\_ZwUnmapViewOfSection

ntdll\_memcpy

ntdll\_RtlZeroMemory

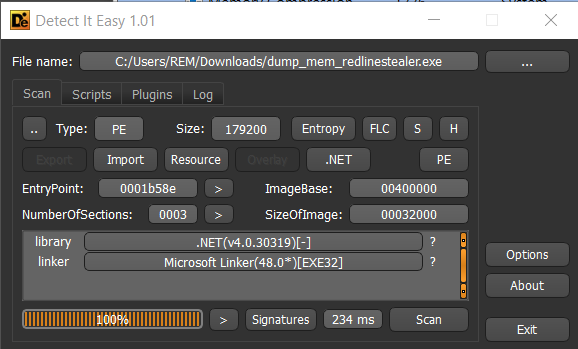
Tiếp theo nó sẽ call tới các API như CreateProcess , ntdll\_memcpy , WriteProcess , …nhằm mục đích là suspend tiến trình ban đầu và inject thực thi tiến trình vbc.exe từ memory 

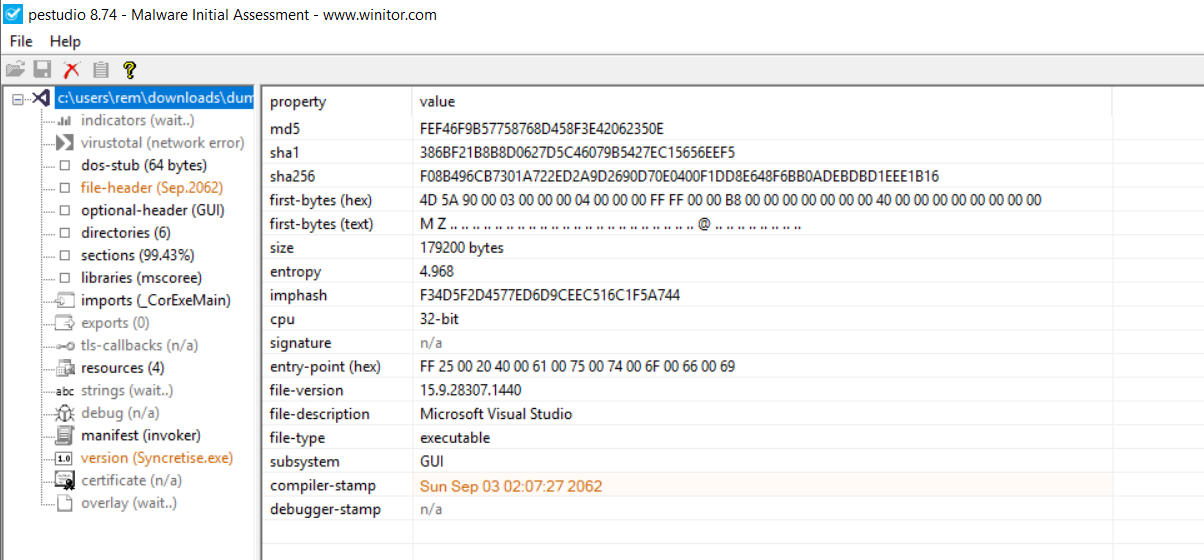
Chúng ta có thể thấy nó đã inject và đã thực thi vbc.exe



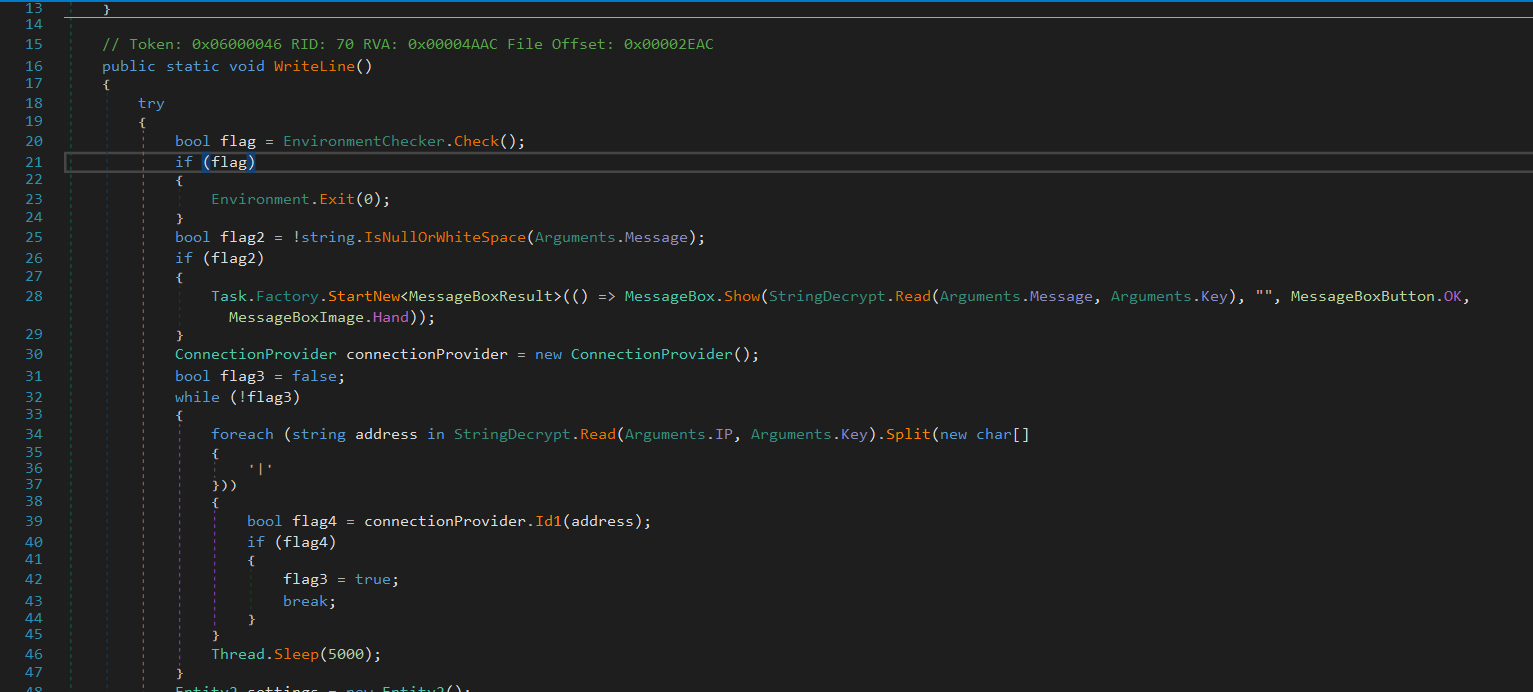
**3.2 Giai đoạn 2**

Để dễ dàng cho việc phân tích tôi đã dump file từ mem ra . DelectItEasy nhận diện nó là 1 file 32bit được complier bằng .NET

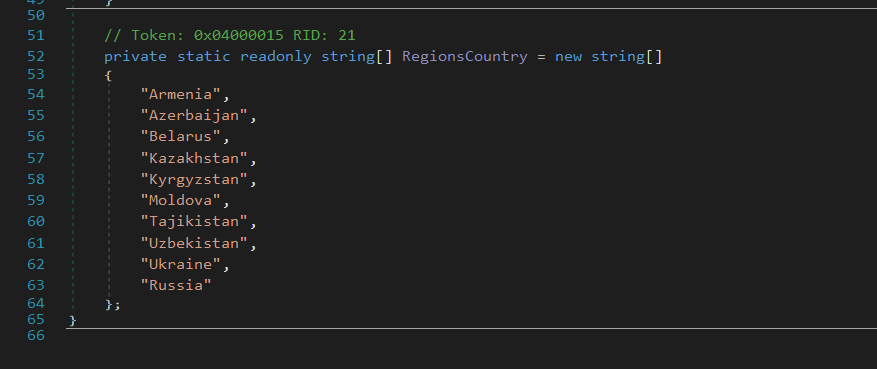




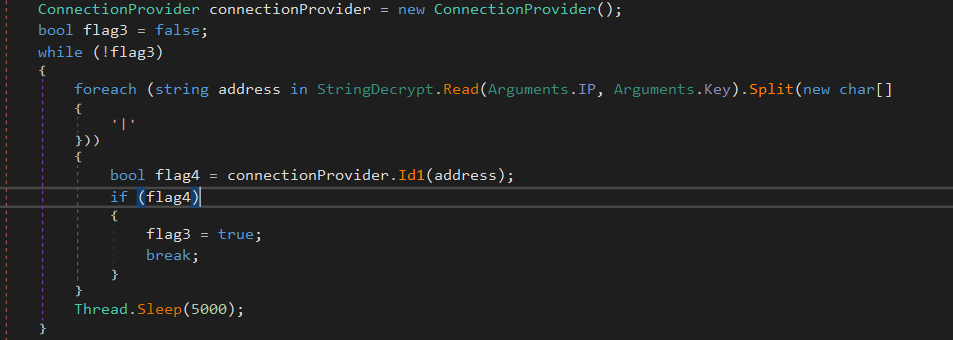
Ở hàm main nó sẽ thực thi hàm Writeline . Nó sẽ có 1 biến flag để kiểm tra hàm check



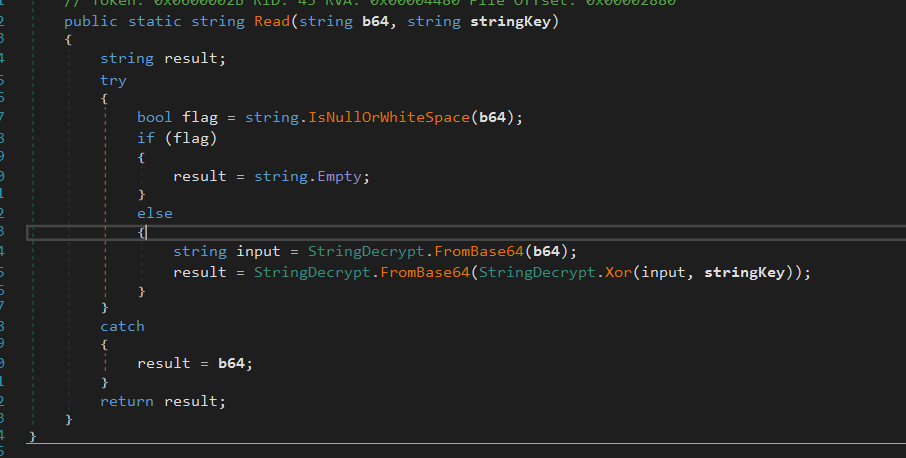
Ở hàm check nó sẽ kiểm tra xem nếu regionsCountry thuộc những vùng country này thì nó sẽ dừng thực thi . Có vẻ như mẫu này là liên quan đến Russia vì ta có thể thấy những country là đồng minh của Russia



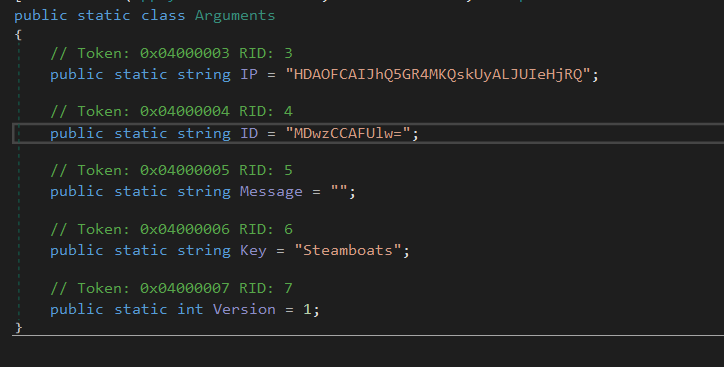
Tiếp theo nó sẽ làm hàm kết nối tới C&C của nó . Đầu tiên nó khởi tạo 1 object connectionProvider sau đó sử dụng hàm Read với parameter là IP với key



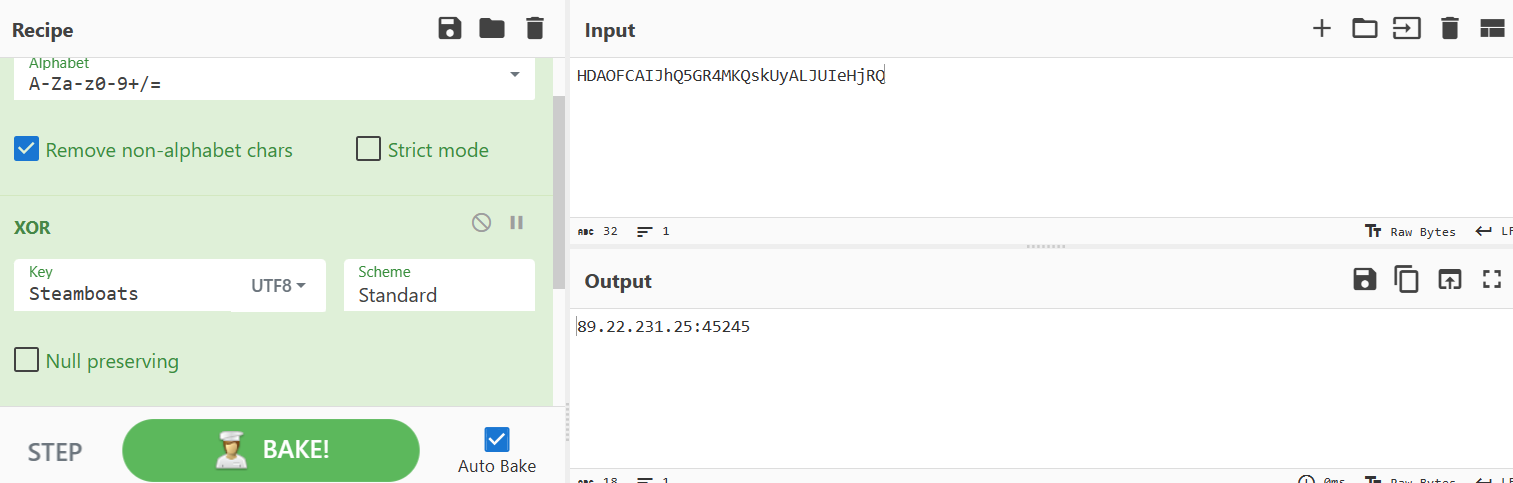
Ở hàm Read nó sử dụng mã hóa base64 với đầu vào và tiếp tục sử dụng key xor để giải mà và tiếp tục sử dụng kết quả đó giải mã base64 tiếp



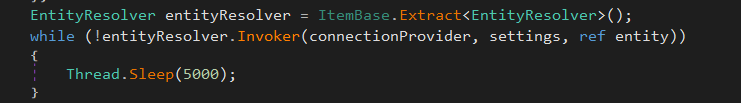
Ở class Argument nó đã khởi tạo sẵn IP và key lẫn ID



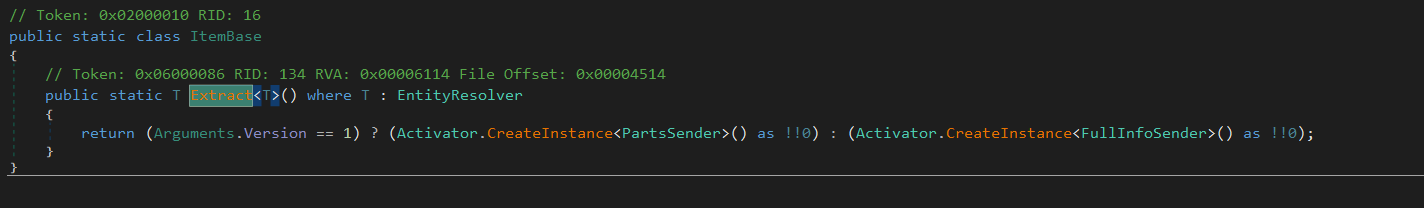
Nó kết nối đến C&C là 89.22.231.25:45245



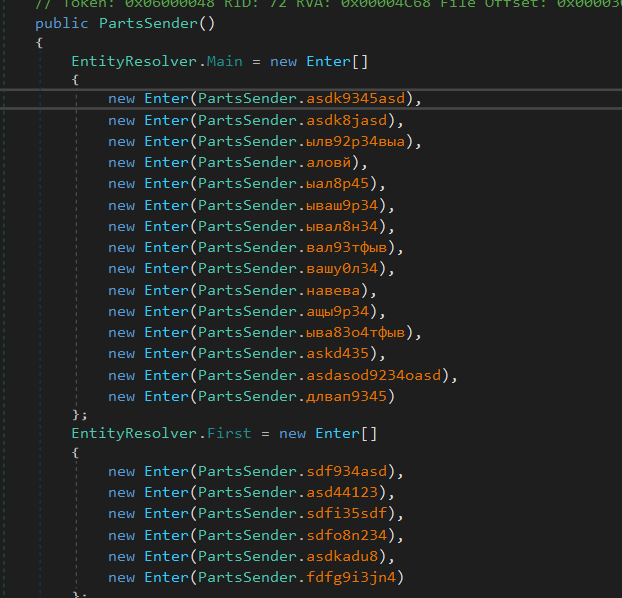
Nó sẽ thiết lập kết nối đến C&C để có thể thực thi các method Tiếp theo nó sẽ khởi tạo 1 đối tượng entityResolver và gọi tới hàm Extract .



Tùy thuộc vào giá trị Version mà nó sẽ thực thi hàm PartsSender hay hàm FullInfoSender

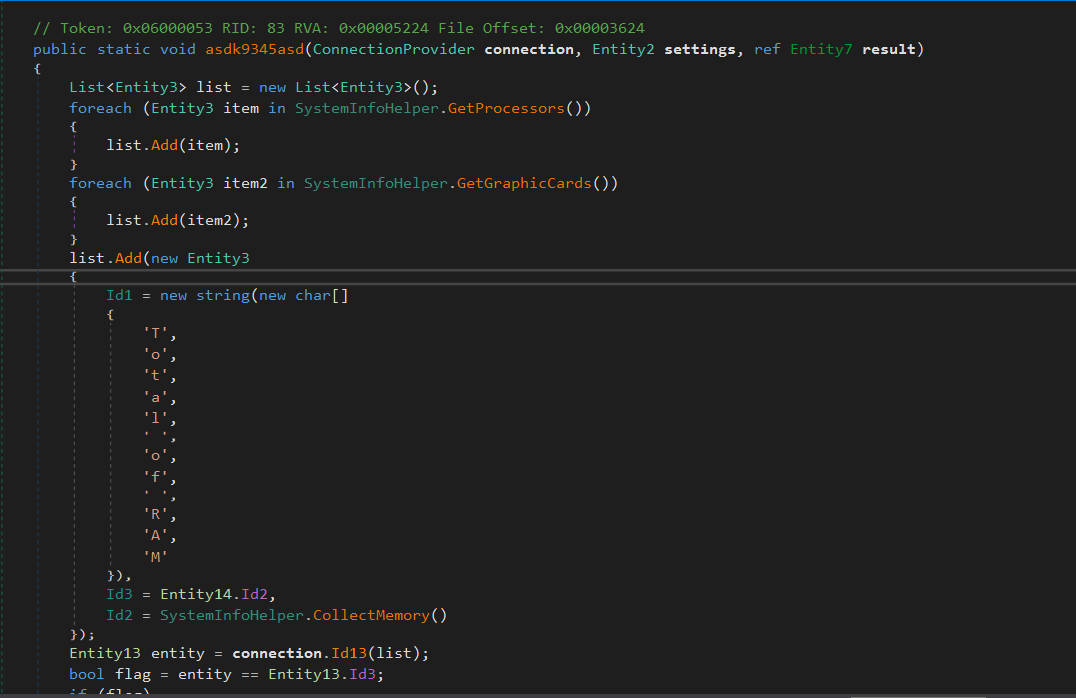


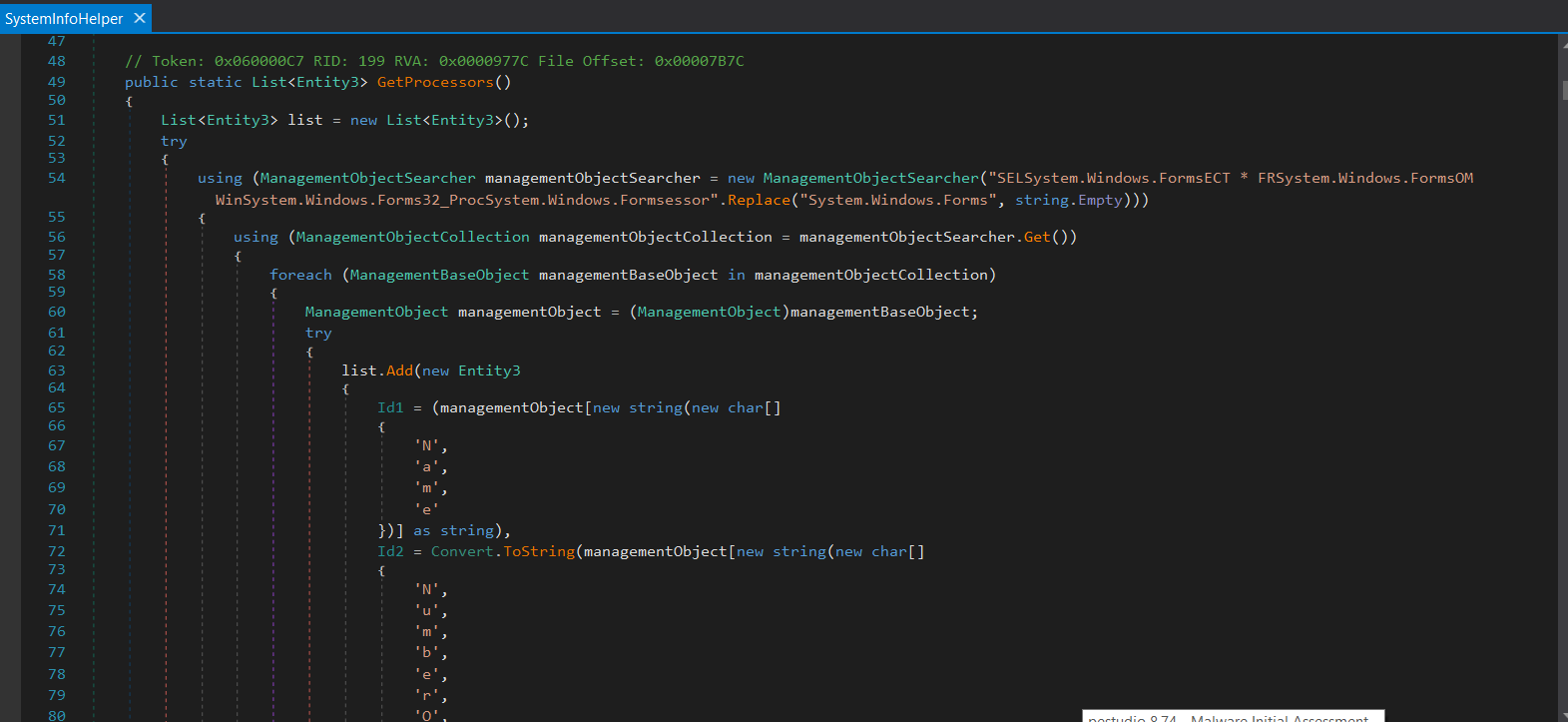
Nó khởi tạo rất nhiều hàm với tên ngẫu nhiên và ứng với mỗi hàm sẽ thực thi cho việc collect data khác nhau . Những data mà nó sẽ collect như IPv4 Address, Domain Name, Windows Version, Virtual Display Size, Country, User Name, Processor Info, Graphics Card Info, RAM Info, Browsers Data, Installed Programs, Running Processes, Available Languages, Telegram data, Discord tokens, Steam configuration, VPN Credentials( OpenVPN , ProtonVPN), Antiviruses, screenshots

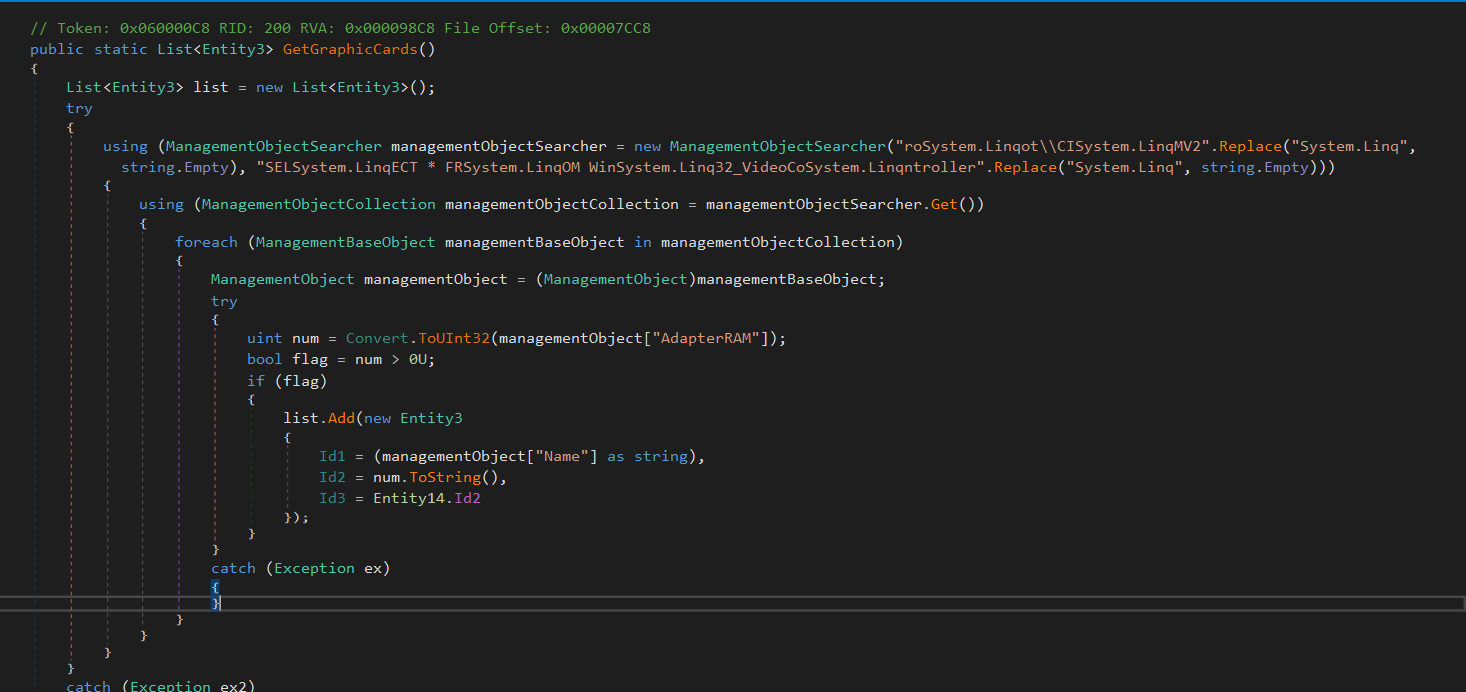


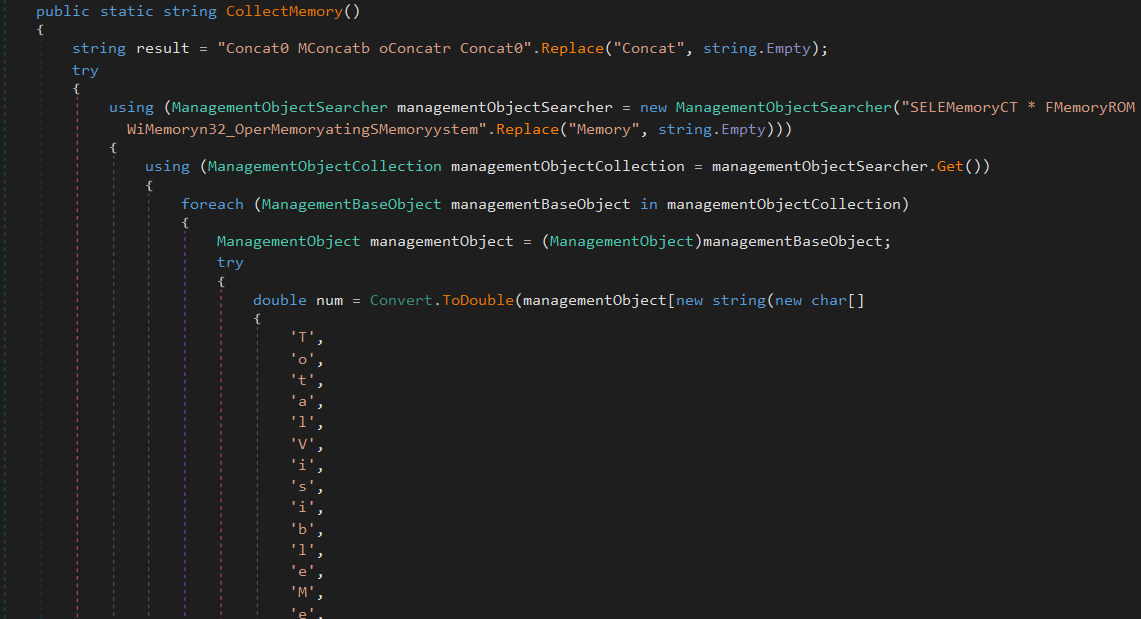
**Collect Systeminfomation**

Ở hàm này đầu tiên này tạo danh sách list chứa thông tin về bộ xử lý và card đồ họa của system . Nó sẽ call tới các hàm GetProcessors và GetGraphicCards , CollectMemory để thu thập dữ liệu



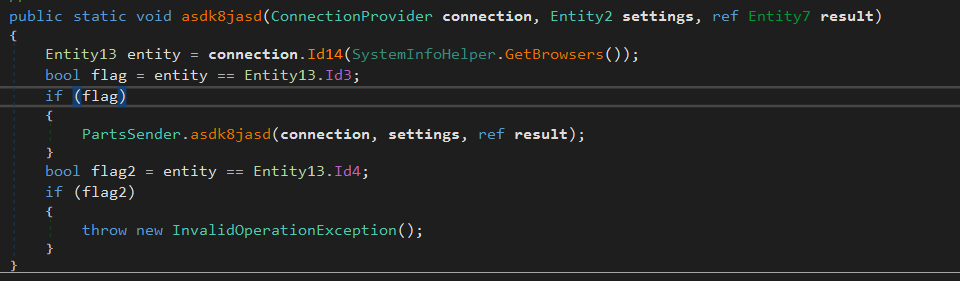


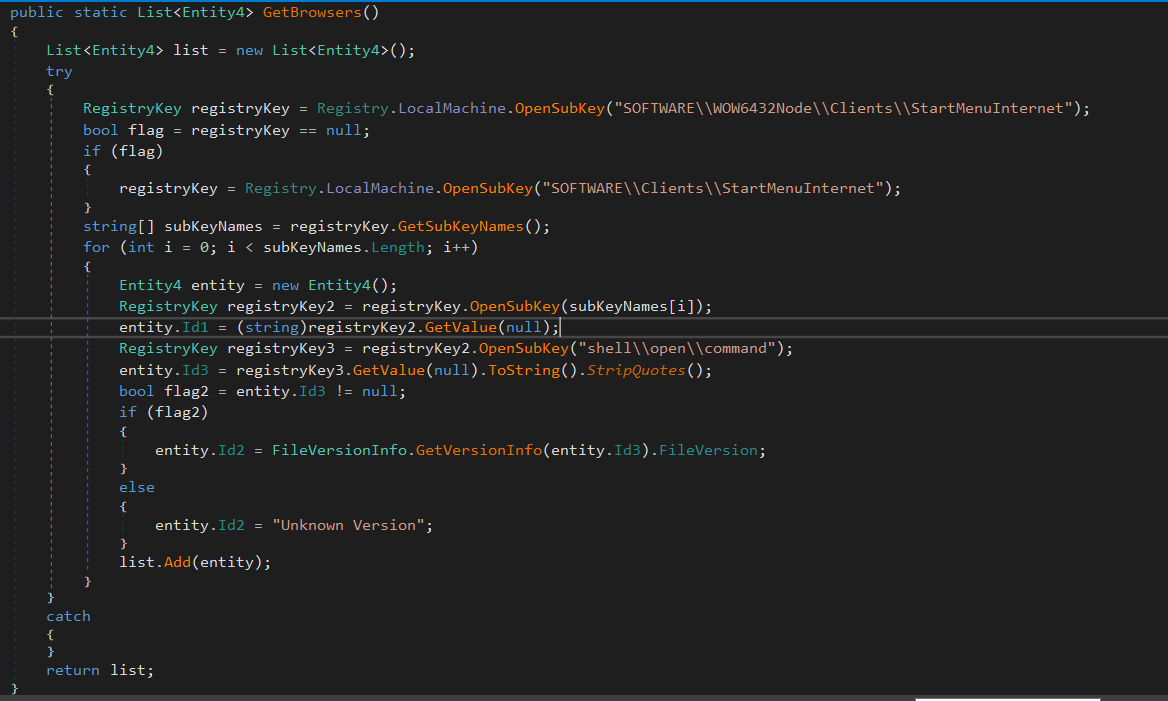




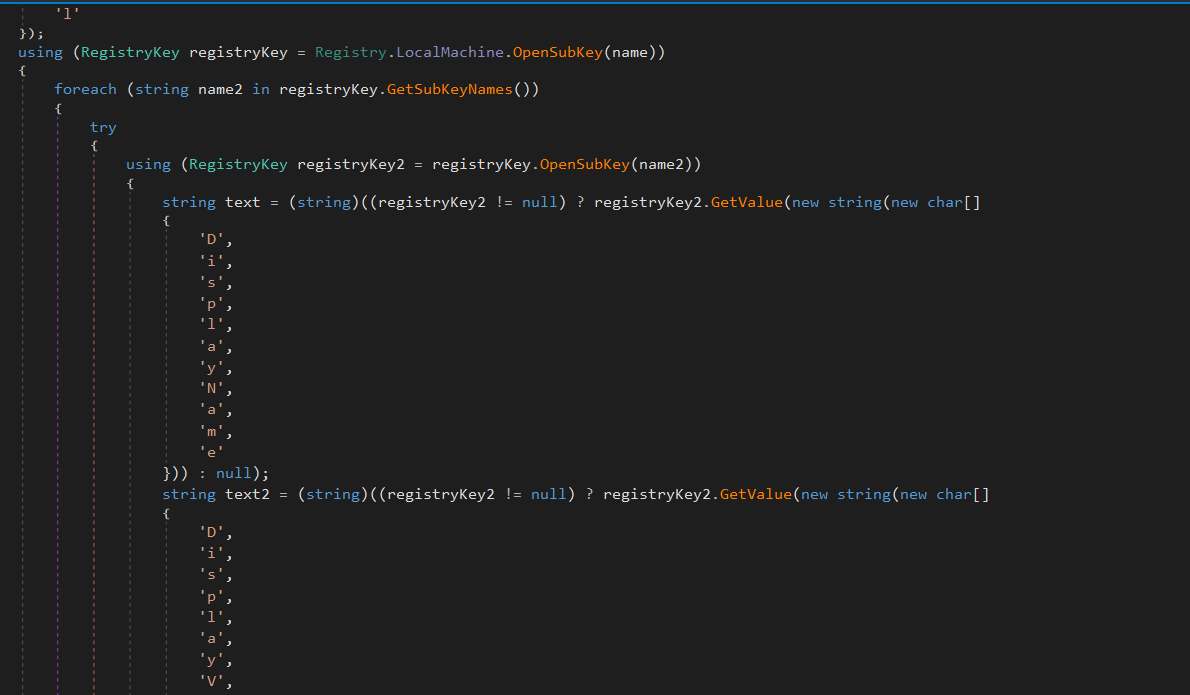
**Collect Broswer**

Nó sẽ thu danh sách các trình duyệt web đã cài đặt trên hệ thống như đường dẫn , phiên bản và tên trình duyệt

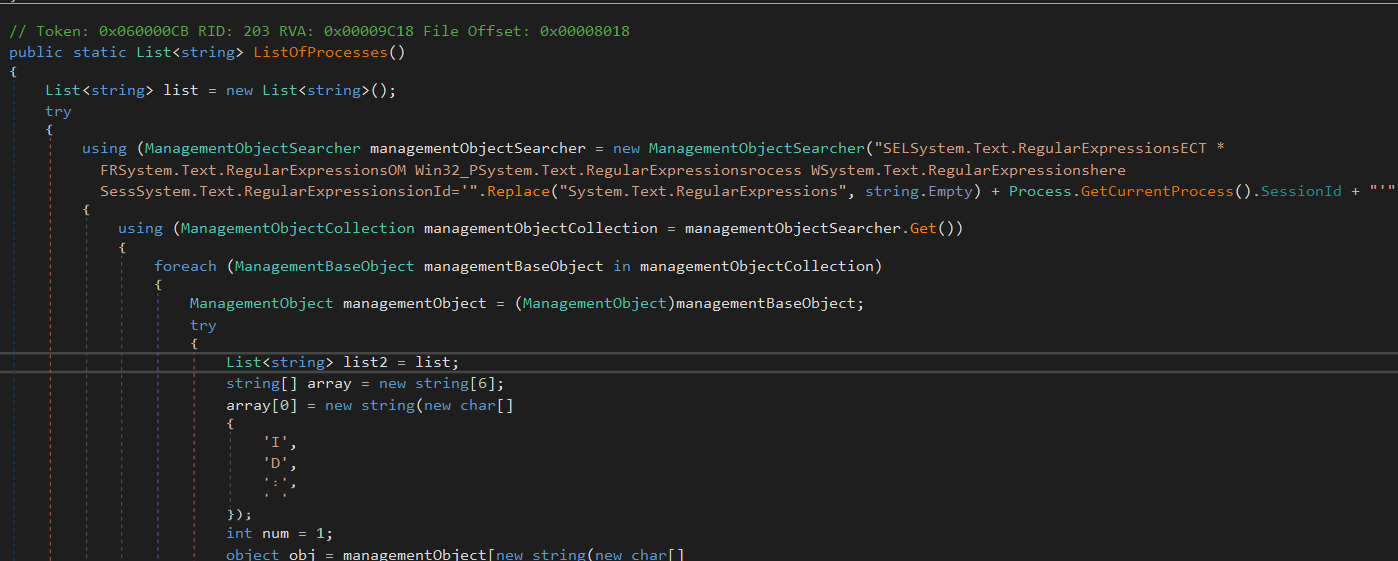




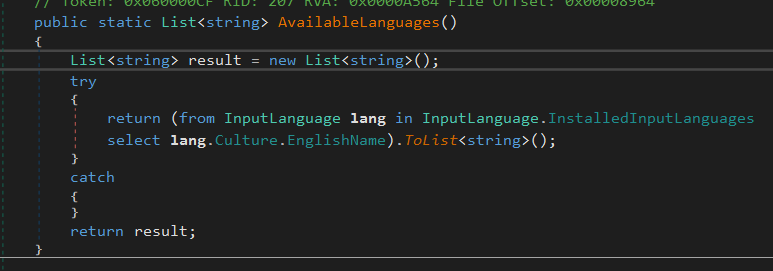
Collect Listprogram



Collect ListProcess



Collect AvailableLanguages



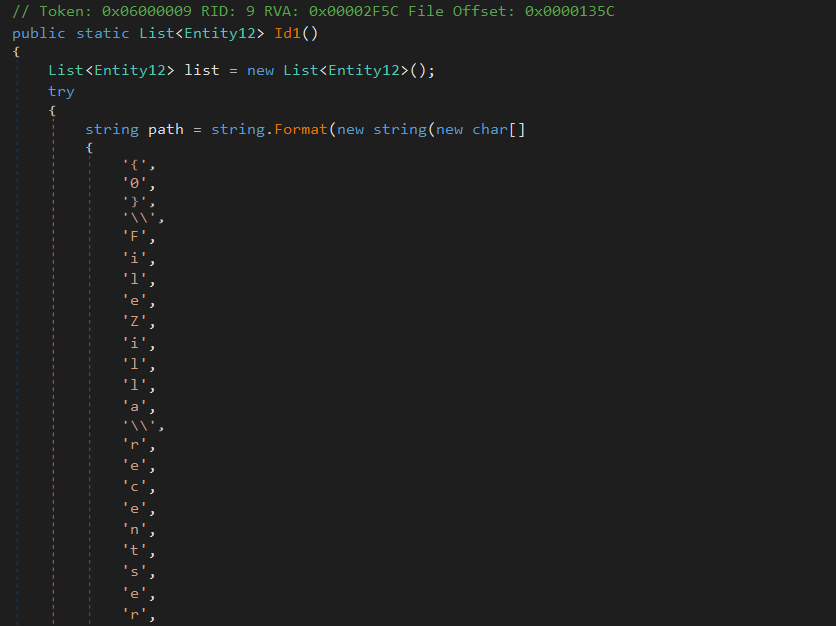
Search File



Collect Infofile from Profile

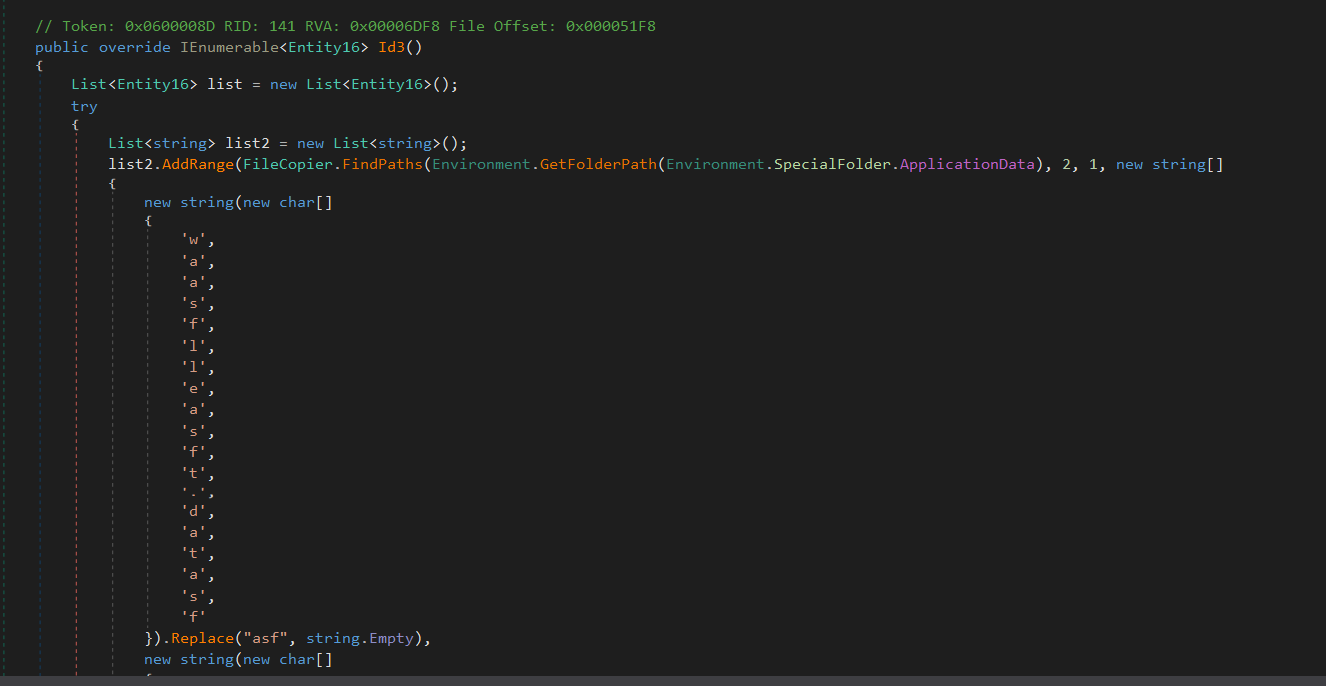


Collect foder ApplicationData

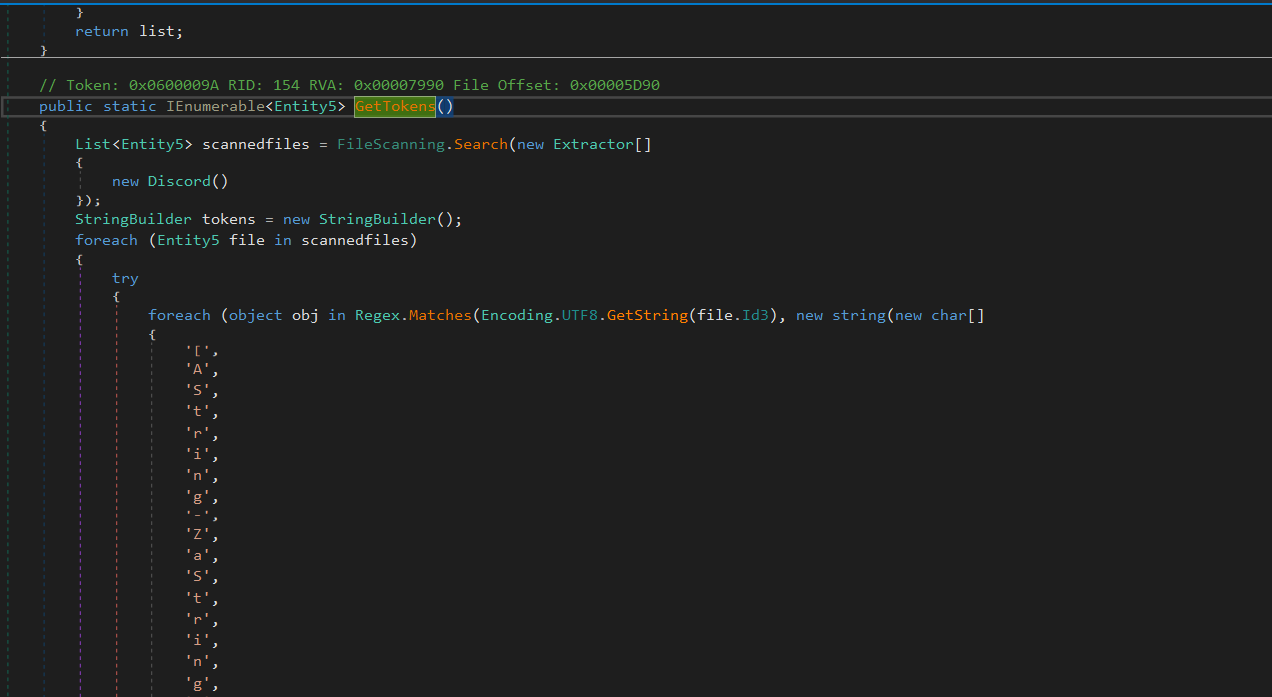


Collect AllWallets

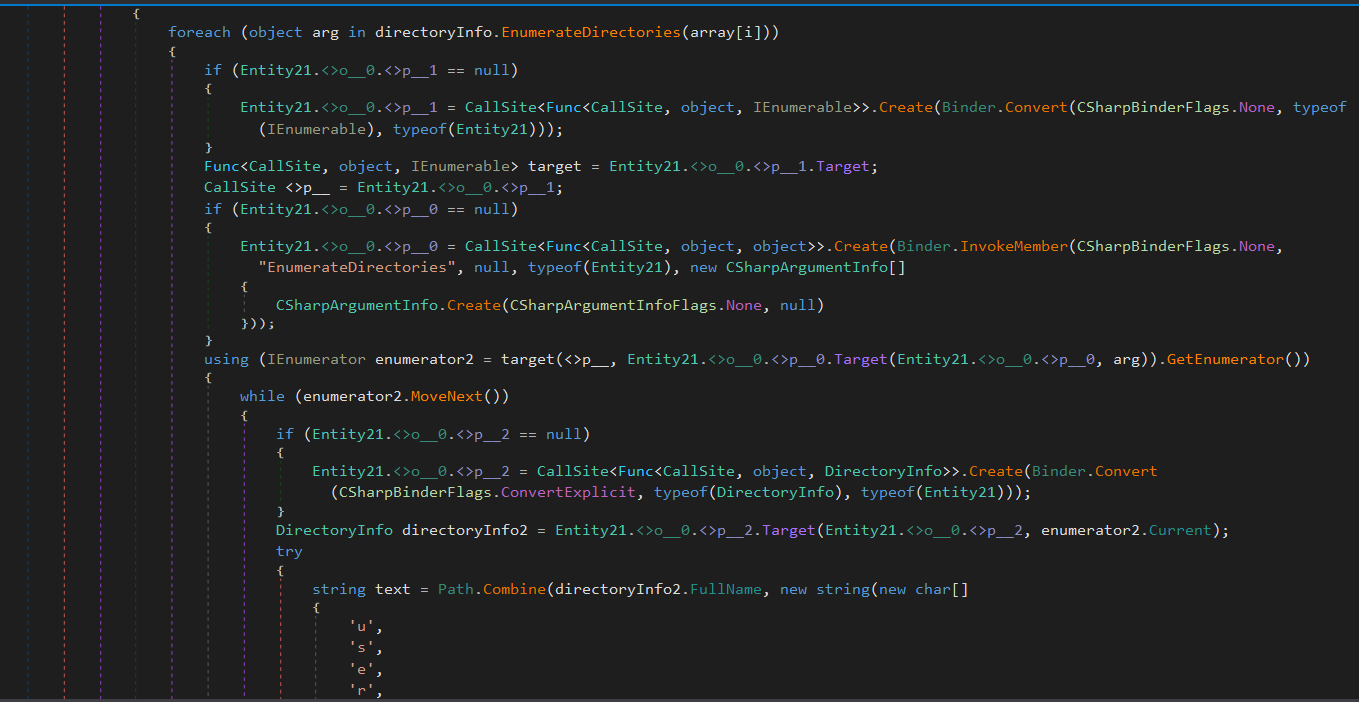




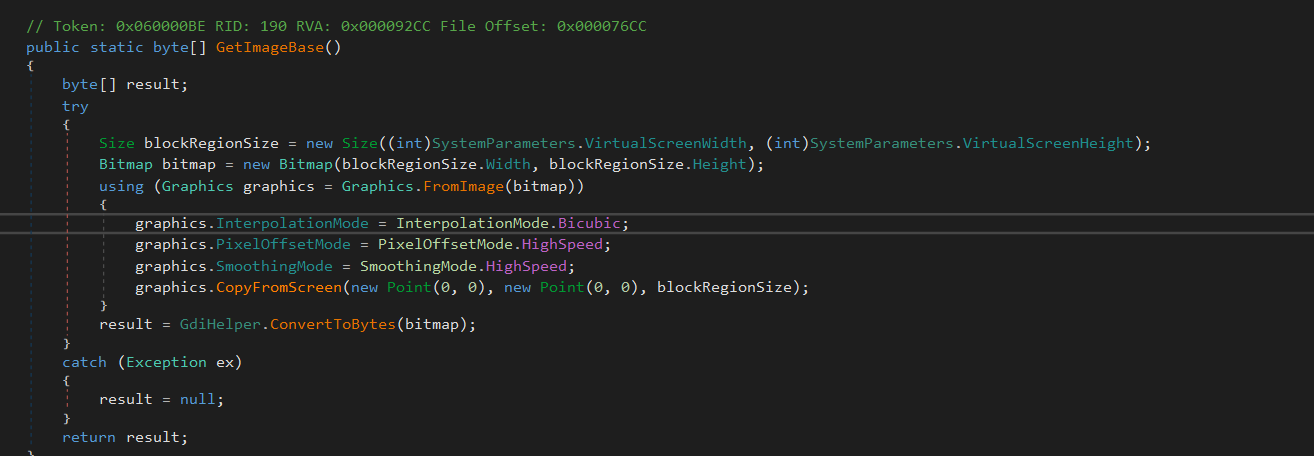
GetToken Discord



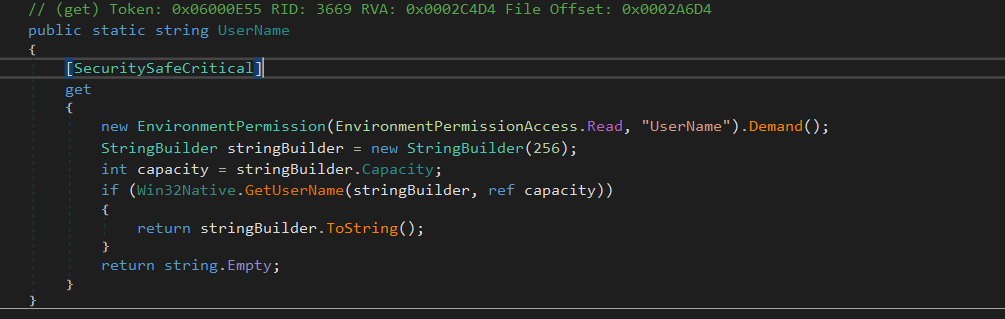
Collect Configdata



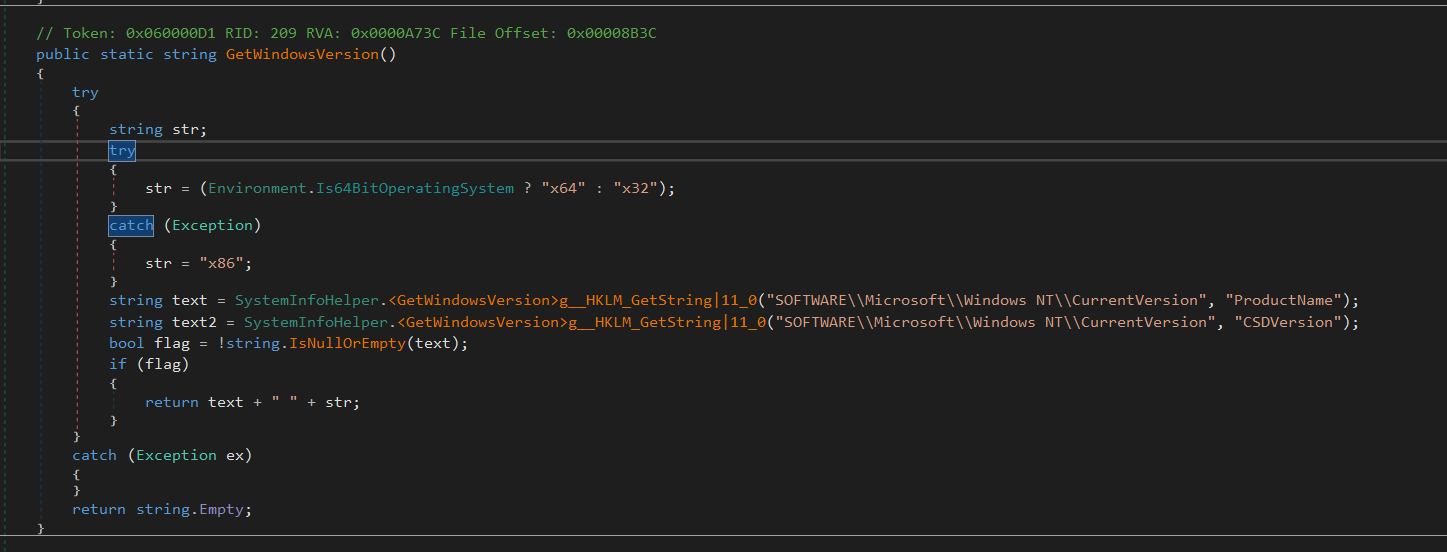
Collect ScreenShot

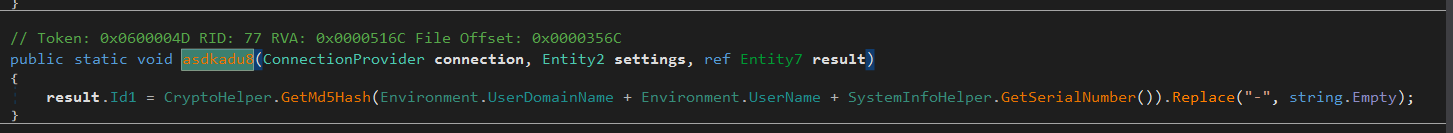


Collect UserName

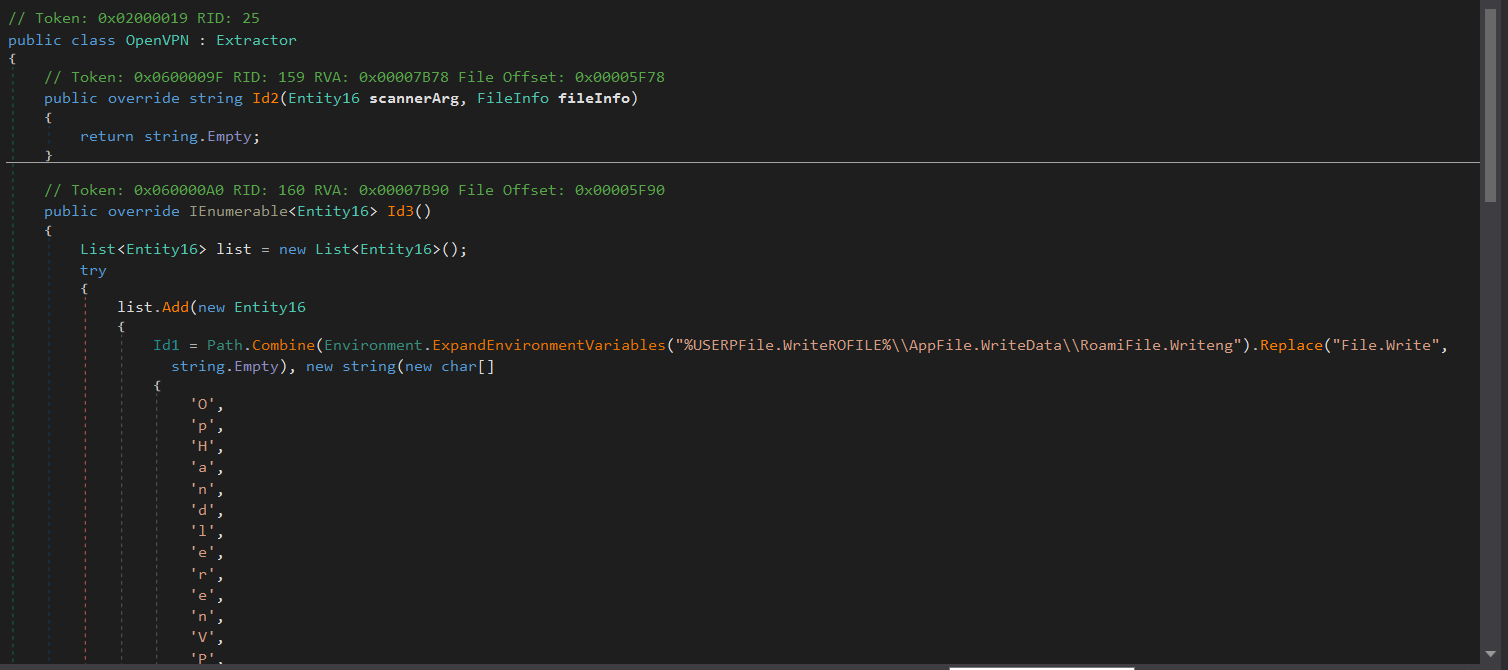


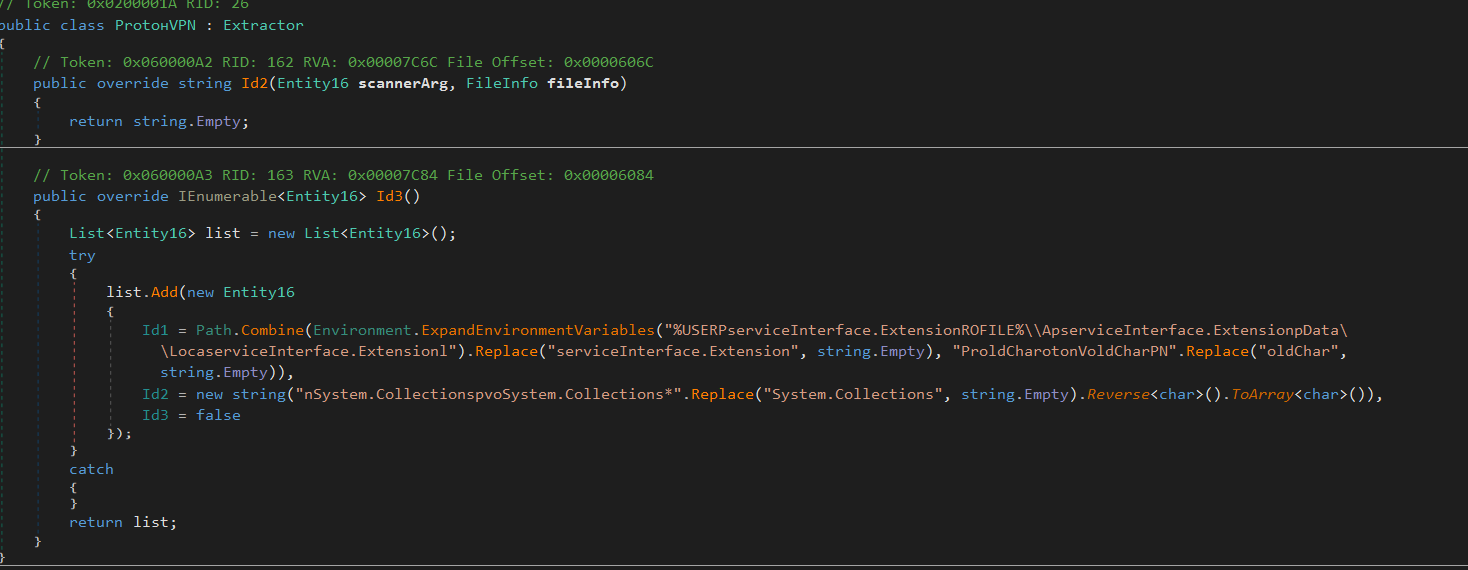
Collect WindowsVersion





Collect DataOpenVPN

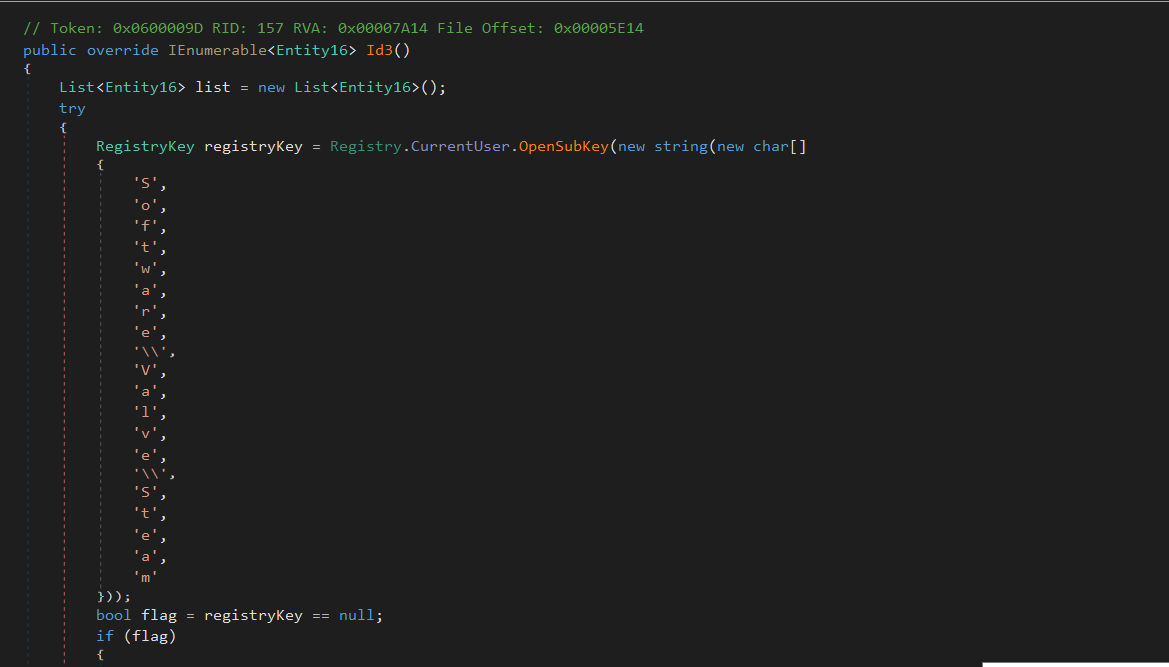




Collect IPv4



Collect dataSteam



**4.IOC**

C&C: 89.22.231.25:45245

Hash :

Filename : RedlineStealer

Md5: fea0d408c87697be73c07b988419dc12

Filename : vbc.exe

Md5: fef46f9b57758768d458f3e42062350e

Directory created : %LocalApplicationData%\Yandex\YaAddon