CSEC 743 – Malware Reverse Engineering

Reversing Project

Name: Praj Sanjay Shete (ps7600@rit.edu)

Malware Sample: Redline Stealer

MD5 Hash: eaa7ddf9a5fe256bc115f2604c8bd754

URL for the sample:
https://bazaar.abuse.ch/sample/8df6ff949de778a20deb98bd90e21d9e9449045b73f75cd62c051957997882bb/

Introduction

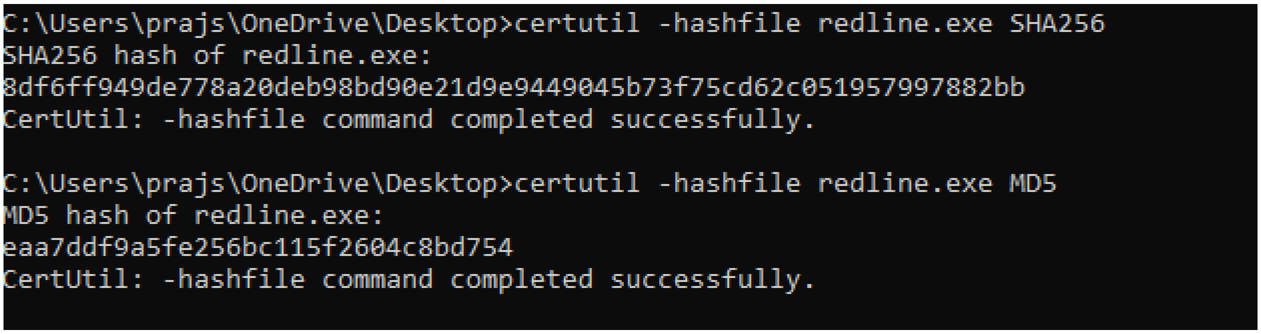
RedLine Stealer, which was originally discovered around March 2020, is a potent data-gathering tool with the ability to steal login credentials from numerous applications and platforms, including web browsers, FTP clients, email apps, Steam, instant messaging clients, and VPNs. But it also gathers chat logs, local files, card numbers stored in browsers, databases for cryptocurrency wallets, and even authentication cookies and card numbers. Additionally, it acquires comprehensive data about the victim's system, including as their IP address, city, and country, as well as their current username, operating system, UAC settings, administrator rights, user-agent, and information about infected PC hardware and graphics cards. Even installed antivirus software can be recognized by it.

Contents:

Basic Static Analysis

File Hash

First I acquired the file hash of the sample to be analyzed, to confirm whether I am analyzing a true positive case. I acquired the file hash using the certutil tool.

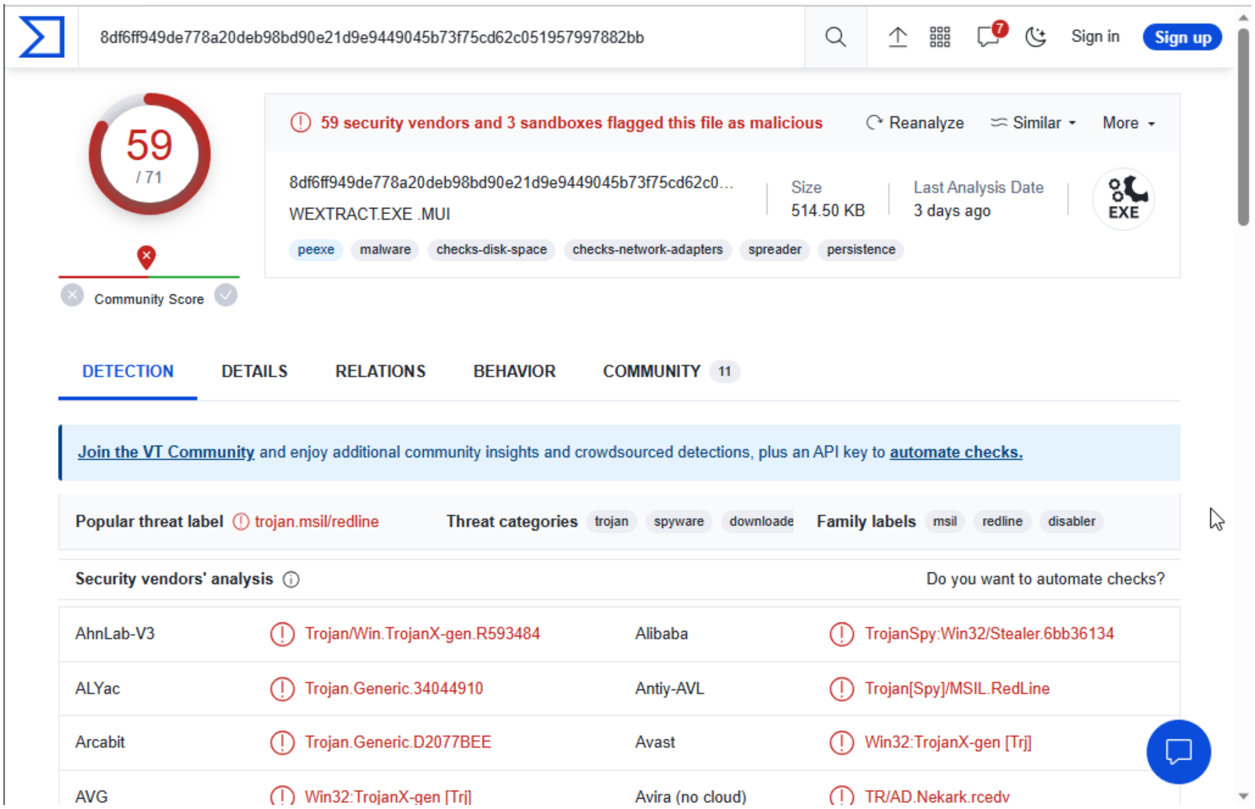


```
C:\Users\prajs\OneDrive\Desktop>certutil -hashfile redline.exe SHA256
SHA256 hash of redline.exe:
8df6ff949de778a20deb98bd90e21d9e9449045b73f75cd62c051957997882bb
CertUtil: -hashfile command completed successfully.

C:\Users\prajs\OneDrive\Desktop>certutil -hashfile redline.exe MD5
MD5 hash of redline.exe:
eaa7ddf9a5fe256bc115f2604c8bd754
CertUtil: -hashfile command completed successfully.
```

VirusTotal

I the uploaded the PE file onto the virus total to cross check the file hash and its malicious nature in the wild. The sample seemed malicious with around 60 hits by different security vendors.



8df6ff949de778a20deb98bd90e21d9e9449045b73f75cd62c051957997882bb

59 / 71

Community Score

59 security vendors and 3 sandboxes flagged this file as malicious

8df6ff949de778a20deb98bd90e21d9e9449045b73f75cd62c0...
WEXTRACT.EXE .MUI

Size 514.50 KB

Last Analysis Date 3 days ago

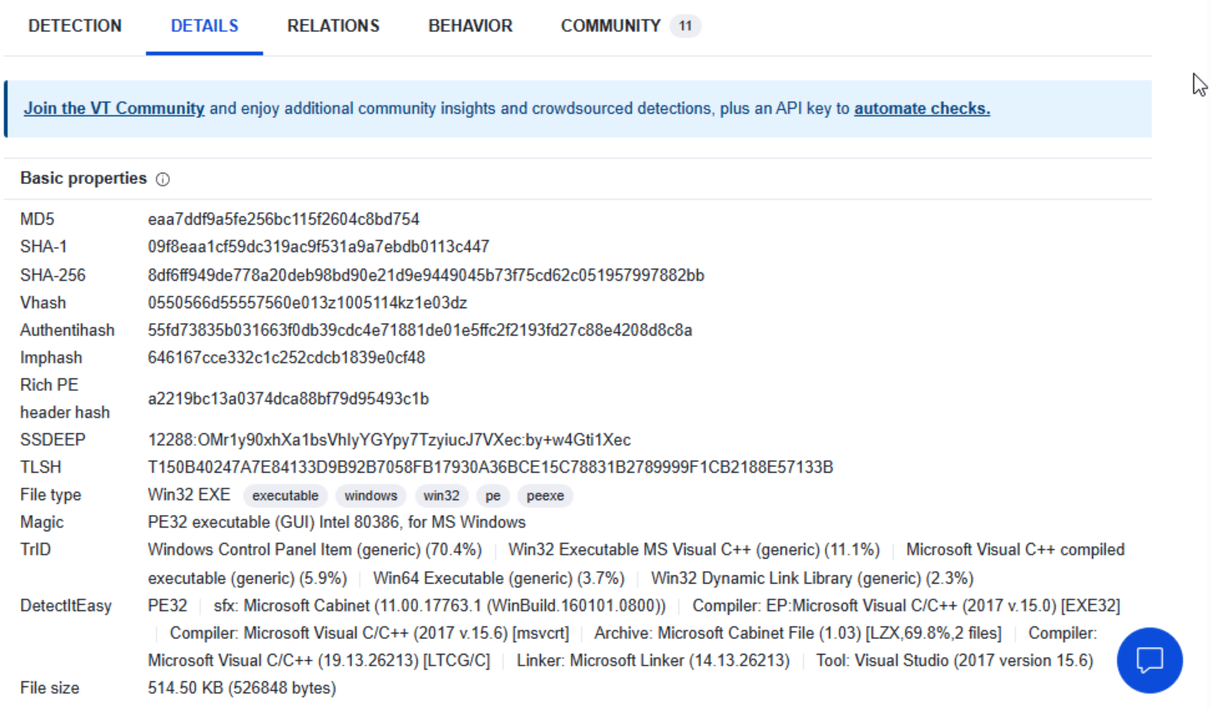
peexe malware checks-disk-space checks-network-adapters spreader persistence

Join the VT Community and enjoy additional community insights and crowdsourced detections, plus an API key to automate checks.

Popular threat label trojan.msil/redline Threat categories trojan spyware downloade Family labels msil redline disabler

Security vendors' analysis Do you want to automate checks?

AhnLab-V3	Trojan/Win.TrojanX-gen.R593484	Alibaba	TrojanSpy:Win32/Stealer.6bb36134
ALYac	Trojan.Generic.34044910	Antiy-AVL	Trojan[Spy]/MSIL.RedLine
Arcabit	Trojan.Generic.D2077BEE	Avast	Win32:TrojanX-gen [Trj]
AVG	Win32:TrojanX-gen [Trj]	Avira (no cloud)	TR/AD.Nekark.rcedv



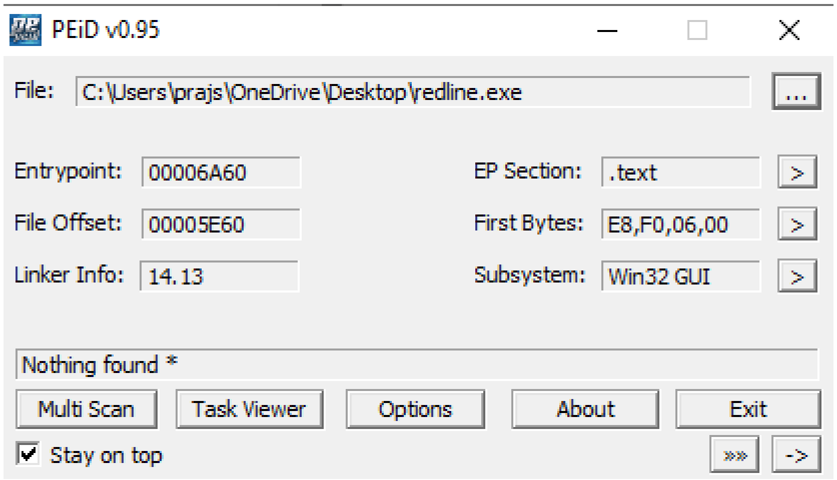
Join the VT Community and enjoy additional community insights and crowdsourced detections, plus an API key to automate checks.

Basic properties ⓘ

MD5	eaa7ddf9a5fe256bc115f2604c8bd754
SHA-1	09f8eaa1cf59dc319ac9f531a9a7ebdb0113c447
SHA-256	8df6ff949de778a20deb98bd90e21d9e9449045b73f75cd62c051957997882bb
Vhash	0550566d55557560e013z1005114kz1e03dz
Authentihash	55fd73835b031663f0db39cdc4e71881de01e5ffc2f2193fd27c88e4208d8c8a
Imphash	646167cce332c1c252cdcb1839e0cf48
Rich PE header hash	a2219bc13a0374dca88bf79d95493c1b
SSDEEP	12288:OMr1y90xhXa1bsVhlyYGYpy7TzyiucJ7VXec:by+w4Gti1Xec
TLSH	T150B40247A7E84133D9B92B7058FB17930A36BCE15C78831B2789999F1CB2188E57133B
File type	Win32 EXE executable windows win32 pe peexe
Magic	PE32 executable (GUI) Intel 80386, for MS Windows
TrID	Windows Control Panel Item (generic) (70.4%) | Win32 Executable MS Visual C++ (generic) (11.1%) | Microsoft Visual C++ compiled executable (generic) (5.9%) | Win64 Executable (generic) (3.7%) | Win32 Dynamic Link Library (generic) (2.3%)
DetectItEasy	PE32 | sfx: Microsoft Cabinet (11.00.17763.1 (WinBuild.160101.0800)) | Compiler: EP:Microsoft Visual C/C++ (2017 v.15.0) [EXE32] | Compiler: Microsoft Visual C/C++ (2017 v.15.6) [msvcrt] | Archive: Microsoft Cabinet File (1.03) [LZX,69.8%,2 files] | Compiler: Microsoft Visual C/C++ (19.13.26213) [LTCG/C] | Linker: Microsoft Linker (14.13.26213) | Tool: Visual Studio (2017 version 15.6)
File size	514.50 KB (526848 bytes)

PeiD

PeiD helped me to determine the packed nature of the binary, as I uploaded the sample it showed no results which can be inferred that binary isn't packed.



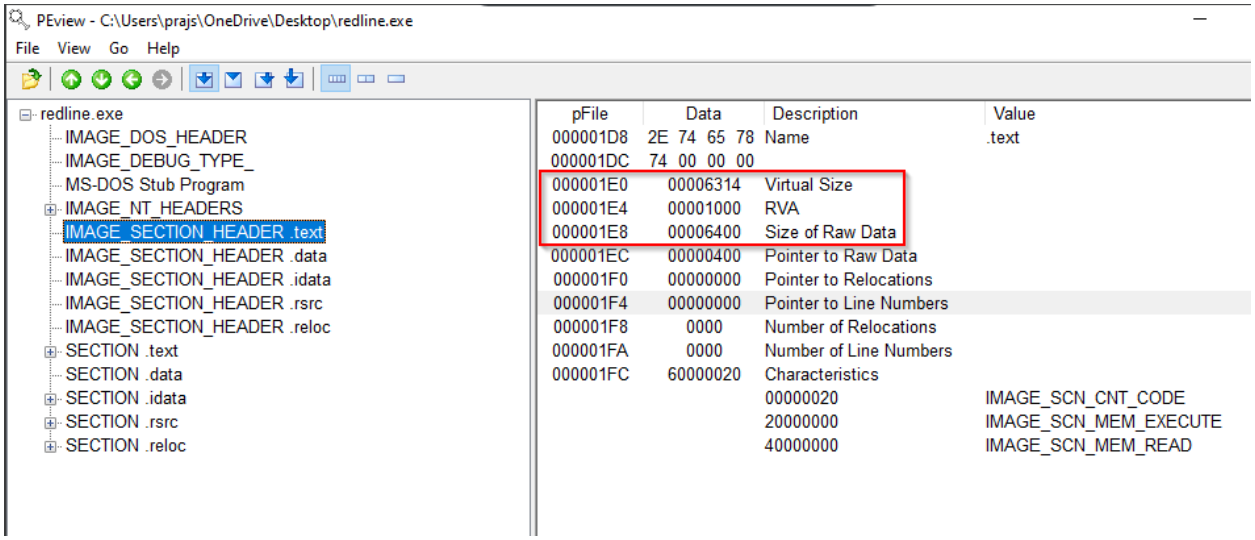
File: C:\Users\prajs\OneDrive\Desktop\redline.exe

Entrypoint:	00006A60	EP Section:	.text
File Offset:	00005E60	First Bytes:	E8,F0,06,00
Linker Info:	14.13	Subsystem:	Win32 GUI

Nothing found *

☑ Stay on top

To confirm this we can use the PEView tool to check and compare the Virtual Size and Size of the Raw data, if both are close enough the binary is not packed.



PEview - C:\Users\prajs\OneDrive\Desktop\redline.exe

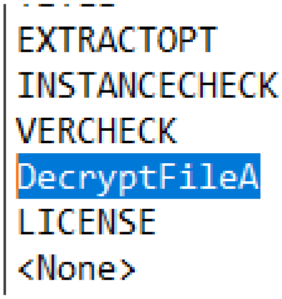
- redline.exe
  - IMAGE_DOS_HEADER
  - IMAGE_DEBUG_TYPE_
  - MS-DOS Stub Program
  - IMAGE_NT_HEADERS
  - IMAGE_SECTION_HEADER .text
  - IMAGE_SECTION_HEADER .data
  - IMAGE_SECTION_HEADER .idata
  - IMAGE_SECTION_HEADER .rsrc
  - IMAGE_SECTION_HEADER .reloc
  - SECTION .text
  - SECTION .data
  - SECTION .idata
  - SECTION .rsrc
  - SECTION .reloc

pFile	Data	Description	Value
000001D8	2E 74 65 78	Name	.text
000001DC	74 00 00 00		
000001E0	00006314	Virtual Size	
000001E4	00001000	RVA	
000001E8	00006400	Size of Raw Data	
000001EC	00000400	Pointer to Raw Data	
000001F0	00000000	Pointer to Relocations	
000001F4	00000000	Pointer to Line Numbers	
000001F8	0000	Number of Relocations	
000001FA	0000	Number of Line Numbers	
000001FC	60000020	Characteristics	
		00000020	IMAGE_SCN_CNT_CODE
		20000000	IMAGE_SCN_MEM_EXECUTE
		40000000	IMAGE_SCN_MEM_READ

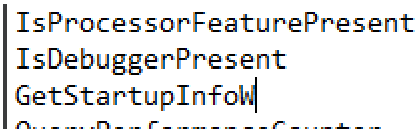
Strings

Moving on the strings part, I used Strings tool to extract strings from the binary. This helped my analysis greatly as strings provide a brief overview about the overall malicious intent of the binary file.

The most important strings I observed were DecryptFileA and IsDebuggerPresent. The two strings gave me a hint that the malware is acting as a Loader at first stage and will load additional payloads as it will decrypt some files as it will execute. IsDebuggerPresent is often used to avoid detection and will generally terminate its execution if returns true.

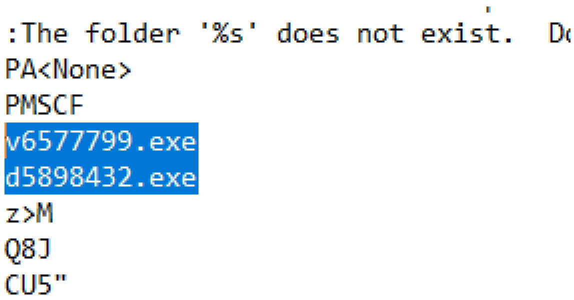


```
EXTRACTOPT
INSTANCECHECK
VERCHECK
DecryptFileA
LICENSE
<None>
```



```
IsProcessorFeaturePresent
IsDebuggerPresent
GetStartupInfoW
```

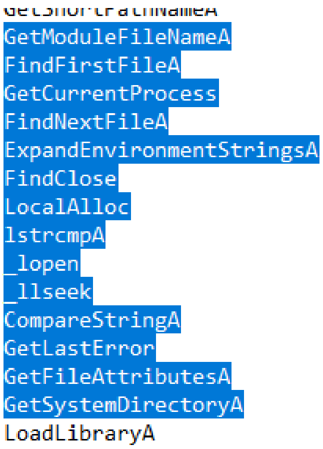
Since my attention got caught on DecryptFileA string, I looked for additional payloads being created, I then searched for any strings with exe files.



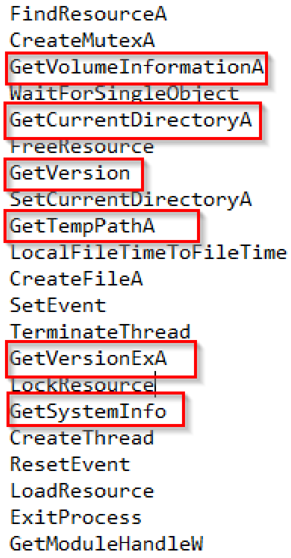
```
:The folder '%s' does not exist.  D
PA<None>
PMSCF
v6577799.exe
d5898432.exe
z>M
Q8J
CU5"
```

I got two strings with 2 exe files which probably can be created/written on runtime since I didn't observe any URLs, domain names, IP addresses the malware can connect to download any files from the Internet. Also there weren't any API calls related to that.

There were some other suspicious API calls too which modified/searched for registry keys, manipulated the filesystems, Process information.



```
GetShortPathNameA
GetModuleFileNameA
FindFirstFileA
GetCurrentProcess
FindNextFileA
ExpandEnvironmentStringsA
FindClose
LocalAlloc
lstrcmpA
_lopen
_llseek
CompareStringA
GetLastError
GetFileAttributesA
GetSystemDirectoryA
LoadLibraryA
```



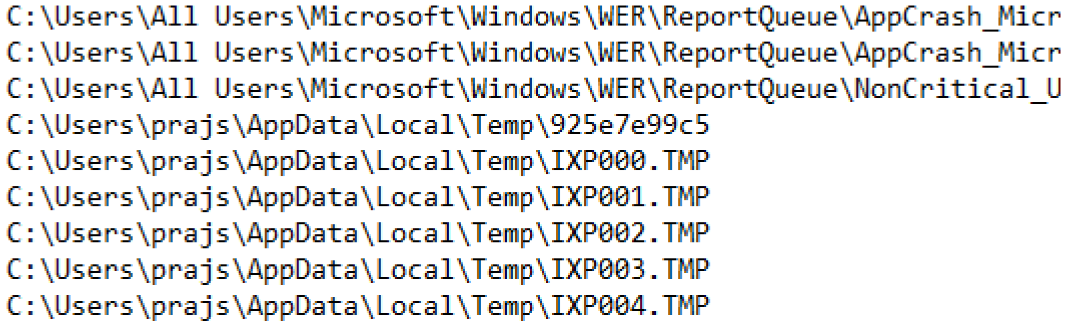
```
FindResourceA
CreateMutexA
GetVolumeInformationA
WaitForSingleObject
GetCurrentDirectoryA
FreeResource
GetVersion
SetCurrentDirectoryA
GetTempPathA
LocalFileTimeToFileTime
CreateFileA
SetEvent
TerminateThread
GetVersionExA
LockResource
GetSystemInfo
CreateThread
ResetEvent
LoadResource
ExitProcess
GetModuleHandleW
```

Basic Dynamic Analysis

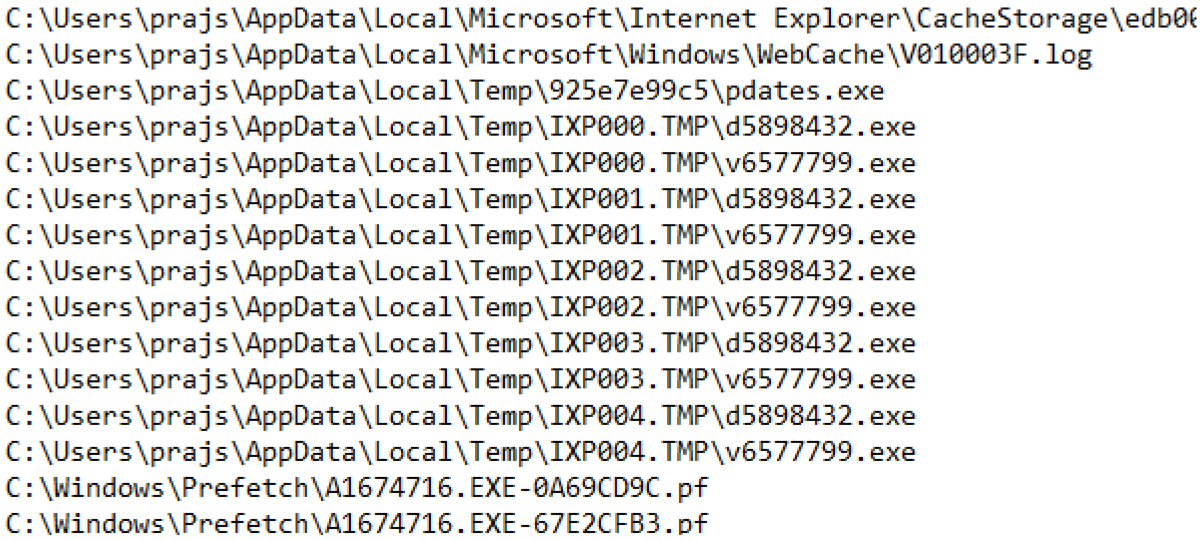
Regshot

I took the first shot before executing the malware and second after executing the malware.

I first looked at the files dropped, since my previous analysis went into that direction



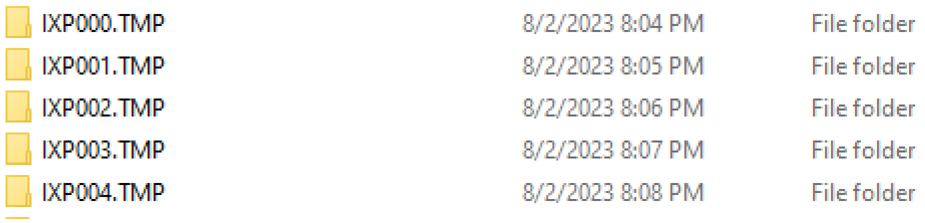
```
C:\Users\All Users\Microsoft\Windows\WER\ReportQueue\AppCrash_Micr
C:\Users\All Users\Microsoft\Windows\WER\ReportQueue\AppCrash_Micr
C:\Users\All Users\Microsoft\Windows\WER\ReportQueue\NonCritical_U
C:\Users\prajs\AppData\Local\Temp\925e7e99c5
C:\Users\prajs\AppData\Local\Temp\IXP000.TMP
C:\Users\prajs\AppData\Local\Temp\IXP001.TMP
C:\Users\prajs\AppData\Local\Temp\IXP002.TMP
C:\Users\prajs\AppData\Local\Temp\IXP003.TMP
C:\Users\prajs\AppData\Local\Temp\IXP004.TMP
```



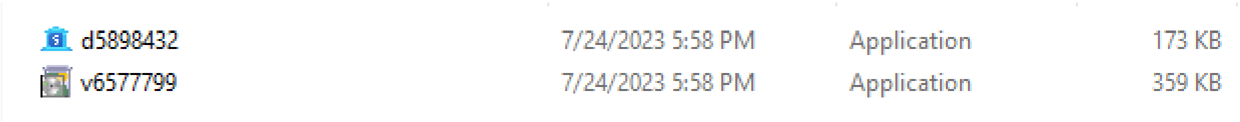
```
C:\Users\prajs\AppData\Local\Microsoft\Internet Explorer\CacheStorage\edb0
C:\Users\prajs\AppData\Local\Microsoft\Windows\WebCache\V010003F.log
C:\Users\prajs\AppData\Local\Temp\925e7e99c5\pdates.exe
C:\Users\prajs\AppData\Local\Temp\IXP000.TMP\d5898432.exe
C:\Users\prajs\AppData\Local\Temp\IXP000.TMP\v6577799.exe
C:\Users\prajs\AppData\Local\Temp\IXP001.TMP\d5898432.exe
C:\Users\prajs\AppData\Local\Temp\IXP001.TMP\v6577799.exe
C:\Users\prajs\AppData\Local\Temp\IXP002.TMP\d5898432.exe
C:\Users\prajs\AppData\Local\Temp\IXP002.TMP\v6577799.exe
C:\Users\prajs\AppData\Local\Temp\IXP003.TMP\d5898432.exe
C:\Users\prajs\AppData\Local\Temp\IXP003.TMP\v6577799.exe
C:\Users\prajs\AppData\Local\Temp\IXP004.TMP\d5898432.exe
C:\Users\prajs\AppData\Local\Temp\IXP004.TMP\v6577799.exe
C:\Windows\Prefetch\A1674716.EXE-0A69CD9C.pf
C:\Windows\Prefetch\A1674716.EXE-67E2CFB3.pf
```

And these were the files created after execution, it created different temp directories, and each directory contained copies of same two files which were seen before in strings.

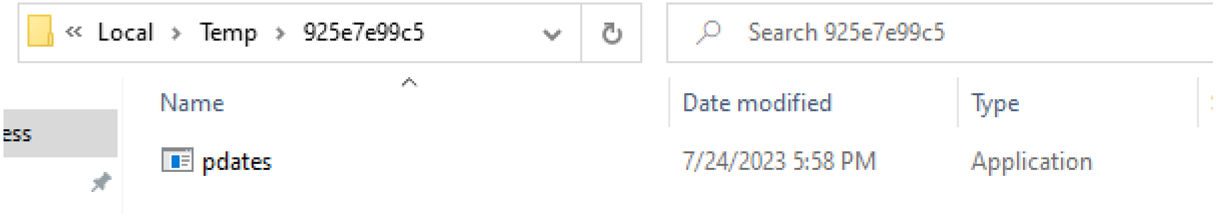
However, I came across another exe file which wasn't there in the strings, **pdates.exe,** I was curious since was this the file that got decrypted? Can it be the actual stealer?



IXP000.TMP	8/2/2023 8:04 PM	File folder
IXP001.TMP	8/2/2023 8:05 PM	File folder
IXP002.TMP	8/2/2023 8:06 PM	File folder
IXP003.TMP	8/2/2023 8:07 PM	File folder
IXP004.TMP	8/2/2023 8:08 PM	File folder

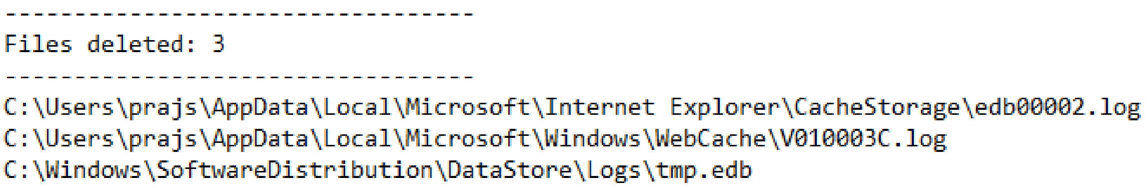


d5898432	7/24/2023 5:58 PM	Application	173 KB
v6577799	7/24/2023 5:58 PM	Application	359 KB



Name	Date modified	Type
pdates	7/24/2023 5:58 PM	Application

There was something else worth noting too, in the files deleted section there were three log files deleted from the system.



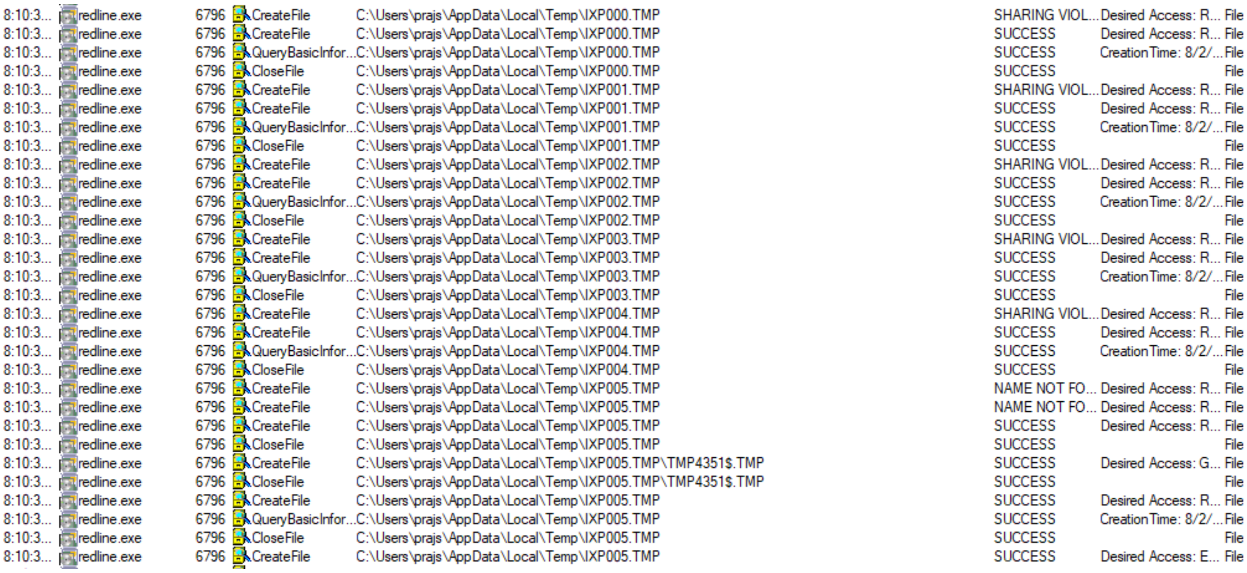
```
----------------------------------
Files deleted: 3
----------------------------------
C:\Users\prajs\AppData\Local\Microsoft\Internet Explorer\CacheStorage\edb00002.log
C:\Users\prajs\AppData\Local\Microsoft\Windows\WebCache\V010003C.log
C:\Windows\SoftwareDistribution\DataStore\Logs\tmp.edb
```

Process Monitor

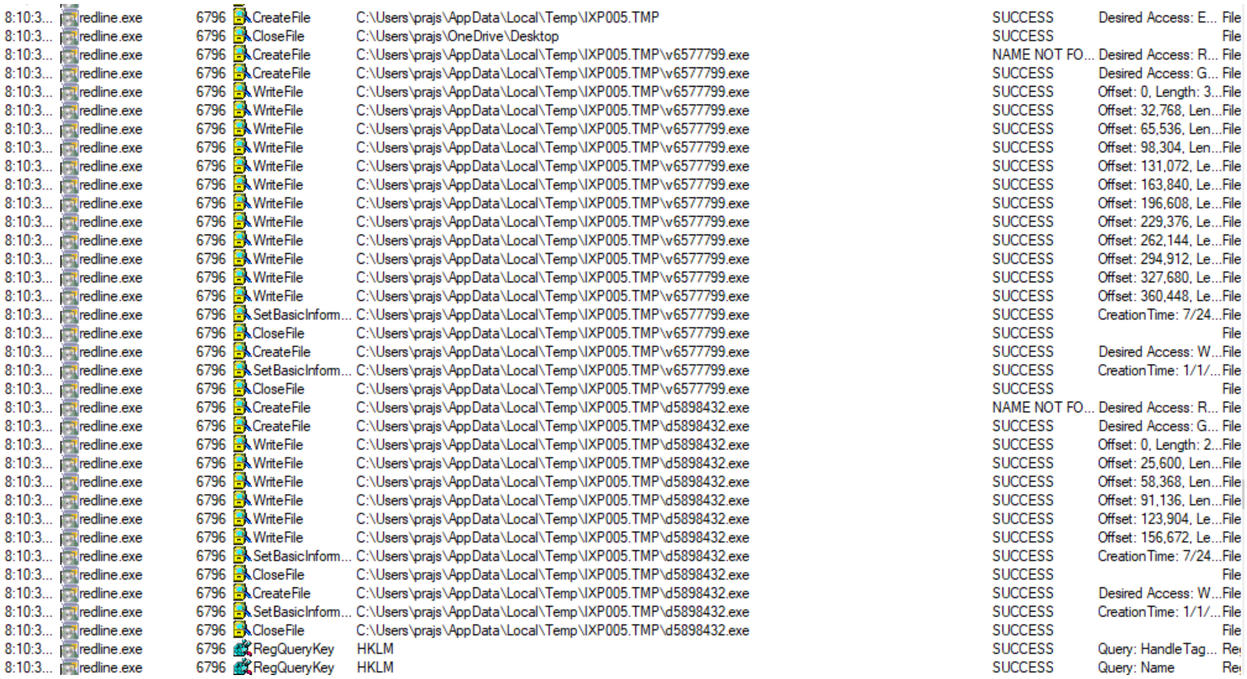
I used Procmon to further investigate how are these files related to each other, how are they created in the file system.

Once again, I executed keeping the filters set to Process name is redline.exe

I got various results related to Create file and Write File

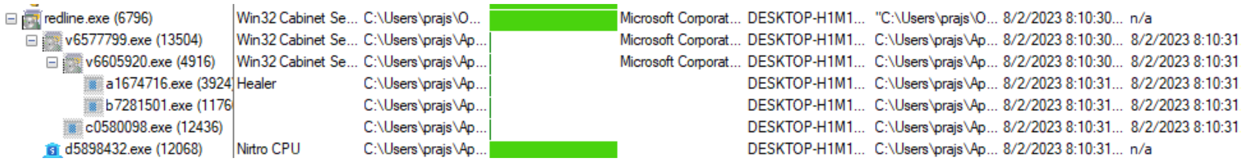


```
8:10:3... redline.exe   6796 CreateFile          C:\Users\prajs\AppData\Local\Temp\IXP000.TMP                   SHARING VIOL... Desired Access: R... File
8:10:3... redline.exe   6796 CreateFile          C:\Users\prajs\AppData\Local\Temp\IXP000.TMP                   SUCCESS         Desired Access: R... File
8:10:3... redline.exe   6796 QueryBasicInfor...  C:\Users\prajs\AppData\Local\Temp\IXP000.TMP                   SUCCESS         CreationTime: 8/2/...File
8:10:3... redline.exe   6796 CloseFile           C:\Users\prajs\AppData\Local\Temp\IXP000.TMP                   SUCCESS                              File
8:10:3... redline.exe   6796 CreateFile          C:\Users\prajs\AppData\Local\Temp\IXP001.TMP                   SHARING VIOL... Desired Access: R... File
8:10:3... redline.exe   6796 CreateFile          C:\Users\prajs\AppData\Local\Temp\IXP001.TMP                   SUCCESS         Desired Access: R... File
8:10:3... redline.exe   6796 QueryBasicInfor...  C:\Users\prajs\AppData\Local\Temp\IXP001.TMP                   SUCCESS         CreationTime: 8/2/...File
8:10:3... redline.exe   6796 CloseFile           C:\Users\prajs\AppData\Local\Temp\IXP001.TMP                   SUCCESS                              File
8:10:3... redline.exe   6796 CreateFile          C:\Users\prajs\AppData\Local\Temp\IXP002.TMP                   SHARING VIOL... Desired Access: R... File
8:10:3... redline.exe   6796 CreateFile          C:\Users\prajs\AppData\Local\Temp\IXP002.TMP                   SUCCESS         Desired Access: R... File
8:10:3... redline.exe   6796 QueryBasicInfor...  C:\Users\prajs\AppData\Local\Temp\IXP002.TMP                   SUCCESS         CreationTime: 8/2/...File
8:10:3... redline.exe   6796 CloseFile           C:\Users\prajs\AppData\Local\Temp\IXP002.TMP                   SUCCESS                              File
8:10:3... redline.exe   6796 CreateFile          C:\Users\prajs\AppData\Local\Temp\IXP003.TMP                   SHARING VIOL... Desired Access: R... File
8:10:3... redline.exe   6796 CreateFile          C:\Users\prajs\AppData\Local\Temp\IXP003.TMP                   SUCCESS         Desired Access: R... File
8:10:3... redline.exe   6796 QueryBasicInfor...  C:\Users\prajs\AppData\Local\Temp\IXP003.TMP                   SUCCESS         CreationTime: 8/2/...File
8:10:3... redline.exe   6796 CloseFile           C:\Users\prajs\AppData\Local\Temp\IXP003.TMP                   SUCCESS                              File
8:10:3... redline.exe   6796 CreateFile          C:\Users\prajs\AppData\Local\Temp\IXP004.TMP                   SHARING VIOL... Desired Access: R... File
8:10:3... redline.exe   6796 CreateFile          C:\Users\prajs\AppData\Local\Temp\IXP004.TMP                   SUCCESS         Desired Access: R... File
8:10:3... redline.exe   6796 QueryBasicInfor...  C:\Users\prajs\AppData\Local\Temp\IXP004.TMP                   SUCCESS         CreationTime: 8/2/...File
8:10:3... redline.exe   6796 CloseFile           C:\Users\prajs\AppData\Local\Temp\IXP004.TMP                   SUCCESS                              File
8:10:3... redline.exe   6796 CreateFile          C:\Users\prajs\AppData\Local\Temp\IXP005.TMP                   NAME NOT FO...  Desired Access: R... File
8:10:3... redline.exe   6796 CreateFile          C:\Users\prajs\AppData\Local\Temp\IXP005.TMP                   NAME NOT FO...  Desired Access: R... File
8:10:3... redline.exe   6796 CreateFile          C:\Users\prajs\AppData\Local\Temp\IXP005.TMP                   SUCCESS         Desired Access: R... File
8:10:3... redline.exe   6796 CloseFile           C:\Users\prajs\AppData\Local\Temp\IXP005.TMP                   SUCCESS                              File
8:10:3... redline.exe   6796 CreateFile          C:\Users\prajs\AppData\Local\Temp\IXP005.TMP\TMP4351$.TMP      SUCCESS         Desired Access: G... File
8:10:3... redline.exe   6796 CloseFile           C:\Users\prajs\AppData\Local\Temp\IXP005.TMP\TMP4351$.TMP      SUCCESS                              File
8:10:3... redline.exe   6796 CreateFile          C:\Users\prajs\AppData\Local\Temp\IXP005.TMP                   SUCCESS         Desired Access: R... File
8:10:3... redline.exe   6796 QueryBasicInfor...  C:\Users\prajs\AppData\Local\Temp\IXP005.TMP                   SUCCESS         CreationTime: 8/2/...File
8:10:3... redline.exe   6796 CloseFile           C:\Users\prajs\AppData\Local\Temp\IXP005.TMP                   SUCCESS                              File
8:10:3... redline.exe   6796 CreateFile          C:\Users\prajs\AppData\Local\Temp\IXP005.TMP                   SUCCESS         Desired Access: E... File
```



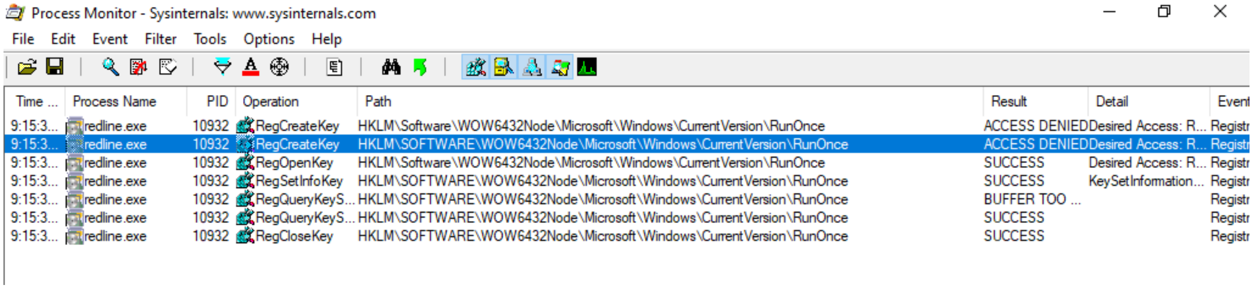
```
8:10:3... redline.exe   6796 CreateFile          C:\Users\prajs\AppData\Local\Temp\IXP005.TMP                   SUCCESS         Desired Access: E... File
8:10:3... redline.exe   6796 CloseFile           C:\Users\prajs\OneDrive\Desktop                                SUCCESS                              File
8:10:3... redline.exe   6796 CreateFile          C:\Users\prajs\AppData\Local\Temp\IXP005.TMP\v6577799.exe      NAME NOT FO...  Desired Access: R... File
8:10:3... redline.exe   6796 CreateFile          C:\Users\prajs\AppData\Local\Temp\IXP005.TMP\v6577799.exe      SUCCESS         Desired Access: G... File
8:10:3... redline.exe   6796 WriteFile           C:\Users\prajs\AppData\Local\Temp\IXP005.TMP\v6577799.exe      SUCCESS         Offset: 0, Length: 3...File
8:10:3... redline.exe   6796 WriteFile           C:\Users\prajs\AppData\Local\Temp\IXP005.TMP\v6577799.exe      SUCCESS         Offset: 32,768, Len...File
8:10:3... redline.exe   6796 WriteFile           C:\Users\prajs\AppData\Local\Temp\IXP005.TMP\v6577799.exe      SUCCESS         Offset: 65,536, Len...File
8:10:3... redline.exe   6796 WriteFile           C:\Users\prajs\AppData\Local\Temp\IXP005.TMP\v6577799.exe      SUCCESS         Offset: 98,304, Len...File
8:10:3... redline.exe   6796 WriteFile           C:\Users\prajs\AppData\Local\Temp\IXP005.TMP\v6577799.exe      SUCCESS         Offset: 131,072, Le...File
8:10:3... redline.exe   6796 WriteFile           C:\Users\prajs\AppData\Local\Temp\IXP005.TMP\v6577799.exe      SUCCESS         Offset: 163,840, Le...File
8:10:3... redline.exe   6796 WriteFile           C:\Users\prajs\AppData\Local\Temp\IXP005.TMP\v6577799.exe      SUCCESS         Offset: 196,608, Le...File
8:10:3... redline.exe   6796 WriteFile           C:\Users\prajs\AppData\Local\Temp\IXP005.TMP\v6577799.exe      SUCCESS         Offset: 229,376, Le...File
8:10:3... redline.exe   6796 WriteFile           C:\Users\prajs\AppData\Local\Temp\IXP005.TMP\v6577799.exe      SUCCESS         Offset: 262,144, Le...File
8:10:3... redline.exe   6796 WriteFile           C:\Users\prajs\AppData\Local\Temp\IXP005.TMP\v6577799.exe      SUCCESS         Offset: 294,912, Le...File
8:10:3... redline.exe   6796 WriteFile           C:\Users\prajs\AppData\Local\Temp\IXP005.TMP\v6577799.exe      SUCCESS         Offset: 327,680, Le...File
8:10:3... redline.exe   6796 WriteFile           C:\Users\prajs\AppData\Local\Temp\IXP005.TMP\v6577799.exe      SUCCESS         Offset: 360,448, Le...File
8:10:3... redline.exe   6796 SetBasicInform...   C:\Users\prajs\AppData\Local\Temp\IXP005.TMP\v6577799.exe      SUCCESS         CreationTime: 7/24...File
8:10:3... redline.exe   6796 CloseFile           C:\Users\prajs\AppData\Local\Temp\IXP005.TMP\v6577799.exe      SUCCESS                              File
8:10:3... redline.exe   6796 CreateFile          C:\Users\prajs\AppData\Local\Temp\IXP005.TMP\v6577799.exe      SUCCESS         Desired Access: W...File
8:10:3... redline.exe   6796 SetBasicInform...   C:\Users\prajs\AppData\Local\Temp\IXP005.TMP\v6577799.exe      SUCCESS         CreationTime: 1/1/...File
8:10:3... redline.exe   6796 CloseFile           C:\Users\prajs\AppData\Local\Temp\IXP005.TMP\v6577799.exe      SUCCESS                              File
8:10:3... redline.exe   6796 CreateFile          C:\Users\prajs\AppData\Local\Temp\IXP005.TMP\d5898432.exe      NAME NOT FO...  Desired Access: R... File
8:10:3... redline.exe   6796 CreateFile          C:\Users\prajs\AppData\Local\Temp\IXP005.TMP\d5898432.exe      SUCCESS         Desired Access: G... File
8:10:3... redline.exe   6796 WriteFile           C:\Users\prajs\AppData\Local\Temp\IXP005.TMP\d5898432.exe      SUCCESS         Offset: 0, Length: 2...File
8:10:3... redline.exe   6796 WriteFile           C:\Users\prajs\AppData\Local\Temp\IXP005.TMP\d5898432.exe      SUCCESS         Offset: 25,600, Len...File
8:10:3... redline.exe   6796 WriteFile           C:\Users\prajs\AppData\Local\Temp\IXP005.TMP\d5898432.exe      SUCCESS         Offset: 58,368, Len...File
8:10:3... redline.exe   6796 WriteFile           C:\Users\prajs\AppData\Local\Temp\IXP005.TMP\d5898432.exe      SUCCESS         Offset: 91,136, Len...File
8:10:3... redline.exe   6796 WriteFile           C:\Users\prajs\AppData\Local\Temp\IXP005.TMP\d5898432.exe      SUCCESS         Offset: 123,904, Le...File
8:10:3... redline.exe   6796 WriteFile           C:\Users\prajs\AppData\Local\Temp\IXP005.TMP\d5898432.exe      SUCCESS         Offset: 156,672, Le...File
8:10:3... redline.exe   6796 SetBasicInform...   C:\Users\prajs\AppData\Local\Temp\IXP005.TMP\d5898432.exe      SUCCESS         CreationTime: 7/24...File
8:10:3... redline.exe   6796 CloseFile           C:\Users\prajs\AppData\Local\Temp\IXP005.TMP\d5898432.exe      SUCCESS                              File
8:10:3... redline.exe   6796 CreateFile          C:\Users\prajs\AppData\Local\Temp\IXP005.TMP\d5898432.exe      SUCCESS         Desired Access: W...File
8:10:3... redline.exe   6796 SetBasicInform...   C:\Users\prajs\AppData\Local\Temp\IXP005.TMP\d5898432.exe      SUCCESS         CreationTime: 1/1/...File
8:10:3... redline.exe   6796 CloseFile           C:\Users\prajs\AppData\Local\Temp\IXP005.TMP\d5898432.exe      SUCCESS                              File
8:10:3... redline.exe   6796 RegQueryKey         HKLM                                                           SUCCESS         Query: HandleTag...  Re
8:10:3... redline.exe   6796 RegQueryKey         HKLM                                                           SUCCESS         Query: Name          Re
```

I checked the Process tree to find the relation between the files loaded into the system



```
⊟ redline.exe (6796)          Win32 Cabinet Se... C:\Users\prajs\O...  Microsoft Corporat... DESKTOP-H1M1... "C:\Users\prajs\O... 8/2/2023 8:10:30... n/a
  ⊟ v6577799.exe (13504)      Win32 Cabinet Se... C:\Users\prajs\Ap... Microsoft Corporat... DESKTOP-H1M1... C:\Users\prajs\Ap... 8/2/2023 8:10:30... 8/2/2023 8:10:31
    ⊟ v6605920.exe (4916)     Win32 Cabinet Se... C:\Users\prajs\Ap... Microsoft Corporat... DESKTOP-H1M1... C:\Users\prajs\Ap... 8/2/2023 8:10:30... 8/2/2023 8:10:31
        a1674716.exe (3924    Healer              C:\Users\prajs\Ap...                      DESKTOP-H1M1... C:\Users\prajs\Ap... 8/2/2023 8:10:31... 8/2/2023 8:10:31
        b7281501.exe (1176                        C:\Users\prajs\Ap...                      DESKTOP-H1M1... C:\Users\prajs\Ap... 8/2/2023 8:10:31... 8/2/2023 8:10:31
      c0580098.exe (12436)                        C:\Users\prajs\Ap...                      DESKTOP-H1M1... C:\Users\prajs\Ap... 8/2/2023 8:10:31... 8/2/2023 8:10:31
    d5898432.exe (12068)      Nirtro CPU          C:\Users\prajs\Ap...                      DESKTOP-H1M1... C:\Users\prajs\Ap... 8/2/2023 8:10:31... n/a
```

Each file was spawned from its previous execution. I checked the registry filter so as how these files were executed, then I got the below results



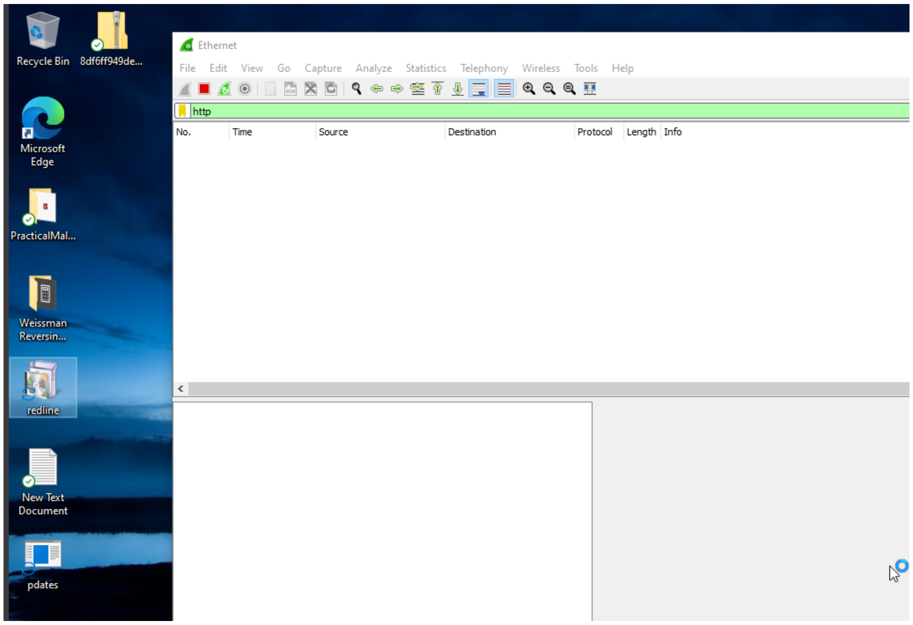
Process Monitor - Sysinternals: www.sysinternals.com
File Edit Event Filter Tools Options Help

Time ...	Process Name	PID	Operation	Path	Result	Detail	Event
9:15:3...	redline.exe	10932	RegCreateKey	HKLM\Software\WOW6432Node\Microsoft\Windows\CurrentVersion\RunOnce	ACCESS DENIED	Desired Access: R...	Registr
9:15:3...	redline.exe	10932	RegCreateKey	HKLM\SOFTWARE\WOW6432Node\Microsoft\Windows\CurrentVersion\RunOnce	ACCESS DENIED	Desired Access: R...	Registr
9:15:3...	redline.exe	10932	RegOpenKey	HKLM\Software\WOW6432Node\Microsoft\Windows\CurrentVersion\RunOnce	SUCCESS	Desired Access: R...	Registr
9:15:3...	redline.exe	10932	RegSetInfoKey	HKLM\SOFTWARE\WOW6432Node\Microsoft\Windows\CurrentVersion\RunOnce	SUCCESS	KeySetInformation...	Registr
9:15:3...	redline.exe	10932	RegQueryKeyS...	HKLM\SOFTWARE\WOW6432Node\Microsoft\Windows\CurrentVersion\RunOnce	BUFFER TOO ...		Registr
9:15:3...	redline.exe	10932	RegQueryKeyS...	HKLM\SOFTWARE\WOW6432Node\Microsoft\Windows\CurrentVersion\RunOnce	SUCCESS		Registr
9:15:3...	redline.exe	10932	RegCloseKey	HKLM\SOFTWARE\WOW6432Node\Microsoft\Windows\CurrentVersion\RunOnce	SUCCESS		Registr

The RunOnce registry keys are used to run/execute the program once after every startup/boot, this is how the malware retains persistence into the system. This is one of the Important IOCs extracted, and plays a major role in attackers success.

Wireshark & ApateDNS

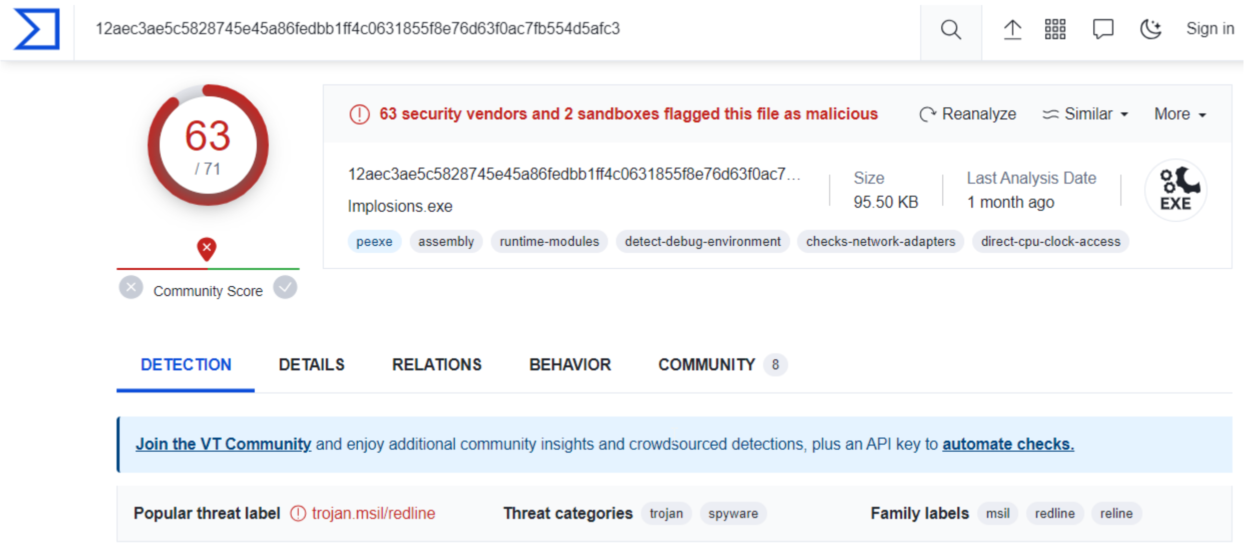
While analysing its C2 communication, I didn't find any major connection to any URL, domain or IP address.



Recycle Bin
8df6ff949de...
Microsoft Edge
PracticalMal...
Weissman Reversin...
redline
New Text Document
pdates
Ethernet
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
http
No. Time Source Destination Protocol Length Info

I used TCPView to check the connection made to remote IP, yet didn't see any results regarding this binary.

As of this malware acted only as a first stage loader, where actual stealer was someone else. To find the actual stealer, I checked each of the binaries loaded from initial binary on VirusTotal and got hits as a tag "Redline" and "Stealer" on binary named "d5898432.exe" which could be the actual stealer. Hence I began my analysis again for this binary.



12aec3ae5c5828745e45a86fedbb1ff4c0631855f8e76d63f0ac7fb554d5afc3

Sign in

63 / 71

Community Score

63 security vendors and 2 sandboxes flagged this file as malicious

Reanalyze | Similar | More

12aec3ae5c5828745e45a86fedbb1ff4c0631855f8e76d63f0ac7...
Implosions.exe

Size: 95.50 KB

Last Analysis Date: 1 month ago

EXE

peexe | assembly | runtime-modules | detect-debug-environment | checks-network-adapters | direct-cpu-clock-access

DETECTION | DETAILS | RELATIONS | BEHAVIOR | COMMUNITY 8

Join the VT Community and enjoy additional community insights and crowdsourced detections, plus an API key to automate checks.

Popular threat label: trojan.msil/redline

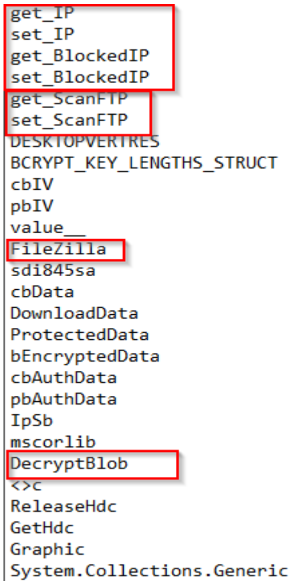
Threat categories: trojan, spyware

Family labels: msil, redline, reline

I checked the strings for this malicious file, once I saw the results, that's where all the juicy stuff was.

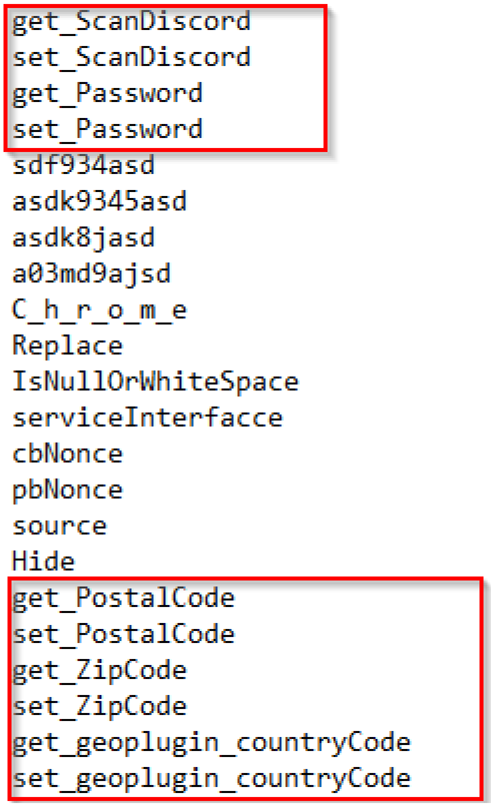
Stealing capacities of the Redline malware:

1. It gets the IP of the machine it infected, scans FTP and FileZilla server



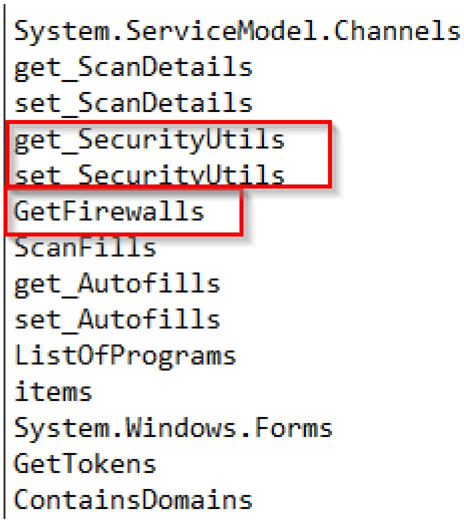
```
get_IP
set_IP
get_BlockedIP
set_BlockedIP
get_ScanFTP
set_ScanFTP
DESKTOPVERTRES
BCRYPT_KEY_LENGTHS_STRUCT
cbIV
pbIV
value__
FileZilla
sdi845sa
cbData
DownloadData
ProtectedData
bEncryptedData
cbAuthData
pbAuthData
IpSb
mscorlib
DecryptBlob
<>c
ReleaseHdc
GetHdc
Graphic
System.Collections.Generic
```

2. Scans for discord accounts, gets any passwords related to the discord accounts, gets postal code, zipcode, country code from any address present



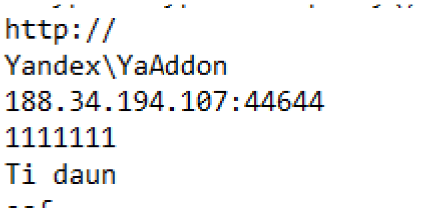
```
get_ScanDiscord
set_ScanDiscord
get_Password
set_Password
sdf934asd
asdk9345asd
asdk8jasd
a03md9ajsd
C_h_r_o_m_e
Replace
IsNullOrWhiteSpace
serviceInterfacce
cbNonce
pbNonce
source
Hide
get_PostalCode
set_PostalCode
get_ZipCode
set_ZipCode
get_geoplugin_countryCode
set_geoplugin_countryCode
```

3. Scans for security related settings/tools, Firewalls present, scan autofills etc.



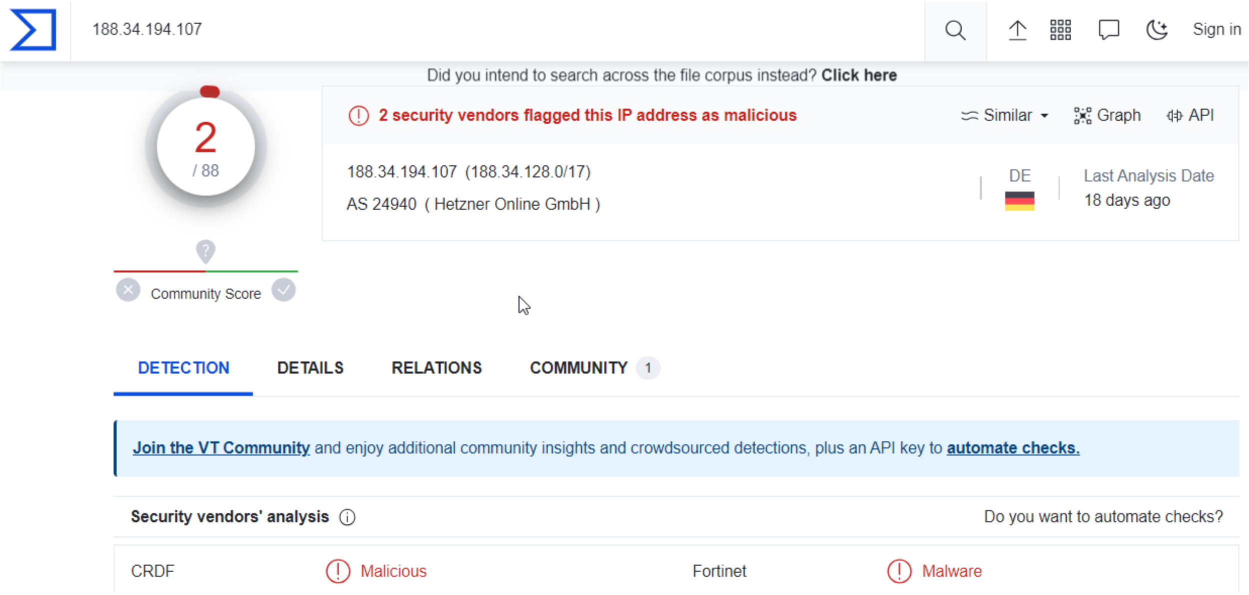
```
System.ServiceModel.Channels
get_ScanDetails
set_ScanDetails
get_SecurityUtils
set_SecurityUtils
GetFirewalls
ScanFills
get_Autofills
set_Autofills
ListOfPrograms
items
System.Windows.Forms
GetTokens
ContainsDomains
```

4. This time I got an important network based IOC, the IP address.



```
http://
Yandex\YaAddon
188.34.194.107:44644
1111111
Ti daun
```

I checked the reputation of this IP address on VirusTotal and was indeed malicious



2 security vendors flagged this IP address as malicious

188.34.194.107 (188.34.128.0/17)
AS 24940 (Hetzner Online GmbH)

DE

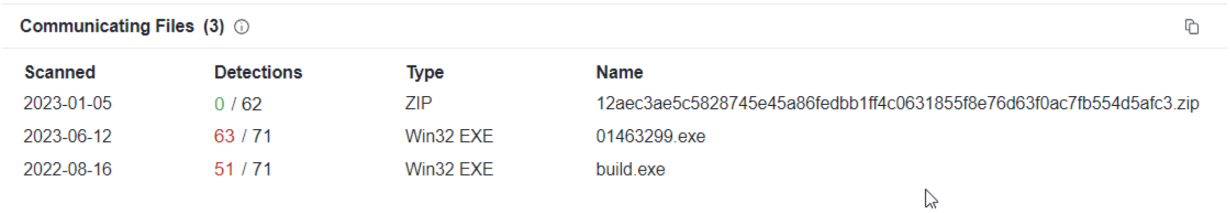
Last Analysis Date
18 days ago

Join the VT Community and enjoy additional community insights and crowdsourced detections, plus an API key to automate checks.

Security vendors' analysis

Do you want to automate checks?

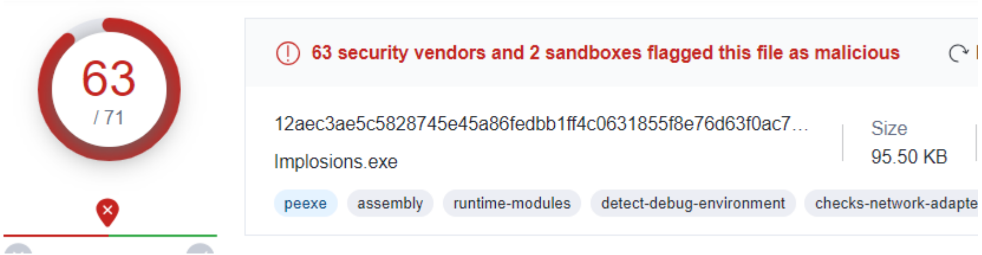
CRDF	Malicious	Fortinet	Malware



Communicating Files (3)

Scanned	Detections	Type	Name
2023-01-05	0 / 62	ZIP	12aec3ae5c5828745e45a86fedbb1ff4c0631855f8e76d63f0ac7fb554d5afc3.zip
2023-06-12	63 / 71	Win32 EXE	01463299.exe
2022-08-16	51 / 71	Win32 EXE	build.exe

Once I clicked the 01463299.exe I received the original name of the file “Impolsions.exe”

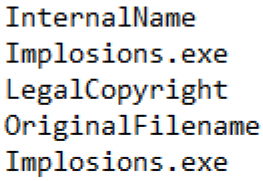


63 security vendors and 2 sandboxes flagged this file as malicious

12aec3ae5c5828745e45a86fedbb1ff4c0631855f8e76d63f0ac7...
Implosions.exe

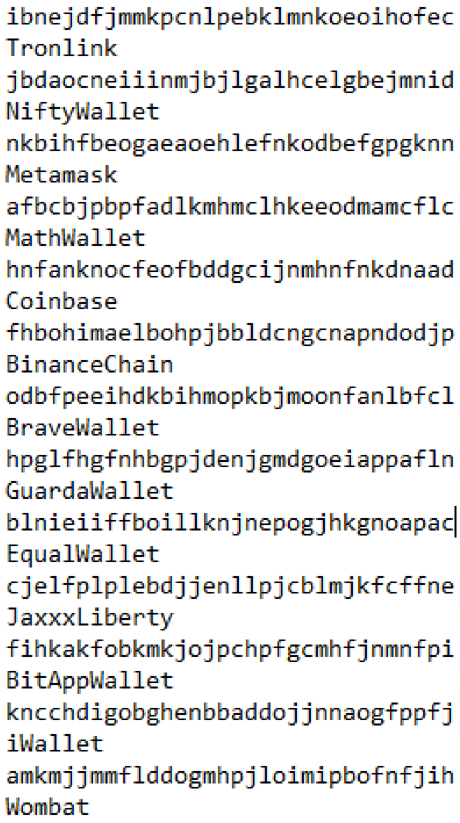
Size
95.50 KB

This filename was present in the strings of the stealer malware too,



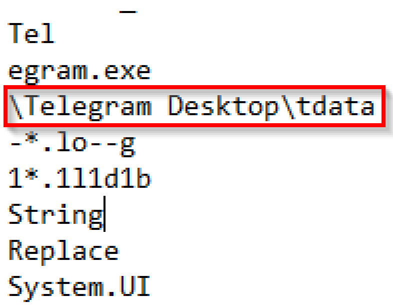
```
InternalName
Implosions.exe
LegalCopyright
OriginalFilename
Implosions.exe
```

5. The malware scans for different crypto wallets



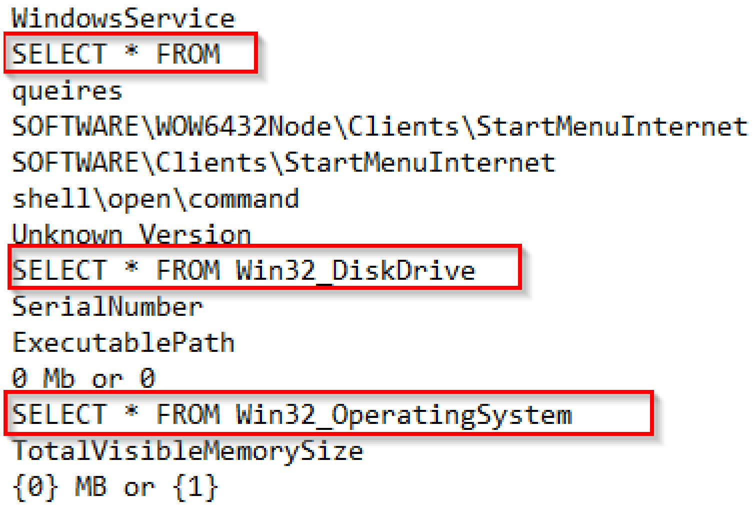
```
ibnejdfjmmkpcnlpebklmnkoeoihofec
Tronlink
jbdaocneiiinmjbjlgalhcelgbejmnid
NiftyWallet
nkbihfbeogaeaoehlefnkodbefgpgknn
Metamask
afbcbjpbpfadlkmhmclhkeeodmamcflc
MathWallet
hnfanknocfeofbddgcijnmhnfnkdnaad
Coinbase
fhbohimaelbohpjbbldcngcnapndodjp
BinanceChain
odbfpeeihdkbihmopkbjmoonfanlbfcl
BraveWallet
hpglfhgfnhbgpjdenjgmdgoeiappafln
GuardaWallet
blnieiiffboillknjnepogjhkgnoapac
EqualWallet
cjelfplplebdjjenllpjcblmjkfcffne
JaxxxLiberty
fihkakfobkmkjojpchpfgcmhfjnmnfpi
BitAppWallet
kncchdigobghenbbaddojjnnaogfppfj
iWallet
amkmjjmmflddogmhpjloimipbofnfjih
Wombat
```

6. It tries to extract Telegram Data



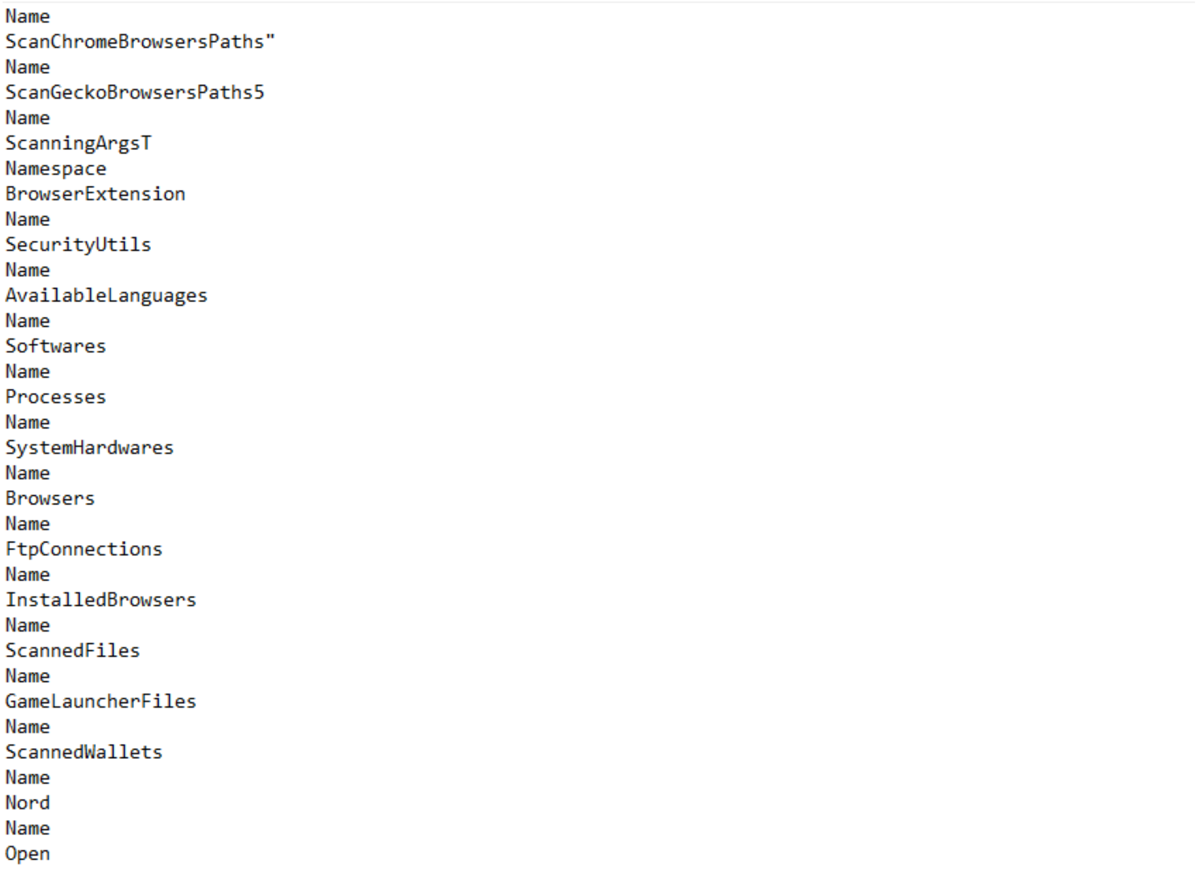
```
        _
Tel
egram.exe
\Telegram Desktop\tdata
-*.lo--g
1*.111d1b
String
Replace
System.UI
```

7. It queries databases and steal them which are related to Windows drives



```
WindowsService
SELECT * FROM
queires
SOFTWARE\WOW6432Node\Clients\StartMenuInternet
SOFTWARE\Clients\StartMenuInternet
shell\open\command
Unknown Version
SELECT * FROM Win32_DiskDrive
SerialNumber
ExecutablePath
0 Mb or 0
SELECT * FROM Win32_OperatingSystem
TotalVisibleMemorySize
{0} MB or {1}
```

8. Scans for FTP connections, installed browsers, processes, softwares, browser extensions and many more



```
Name
ScanChromeBrowsersPaths"
Name
ScanGeckoBrowsersPaths5
Name
ScanningArgsT
Namespace
BrowserExtension
Name
SecurityUtils
Name
AvailableLanguages
Name
Softwares
Name
Processes
Name
SystemHardwares
Name
Browsers
Name
FtpConnections
Name
InstalledBrowsers
Name
ScannedFiles
Name
GameLauncherFiles
Name
ScannedWallets
Name
Nord
Name
Open
```

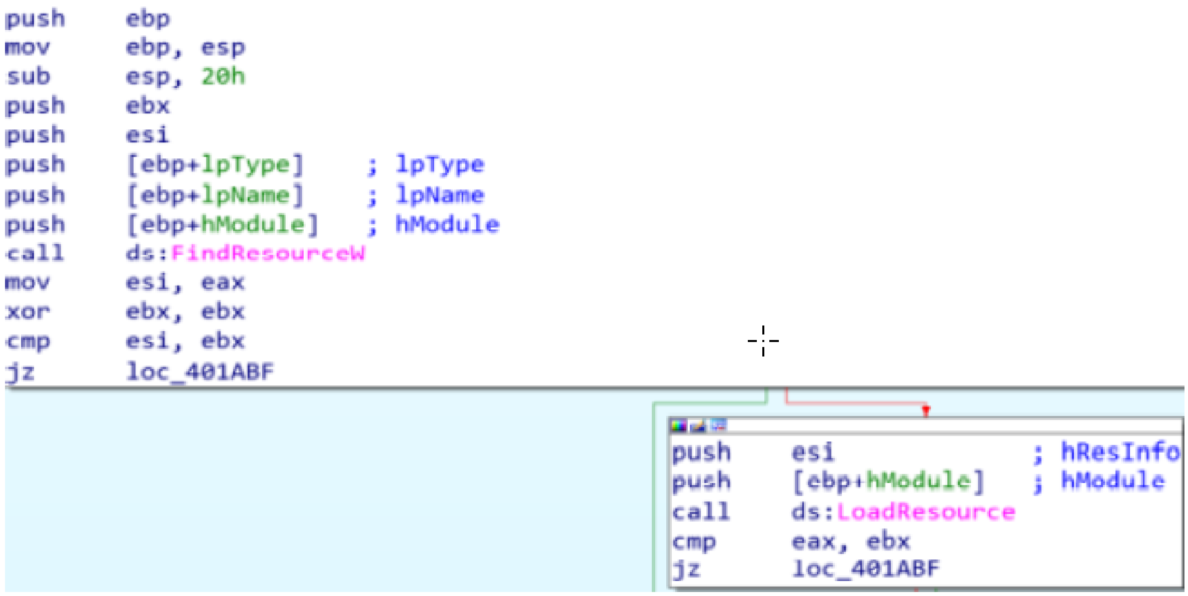
Advanced Static Analysis

I used IDA Freeware 8.0 for my advanced static analysis.

1. Redline (Stage 1 loader)

As I looked through the code, I observed various important code functionalities.

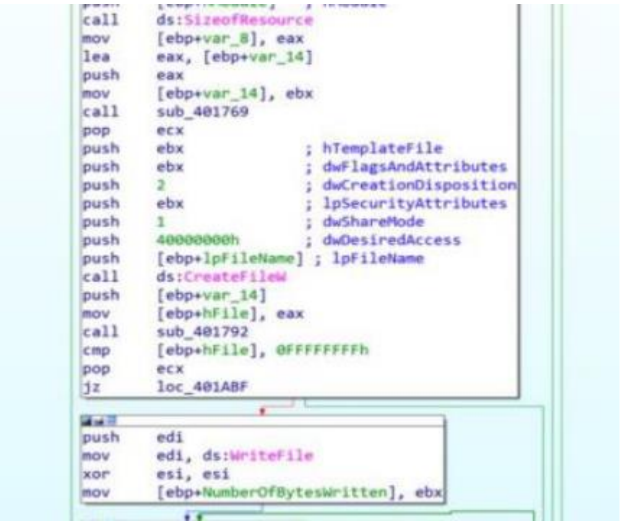
The code, first finds resourse using the arguments that were pushed onto the stack and then uses these arguments to load the resource



```
push    ebp
mov     ebp, esp
sub     esp, 20h
push    ebx
push    esi
push    [ebp+lpType]     ; lpType
push    [ebp+lpName]     ; lpName
push    [ebp+hModule]    ; hModule
call    ds:FindResourceW
mov     esi, eax
xor     ebx, ebx
cmp     esi, ebx
jz      loc_401ABF
```

```
push    esi              ; hResInfo
push    [ebp+hModule]    ; hModule
call    ds:LoadResource
cmp     eax, ebx
jz      loc_401ABF
```

As seen in the basic dynamic analysis, many temp directories were created and exe files were loaded into the filesystem, I tried to search the coder block related to that.

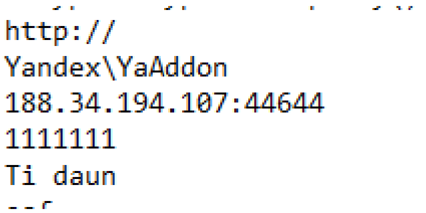


```
call    ds:SizeofResource
mov     [ebp+var_8], eax
lea     eax, [ebp+var_14]
push    eax
mov     [ebp+var_14], ebx
call    sub_401769
pop     ecx
push    ebx                 ; hTemplateFile
push    ebx                 ; dwFlagsAndAttributes
push    2                   ; dwCreationDisposition
push    ebx                 ; lpSecurityAttributes
push    1                   ; dwShareMode
push    40000000h           ; dwDesiredAccess
push    [ebp+lpFileName] ; lpFileName
call    ds:CreateFileW
push    [ebp+var_14]
mov     [ebp+hFile], eax
call    sub_401792
cmp     [ebp+hFile], 0FFFFFFFFh
pop     ecx
jz      loc_401ABF
```

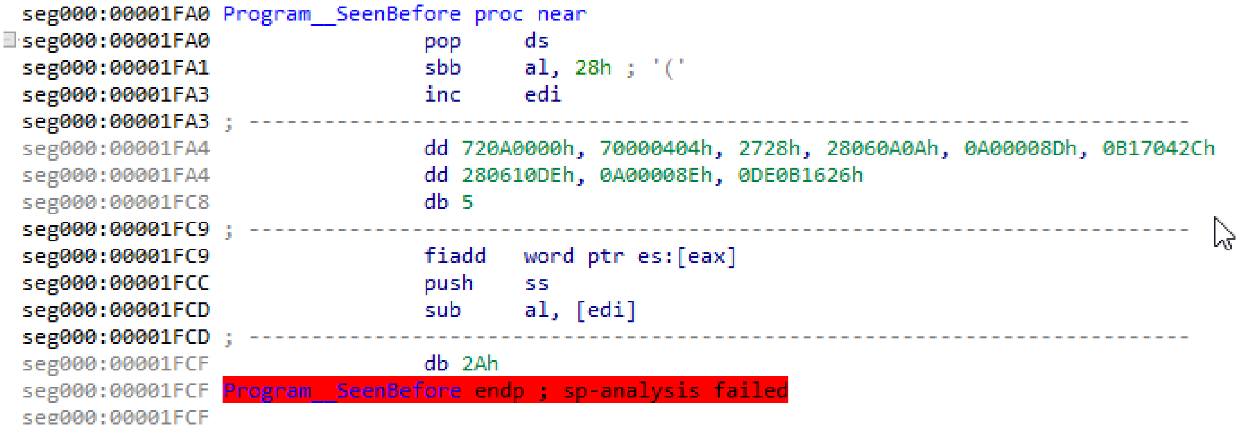
```
push    edi
mov     edi, ds:WriteFile
xor     esi, esi
mov     [ebp+NumberOfBytesWritten], ebx
```

2. Redline Stealer (Actual Stealer)

The first interesting feature of the malware I observed is the seen_before function, this fuction actually checks for a directory creation of path \Yandex\YaaAddon, if it finds this directory, it will continue its execution by notifying the attacker the machine has been infected, if not, it will create one.

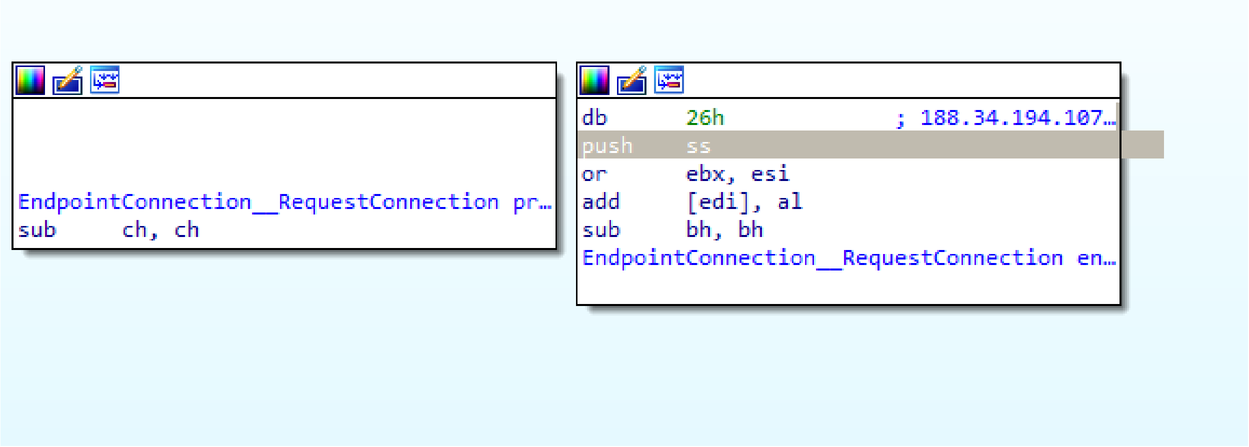


```
http://
Yandex\YaAddon
188.34.194.107:44644
1111111
Ti daun
```



```
seg000:00001FA0 Program__SeenBefore proc near
seg000:00001FA0                 pop     ds
seg000:00001FA1                 sbb     al, 28h ; '('
seg000:00001FA3                 inc     edi
seg000:00001FA3 ; ---------------------------------------------------------------------------
seg000:00001FA4                 dd 720A0000h, 70000404h, 2728h, 28060A0Ah, 0A00008Dh, 0B17042Ch
seg000:00001FA4                 dd 280610DEh, 0A00008Eh, 0DE0B1626h
seg000:00001FC8                 db 5
seg000:00001FC9 ; ---------------------------------------------------------------------------
seg000:00001FC9                 fiadd   word ptr es:[eax]
seg000:00001FCC                 push    ss
seg000:00001FCD                 sub     al, [edi]
seg000:00001FCD ; ---------------------------------------------------------------------------
seg000:00001FCF                 db 2Ah
seg000:00001FCF Program__SeenBefore endp ; sp-analysis failed
seg000:00001FCF
```

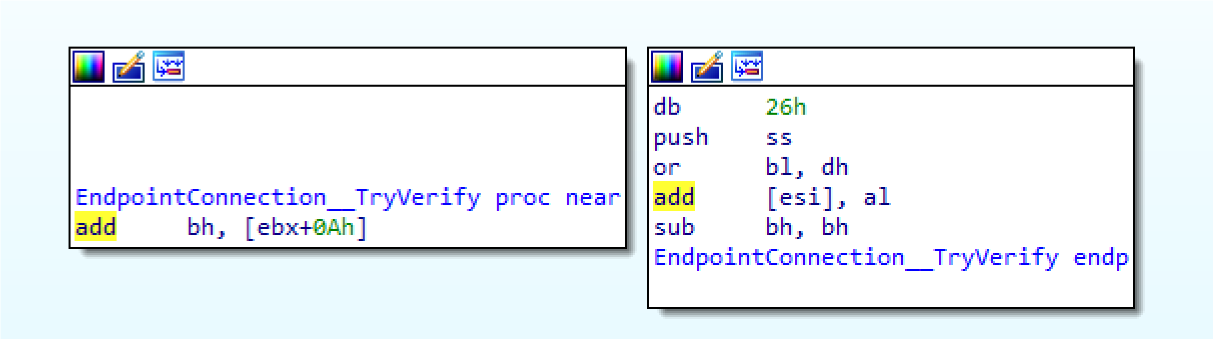
It then connects to the remote IP as mentioned in the strings section.



```
EndpointConnection__RequestConnection pr…
sub     ch, ch
```

```
db      26h             ; 188.34.194.107…
push    ss
or      ebx, esi
add     [edi], al
sub     bh, bh
EndpointConnection__RequestConnection en…
```

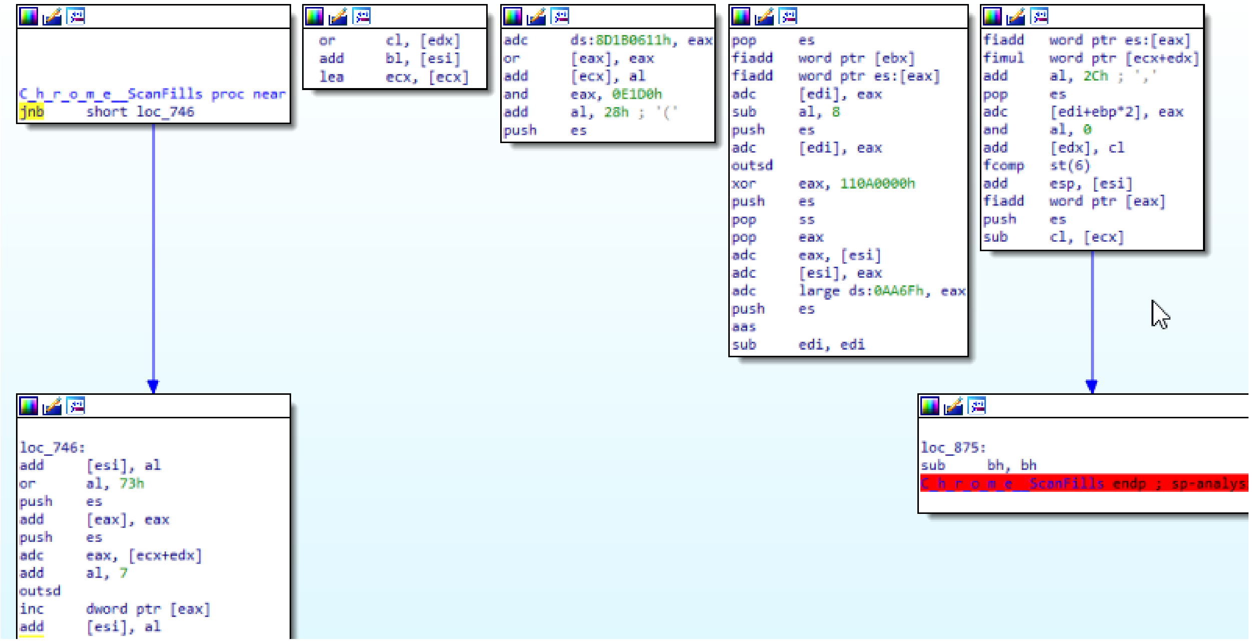
It also verifies the connection made to the remote IP address



```
EndpointConnection__TryVerify proc near
add     bh, [ebx+0Ah]
```

```
db      26h
push    ss
or      bl, dh
add     [esi], al
sub     bh, bh
EndpointConnection__TryVerify endp
```

After this I looked for actual stealing capabilities in the code



```
C_h_r_o_m_e__ScanFills proc near
jnb     short loc_746
```

```
or      cl, [edx]
add     bl, [esi]
lea     ecx, [ecx]
```

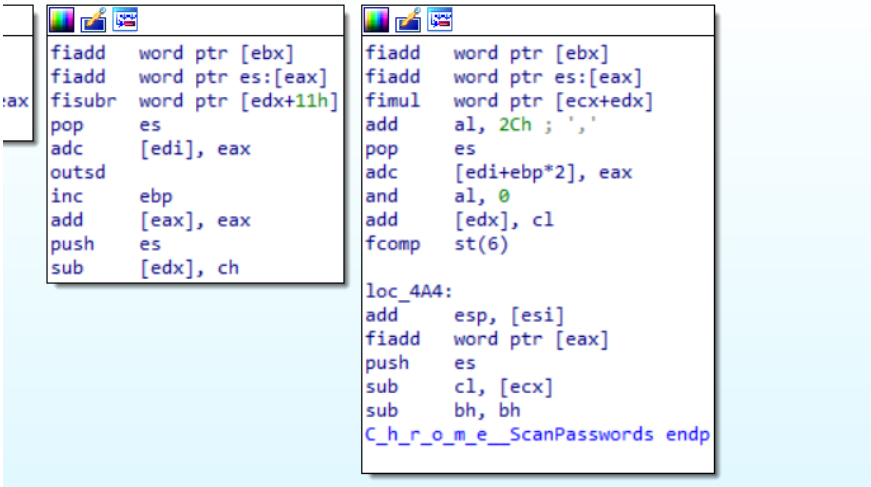
```
adc     ds:8D1B0611h, eax
or      [eax], eax
add     [ecx], al
and     eax, 0E1D0h
add     al, 28h ; '('
push    es
```

```
pop     es
fiadd   word ptr [ebx]
fiadd   word ptr es:[eax]
adc     [edi], eax
sub     al, 8
push    es
adc     [edi], eax
outsd
xor     eax, 110A0000h
push    es
pop     ss
pop     eax
adc     eax, [esi]
adc     [esi], eax
adc     large ds:0AA6Fh, eax
push    es
aas
sub     edi, edi
```

```
fiadd   word ptr es:[eax]
fimul   word ptr [ecx+edx]
add     al, 2Ch ; ','
pop     es
adc     [edi+ebp*2], eax
and     al, 0
add     [edx], cl
fcomp   st(6)
add     esp, [esi]
fiadd   word ptr [eax]
push    es
sub     cl, [ecx]
```

```
loc_746:
add     [esi], al
or      al, 73h
push    es
add     [eax], eax
push    es
adc     eax, [ecx+edx]
add     al, 7
outsd
inc     dword ptr [eax]
add     [esi], al
```

```
loc_875:
sub     bh, bh
C_h_r_o_m_e__ScanFills endp ; sp-analys
```



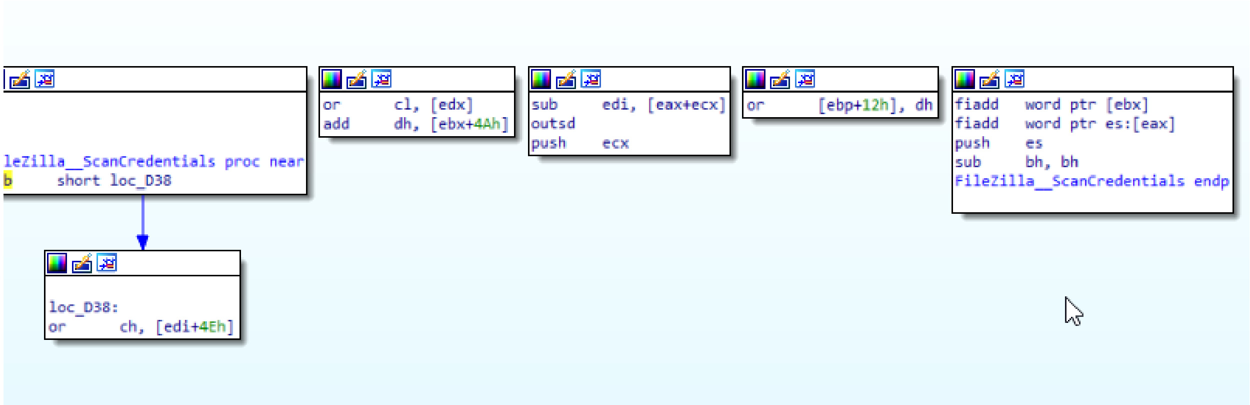
```
fiadd   word ptr [ebx]
fiadd   word ptr es:[eax]
fisubr  word ptr [edx+11h]
pop     es
adc     [edi], eax
outsd
inc     ebp
add     [eax], eax
push    es
sub     [edx], ch
```

```
fiadd   word ptr [ebx]
fiadd   word ptr es:[eax]
fimul   word ptr [ecx+edx]
add     al, 2Ch ; ','
pop     es
adc     [edi+ebp*2], eax
and     al, 0
add     [edx], cl
fcomp   st(6)

loc_4A4:
add     esp, [esi]
fiadd   word ptr [eax]
push    es
sub     cl, [ecx]
sub     bh, bh
C_h_r_o_m_e__ScanPasswords endp
```

This code block shows how the malware tries to scan the chrome autofills and chrome passwords, the malware author has created separate functions for each of the stealing capability.

Function Scanning FileZilla credentials



```
leZilla__ScanCredentials proc near
b       short loc_D38
```

```
loc_D38:
or      ch, [edi+4Eh]
```

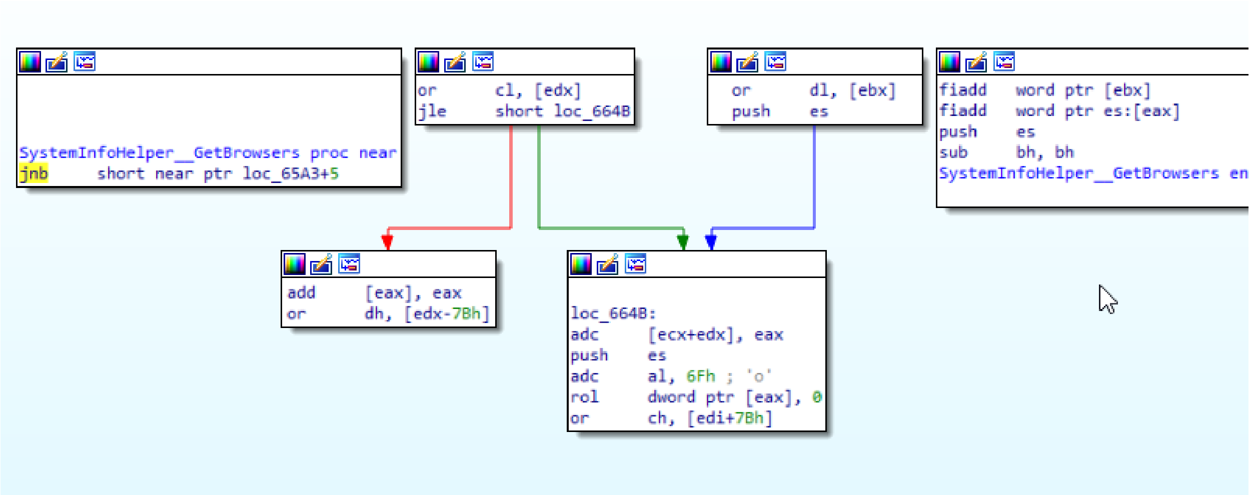
```
or      cl, [edx]
add     dh, [ebx+4Ah]
```

```
sub     edi, [eax+ecx]
outsd
push    ecx
```

```
or      [ebp+12h], dh
```

```
fiadd   word ptr [ebx]
fiadd   word ptr es:[eax]
push    es
sub     bh, bh
FileZilla__ScanCredentials endp
```

Function code block to get a list of browser details



```
SystemInfoHelper__GetBrowsers proc near
jnb     short near ptr loc_65A3+5
```

```
or      cl, [edx]
jle     short loc_664B
```

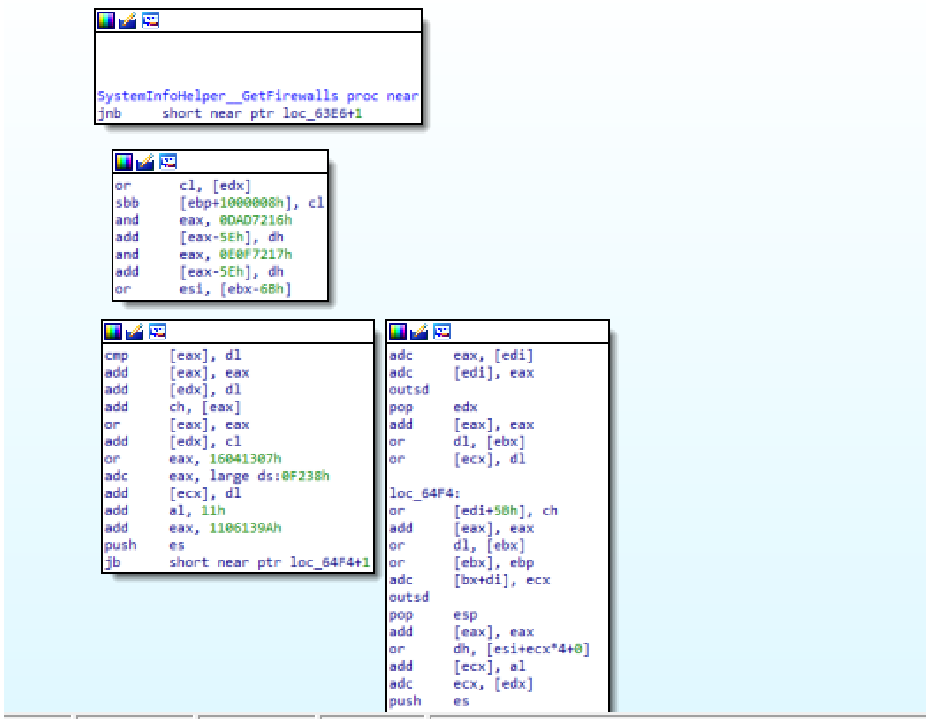
```
add     [eax], eax
or      dh, [edx-7Bh]
```

```
 or      dl, [ebx]
 push    es
```

```
loc_664B:
adc     [ecx+edx], eax
push    es
adc     al, 6Fh ; 'o'
rol     dword ptr [eax], 0
or      ch, [edi+7Bh]
```

```
fiadd   word ptr [ebx]
fiadd   word ptr es:[eax]
push    es
sub     bh, bh
SystemInfoHelper__GetBrowsers en
```

Code to get the list of firewall



```
SystemInfoHelper__GetFirewalls proc near
jnb     short near ptr loc_63E6+1
```

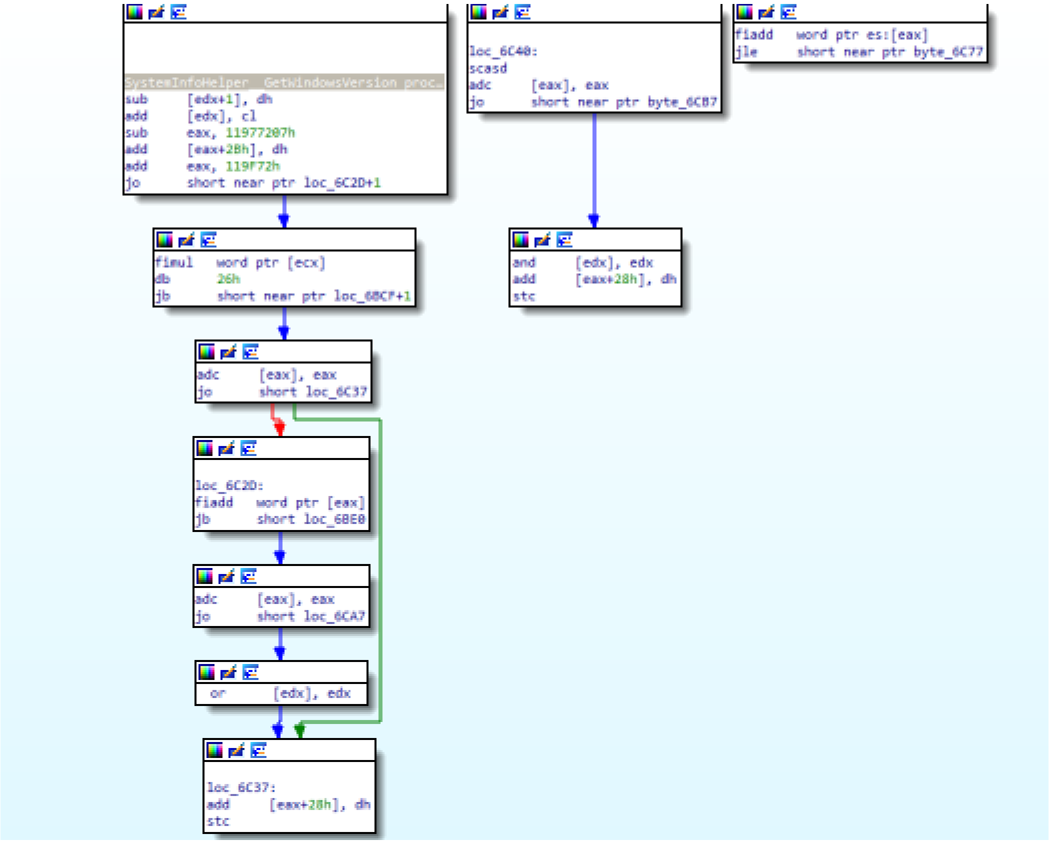
```
or      cl, [edx]
sbb     [ebp+1000008h], cl
and     eax, 0DAD7216h
add     [eax-5Eh], dh
and     eax, 0E0F7217h
add     [eax-5Eh], dh
or      esi, [ebx-6Bh]
```

```
cmp     [eax], dl
add     [eax], eax
add     [edx], dl
add     ch, [eax]
or      [eax], eax
add     [edx], cl
or      eax, 16041307h
adc     eax, large ds:0F238h
add     [ecx], dl
add     al, 11h
add     eax, 1106139Ah
push    es
jb      short near ptr loc_64F4+1
```

```
adc     eax, [edi]
adc     [edi], eax
outsd
pop     edx
add     [eax], eax
or      dl, [ebx]
or      [ecx], dl

loc_64F4:
or      [edi+58h], ch
add     [eax], eax
or      dl, [ebx]
or      [ebx], ebp
adc     [bx+di], ecx
outsd
pop     esp
add     [eax], eax
or      dh, [esi+ecx*4+0]
add     [ecx], al
adc     ecx, [edx]
push    es
```

Get windows versions



```
SystemInfoHelper__GetWindowsVersion proc…
sub     [edx+1], dh
add     [edx], cl
sub     eax, 11977207h
add     [eax+28h], dh
add     eax, 119F72h
jo      short near ptr loc_6C2D+1
```

```
fimul   word ptr [ecx]
db      26h
jb      short near ptr loc_68CF+1
```

```
adc     [eax], eax
jo      short loc_6C37
```

```
loc_6C2D:
fiadd   word ptr [eax]
jb      short loc_68E0
```

```
adc     [eax], eax
jo      short loc_6CA7
```

```
 or      [edx], edx
```

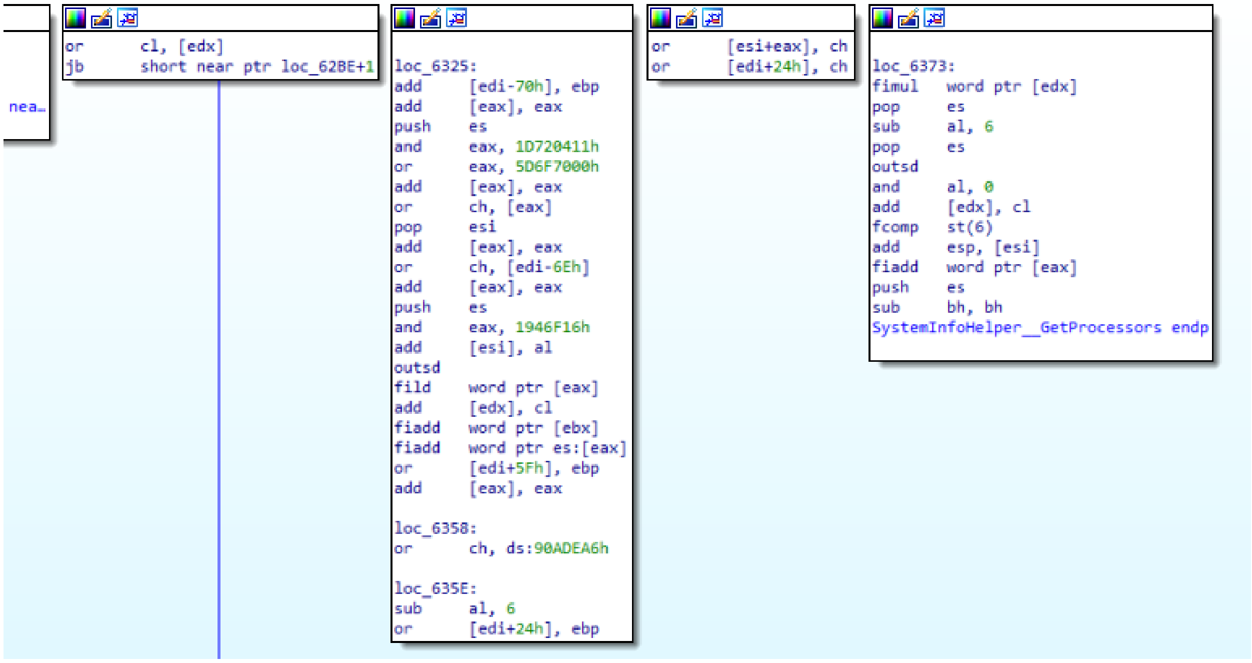
```
loc_6C37:
add     [eax+28h], dh
stc
```

```
loc_6C40:
scasd
adc     [eax], eax
jo      short near ptr byte_6C87
```

```
and     [edx], edx
add     [eax+28h], dh
stc
```

```
fiadd   word ptr es:[eax]
jle     short near ptr byte_6C77
```

Get current list of processes



```
nea…
```

```
or      cl, [edx]
jb      short near ptr loc_62BE+1
```

```
loc_6325:
add     [edi-70h], ebp
add     [eax], eax
push    es
and     eax, 1D720411h
or      eax, 5D6F7000h
add     [eax], eax
or      ch, [eax]
pop     esi
add     [eax], eax
or      ch, [edi-6Eh]
add     [eax], eax
push    es
and     eax, 1946F16h
add     [esi], al
outsd
fild    word ptr [eax]
add     [edx], cl
fiadd   word ptr [ebx]
fiadd   word ptr es:[eax]
or      [edi+5Fh], ebp
add     [eax], eax

loc_6358:
or      ch, ds:90ADEA6h

loc_635E:
sub     al, 6
or      [edi+24h], ebp
```

```
or      [esi+eax], ch
or      [edi+24h], ch
```

```
loc_6373:
fimul   word ptr [edx]
pop     es
sub     al, 6
pop     es
outsd
and     al, 0
add     [edx], cl
fcomp   st(6)
add     esp, [esi]
fiadd   word ptr [eax]
push    es
sub     bh, bh
SystemInfoHelper__GetProcessors endp
```

I am surprised not to see any API calls; I presume these are imported dynamically and hence cannot view them in the static code. Since I am able to view the rest of the code, it doesn't look like the malware is obfuscated, the only possibility hits are the runtime imports of the API calls. This entire behavior can be revealed in the advanced dynamic stage.

IOCs Discovered

Host based artifacts :

9b05f893286b23204a89b982ec2b5a95

bf61df210e8a0e3a58d341582b070f3b

78040623ca989f89701b6b7424f1dd2b

C:\Users\prajs\AppData\Local\Temp\IXP000.TMP\v6577799.exe

C:\Users\prajs\AppData\Local\Temp\IXP000.TMP\d5898432.exe

C:\Users\prajs\AppData\Local\Temp\IXP001.TMP\v6605920.exe

C:\Users\prajs\AppData\Local\Temp\IXP001.TMP\c0580098.exe

C:\Users\prajs\AppData\Local\Temp\IXP002.TMP\a1674716.exe

C:\Users\prajs\AppData\Local\Temp\IXP002.TMP\b7281501.exe

HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\RunOnce

HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\RunOnce\wextract_cleanup0

HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\RunOnce\wextract_cleanup1

Network Based Artifacts

188[.]34[.]194[.]107

Potential Danger from the Malware

The malware is capable of stealing highly classified and personal information, impacting Confidentiality and Integrity, Browsers cache, passwords, credential, account information, even geographical locations, system information, everything one might think of is compromised. This breach can cost an organization not just financially but also reputation would be at stake.

How can this be prevented

- Monitoring file systems, updating the system with latest database of anti-virus and firewalls which out block outgoing connections, not just connections, should be able to detect the data exfiltration.
- Users in the organizations should employ multi-factor authentication for their accounts.
- Operating systems must be updated and all servers should be patched on a regular basis.