FareIt Malware Analysis using static and dynamic method

What is FareIt?

This form of malware was discovered in 2012, but has continued modifying throughout the years to bypass anti-virus protection. It is an information stealer that targets FTP credentials, email passwords and browser stored passwords. During dynamic analysis, it is observed all of the above being performed after the malware disabled local security tools.

How it is spreading to the crowd?

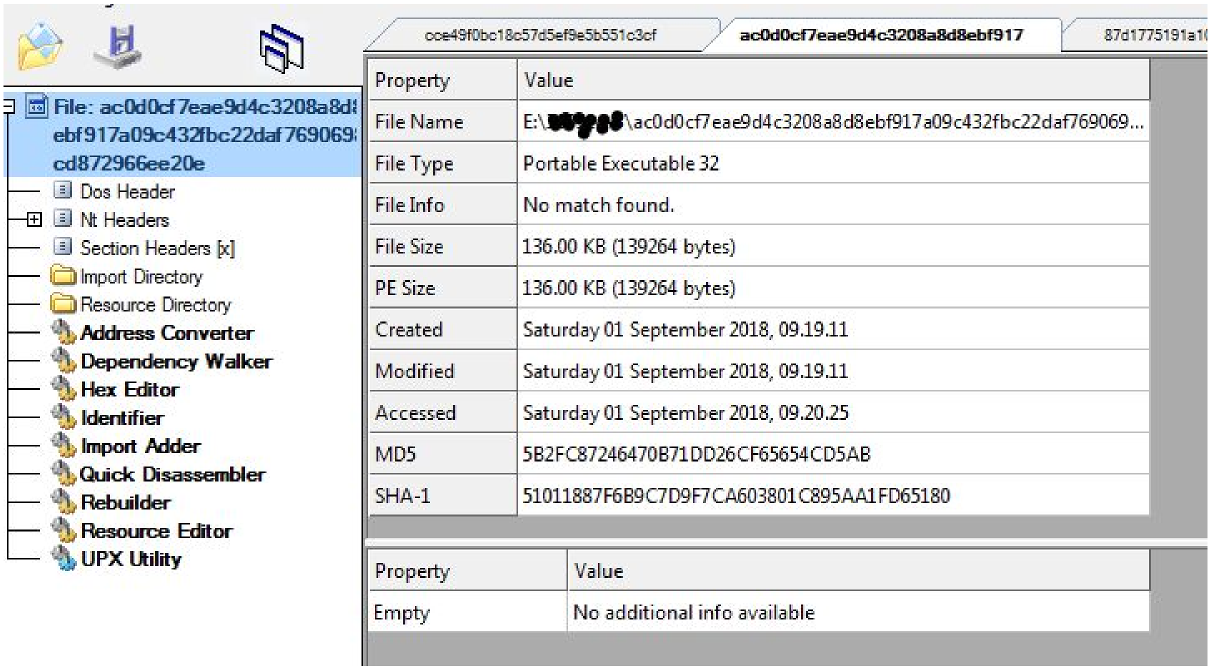
The most recent Fareit malware threat is being distributed via a phishing attack. A phishing attack is an email with a malicious link or attachment, designed to make you click on those links/attachments. This most recent phishing attack includes malicious executable disguised as a DOC, XLS, ISO, PPT file attachment, which includes the malware. Once the user downloads the Attachment, their computer becomes infected and the malware scans for any credentials that may be of value. This may range from banking information, various account login credentials, administrative credentials, etc.

Spam email with an attachment *Attached Malicious Document file* *Fareit Malware*

Fareit Static Analysis:

SHA256: ac0d0cf7eae9d4c3208a8d8ebf917a09c432fbc22daf7690698cd872966ee20e

File General information:

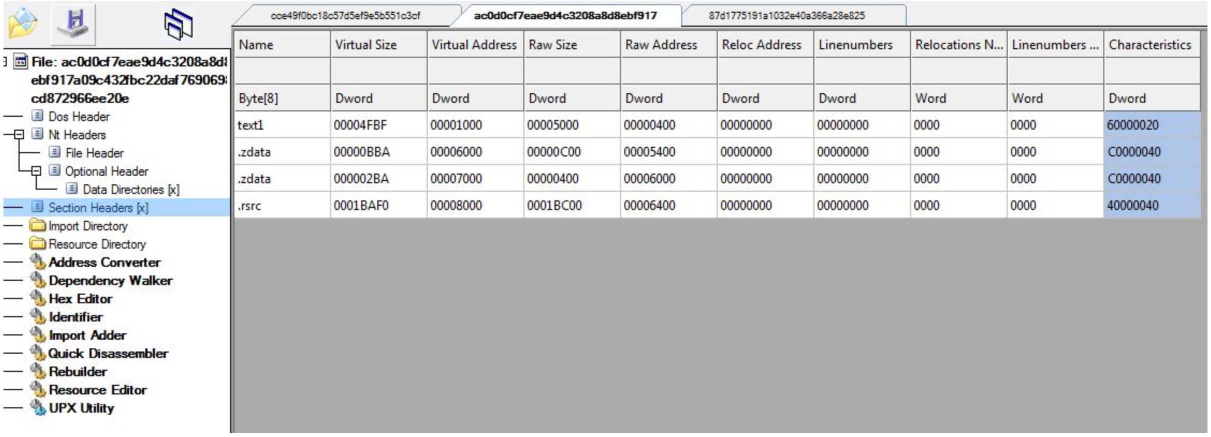


Property	Value
File Name	E:\[redacted]\ac0d0cf7eae9d4c3208a8d8ebf917a09c432fbc22daf769069...
File Type	Portable Executable 32
File Info	No match found.
File Size	136.00 KB (139264 bytes)
PE Size	136.00 KB (139264 bytes)
Created	Saturday 01 September 2018, 09.19.11
Modified	Saturday 01 September 2018, 09.19.11
Accessed	Saturday 01 September 2018, 09.20.25
MD5	5B2FC87246470B71DD26CF65654CD5AB
SHA-1	51011887F6B9C7D9F7CA603801C895AA1FD65180

Property	Value
Empty	No additional info available

- No information related to Company Name, File Description, Legal Copyright, Product Version ext. (suspicious)

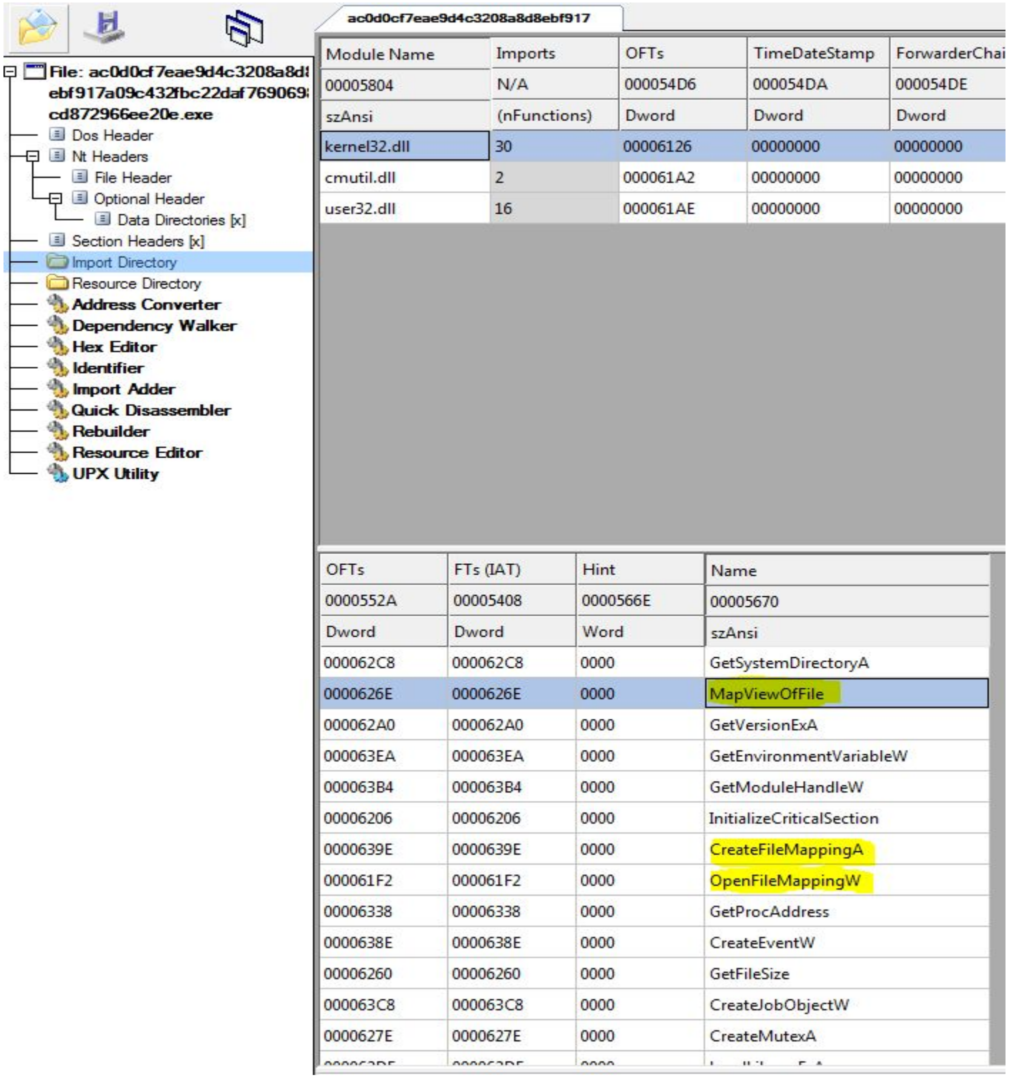
File Section headers:



Name	Virtual Size	Virtual Address	Raw Size	Raw Address	Reloc Address	Linenumbers	Relocations N...	Linenumbers ...	Characteristics
Byte[8]	Dword	Dword	Dword	Dword	Dword	Dword	Word	Word	Dword
text1	00004FBF	00001000	00005000	00000400	00000000	00000000	0000	0000	60000020
.zdata	00000BBA	00006000	00000C00	00005400	00000000	00000000	0000	0000	C0000040
.zdata	000002BA	00007000	00000400	00006000	00000000	00000000	0000	0000	C0000040
.rsrc	0001BAF0	00008000	0001BC00	00006400	00000000	00000000	0000	0000	40000040

- Odd looking Non-Standard sections name: text1, odata, .wdata (suspicious)

File Import Directory:



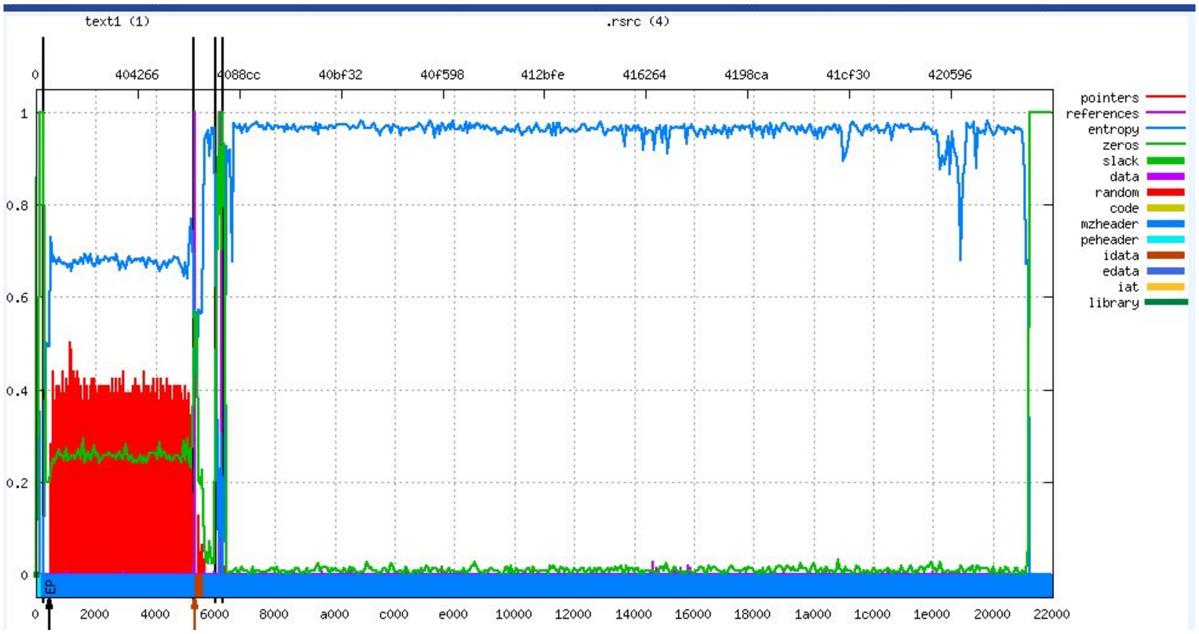
Module Name	Imports	OFTs	TimeDateStamp	ForwarderChai
00005804	N/A	000054D6	000054DA	000054DE
szAnsi	(nFunctions)	Dword	Dword	Dword
kernel32.dll	30	00006126	00000000	00000000
cmutil.dll	2	000061A2	00000000	00000000
user32.dll	16	000061AE	00000000	00000000

OFTs	FTs (IAT)	Hint	Name
0000552A	00005408	0000566E	00005670
Dword	Dword	Word	szAnsi
000062C8	000062C8	0000	GetSystemDirectoryA
0000626E	0000626E	0000	MapViewOfFile
000062A0	000062A0	0000	GetVersionExA
000063EA	000063EA	0000	GetEnvironmentVariableW
000063B4	000063B4	0000	GetModuleHandleW
00006206	00006206	0000	InitializeCriticalSection
0000639E	0000639E	0000	CreateFileMappingA
000061F2	000061F2	0000	OpenFileMappingW
00006338	00006338	0000	GetProcAddress
0000638E	0000638E	0000	CreateEventW
00006260	00006260	0000	GetFileSize
000063C8	000063C8	0000	CreateJobObjectW
0000627E	0000627E	0000	CreateMutexA

- It is importing some functions related to File mapping.

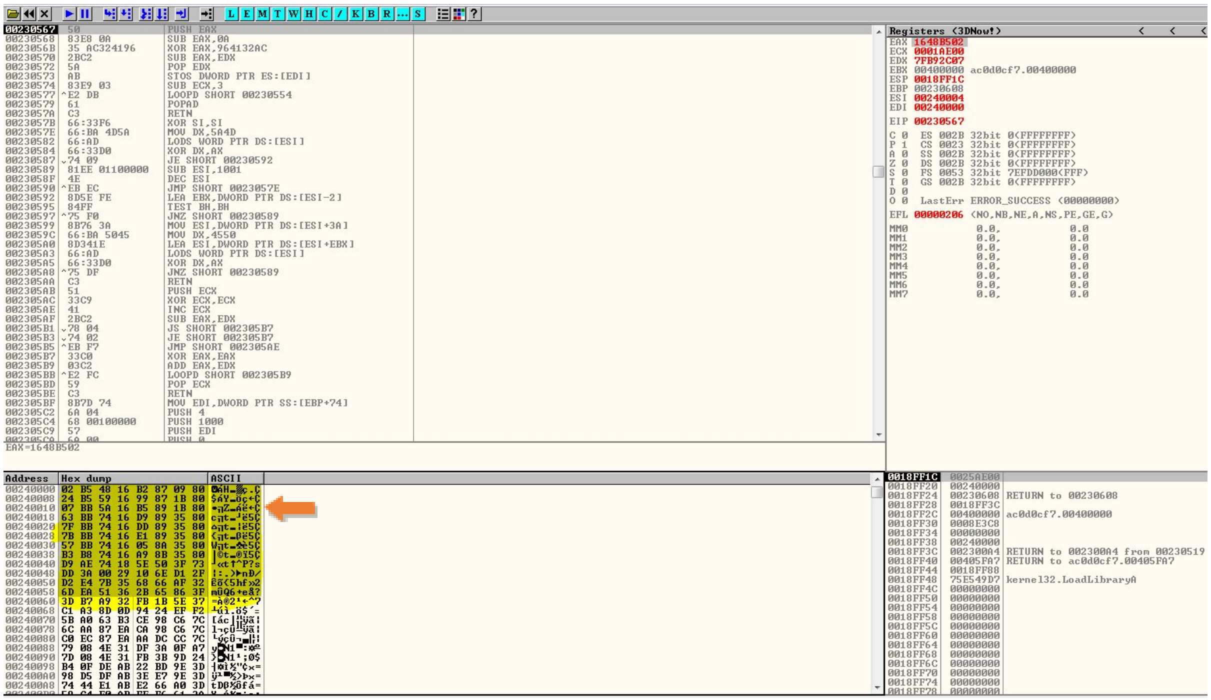
MapViewOfFile: Maps a view of a file mapping into the address space of the calling process.
CreateFileMappingA: Creates or opens a named or unnamed file mapping object for a specified file.
OpenFileMappingW: Opens a named file mapping object.

File Heu:



- File’s Resource section is very large and dens, probably having encrypted data

I have opened same file in ollyDbg and found that file is putting encrypted data to Hex Dump



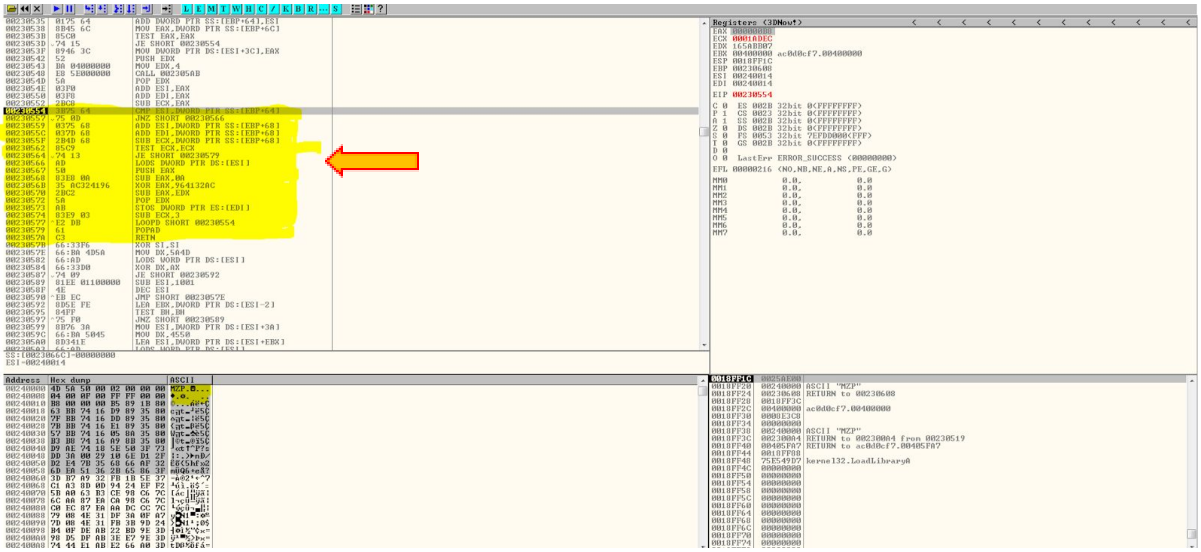
```
00230567  50              PUSH EAX
00230568  83E8 0A         SUB EAX,0A
0023056B  35 AC324196     XOR EAX,964132AC
00230570  2BC2            SUB EAX,EDX
00230572  5A              POP EDX
00230573  AB              STOS DWORD PTR ES:[EDI]
00230574  83E9 03         SUB ECX,3
00230577 ^E2 DB           LOOPD SHORT 00230554
00230579  61              POPAD
0023057A  C3              RETN
0023057B  66:33F6         XOR SI,SI
0023057E  66:BA 4D5A      MOV DX,5A4D
00230582  66:AD           LODS WORD PTR DS:[ESI]
00230584  66:33D0         XOR DX,AX
00230587 v74 09           JE SHORT 00230592
00230589  81EE 01100000   SUB ESI,1001
0023058F  4E              DEC ESI
00230590 ^EB EC           JMP SHORT 0023057E
00230592  8D5E FE         LEA EBX,DWORD PTR DS:[ESI-2]
00230595  84FF            TEST BH,BH
00230597 ^75 F0           JNZ SHORT 00230589
00230599  8B76 3A         MOV ESI,DWORD PTR DS:[ESI+3A]
0023059C  66:BA 5045      MOV DX,4550
002305A0  8D341E          LEA ESI,DWORD PTR DS:[ESI+EBX]
002305A3  66:AD           LODS WORD PTR DS:[ESI]
002305A5  66:33D0         XOR DX,AX
002305A8 ^75 DF           JNZ SHORT 00230589
002305AA  C3              RETN
002305AB  51              PUSH ECX
002305AC  33C9            XOR ECX,ECX
002305AE  41              INC ECX
002305AF  2BC2            SUB EAX,EDX
002305B1 v78 04           JS SHORT 002305B7
002305B3 v74 02           JE SHORT 002305B7
002305B5 ^EB F7           JMP SHORT 002305AE
002305B7  33C0            XOR EAX,EAX
002305B9  03C2            ADD EAX,EDX
002305BB ^E2 FC           LOOPD SHORT 002305B9
002305BD  59              POP ECX
002305BE  C3              RETN
002305BF  8B7D 74         MOV EDI,DWORD PTR SS:[EBP+74]
002305C2  6A 04           PUSH 4
002305C4  68 00100000     PUSH 1000
002305C9  57              PUSH EDI
002305CA  6A 00           PUSH 0
EAX=1648B502
```

```
Registers (3DNow!)
EAX 1648B502
ECX 0001AE00
EDX 7FB92C07
EBX 00400000 ac0d0cf7.00400000
ESP 0018FF1C
EBP 00230608
ESI 00240004
EDI 00240000

EIP 00230567

C 0  ES 002B 32bit 0(FFFFFFFF)
P 1  CS 0023 32bit 0(FFFFFFFF)
A 0  SS 002B 32bit 0(FFFFFFFF)
Z 0  DS 002B 32bit 0(FFFFFFFF)
S 0  FS 0053 32bit 7EFDD000(FFF)
T 0  GS 002B 32bit 0(FFFFFFFF)
D 0
O 0  LastErr ERROR_SUCCESS (00000000)

EFL 00000206 (NO,NB,NE,A,NS,PE,GE,G)

MM0          0.0,          0.0
MM1          0.0,          0.0
MM2          0.0,          0.0
MM3          0.0,          0.0
MM4          0.0,          0.0
MM5          0.0,          0.0
MM6          0.0,          0.0
MM7          0.0,          0.0
```

```
Address   Hex dump                  ASCII
00240000  02 B5 48 16 B2 87 09 80   ☻áH▬▒ç.Ç
00240008  24 B5 59 16 99 87 1B 80   $áY▬öç←Ç
00240010  07 BB 5A 16 B5 89 1B 80   •╗Z▬Áë←Ç
00240018  63 BB 74 16 D9 89 35 80   c╗t▬┘ë5Ç
00240020  7F BB 74 16 DD 89 35 80   ⌂╗t▬¦ë5Ç
00240028  7B BB 74 16 E1 89 35 80   {╗t▬ßë5Ç
00240030  57 BB 74 16 05 8A 35 80   W╗t▬♣è5Ç
00240038  B3 B8 74 16 A9 8B 35 80   │©t▬®ï5Ç
00240040  D9 AE 74 18 5E 50 3F 73   ┘«t↑^P?s
00240048  DD 3A 00 29 10 6E D1 2F   ¦:.)►nÐ/
00240050  D2 E4 7B 35 68 66 AF 32   Êõ{5hf»2
00240058  6D EA 51 36 2B 65 86 3F   mÛQ6+eå?
00240060  3D B7 A9 32 FB 1B 5E 37   =À®2¹←^7
00240068  C1 A3 8D 0D 94 24 EF F2   ┴úì.ö$´=
00240070  5B A0 63 B3 CE 98 C6 7C   [ác│╬ÿã|
00240078  6C AA 87 EA CA 98 C6 7C   l¬çÛ╩ÿã|
00240080  C0 EC 87 EA AA DC CC 7C   └ýçÛ¬▄╠|
00240088  79 08 4E 31 DF 3A 0F A7   y◘N1▀:☼º
00240090  7D 08 4E 31 FB 3B 9D 24   }◘N1¹;Ø$
00240098  B4 0F DE AB 22 BD 9E 3D   ┤☼Ì½"¢×=
002400A0  98 D5 DF AB 3E E7 9E 3D   ÿı▀½>þ×=
002400A8  74 44 E1 AB E2 66 A0 3D   tDß½Ôfá=
```

```
0018FF1C  0025AE00
0018FF20  00240000
0018FF24  00230608  RETURN to 00230608
0018FF28  0018FF3C
0018FF2C  00400000  ac0d0cf7.00400000
0018FF30  0008E3C8
0018FF34  00000000
0018FF38  00240000
0018FF3C  002300A4  RETURN to 002300A4 from 00230519
0018FF40  00405FA7  RETURN to ac0d0cf7.00405FA7
0018FF44  0018FF88
0018FF48  75E549D7  kernel32.LoadLibraryA
0018FF4C  00000000
0018FF50  00000000
0018FF54  00000000
0018FF58  00000000
0018FF5C  00000000
0018FF60  00000000
0018FF64  00000000
0018FF68  00000000
0018FF6C  00000000
0018FF70  00000000
0018FF74  00000000
0018FF78  00000000
```

Now the data will get decrypted by a decryption loop:



```
00230535  0175 64         ADD DWORD PTR SS:[EBP+64],ESI
00230538  8B45 6C         MOV EAX,DWORD PTR SS:[EBP+6C]
0023053B  85C0            TEST EAX,EAX
0023053D v74 15           JE SHORT 00230554
0023053F  8946 3C         MOV DWORD PTR DS:[ESI+3C],EAX
00230542  52              PUSH EDX
00230543  BA 04000000     MOV EDX,4
00230548  E8 5E000000     CALL 002305AB
0023054D  5A              POP EDX
0023054E  03F0            ADD ESI,EAX
00230550  03F8            ADD EDI,EAX
00230552  2BC8            SUB ECX,EAX
00230554  3B75 64         CMP ESI,DWORD PTR SS:[EBP+64]
00230557 v75 0D           JNZ SHORT 00230566
00230559  0375 68         ADD ESI,DWORD PTR SS:[EBP+68]
0023055C  037D 68         ADD EDI,DWORD PTR SS:[EBP+68]
0023055F  2B4D 68         SUB ECX,DWORD PTR SS:[EBP+68]
00230562  85C9            TEST ECX,ECX
00230564 v74 13           JE SHORT 00230579
00230566  AD              LODS DWORD PTR DS:[ESI]
00230567  50              PUSH EAX
00230568  83E8 0A         SUB EAX,0A
0023056B  35 AC324196     XOR EAX,964132AC
00230570  2BC2            SUB EAX,EDX
00230572  5A              POP EDX
00230573  AB              STOS DWORD PTR ES:[EDI]
00230574  83E9 03         SUB ECX,3
00230577 ^E2 DB           LOOPD SHORT 00230554
00230579  61              POPAD
0023057A  C3              RETN
0023057B  66:33F6         XOR SI,SI
0023057E  66:BA 4D5A      MOV DX,5A4D
00230582  66:AD           LODS WORD PTR DS:[ESI]
00230584  66:33D0         XOR DX,AX
00230587 v74 09           JE SHORT 00230592
00230589  81EE 01100000   SUB ESI,1001
0023058F  4E              DEC ESI
00230590 ^EB EC           JMP SHORT 0023057E
00230592  8D5E FE         LEA EBX,DWORD PTR DS:[ESI-2]
00230595  84FF            TEST BH,BH
00230597 ^75 F0           JNZ SHORT 00230589
00230599  8B76 3A         MOV ESI,DWORD PTR DS:[ESI+3A]
0023059C  66:BA 5045      MOV DX,4550
002305A0  8D341E          LEA ESI,DWORD PTR DS:[ESI+EBX]
002305A3  66:AD           LODS WORD PTR DS:[ESI]
SS:[0023066C]=00000000
ESI=00240014
```

```
Registers (3DNow!)
EAX 000000D8
ECX 0001ADEC
EDX 165ABB07
EBX 00400000 ac0d0cf7.00400000
ESP 0018FF1C
EBP 00230608
ESI 00240014
EDI 00240014

EIP 00230554

C 0  ES 002B 32bit 0(FFFFFFFF)
P 1  CS 0023 32bit 0(FFFFFFFF)
A 1  SS 002B 32bit 0(FFFFFFFF)
Z 0  DS 002B 32bit 0(FFFFFFFF)
S 0  FS 0053 32bit 7EFDD000(FFF)
T 0  GS 002B 32bit 0(FFFFFFFF)
D 0
O 0  LastErr ERROR_SUCCESS (00000000)

EFL 00000216 (NO,NB,NE,A,NS,PE,GE,G)

MM0          0.0,          0.0
MM1          0.0,          0.0
MM2          0.0,          0.0
MM3          0.0,          0.0
MM4          0.0,          0.0
MM5          0.0,          0.0
MM6          0.0,          0.0
MM7          0.0,          0.0
```

```
Address   Hex dump                  ASCII
00240000  4D 5A 50 00 02 00 00 00   MZP.☻...
00240008  04 00 0F 00 FF FF 00 00   ♦.☼.  ..
00240010  B8 00 00 00 B5 89 1B 80   ©...Áë←Ç
00240018  63 BB 74 16 D9 89 35 80   c╗t▬┘ë5Ç
00240020  7F BB 74 16 DD 89 35 80   ⌂╗t▬¦ë5Ç
00240028  7B BB 74 16 E1 89 35 80   {╗t▬ßë5Ç
00240030  57 BB 74 16 05 8A 35 80   W╗t▬♣è5Ç
00240038  B3 B8 74 16 A9 8B 35 80   │©t▬®ï5Ç
00240040  D9 AE 74 18 5E 50 3F 73   ┘«t↑^P?s
00240048  DD 3A 00 29 10 6E D1 2F   ¦:.)►nÐ/
00240050  D2 E4 7B 35 68 66 AF 32   Êõ{5hf»2
00240058  6D EA 51 36 2B 65 86 3F   mÛQ6+eå?
00240060  3D B7 A9 32 FB 1B 5E 37   =À®2¹←^7
00240068  C1 A3 8D 0D 94 24 EF F2   ┴úì.ö$´=
00240070  5B A0 63 B3 CE 98 C6 7C   [ác│╬ÿã|
00240078  6C AA 87 EA CA 98 C6 7C   l¬çÛ╩ÿã|
00240080  C0 EC 87 EA AA DC CC 7C   └ýçÛ¬▄╠|
00240088  79 08 4E 31 DF 3A 0F A7   y◘N1▀:☼º
00240090  7D 08 4E 31 FB 3B 9D 24   }◘N1¹;Ø$
00240098  B4 0F DE AB 22 BD 9E 3D   ┤☼Ì½"¢×=
002400A0  98 D5 DF AB 3E E7 9E 3D   ÿı▀½>þ×=
002400A8  74 44 E1 AB E2 66 A0 3D   tDß½Ôfá=
```

```
0018FF1C  0025AE00
0018FF20  00240000  ASCII "MZP"
0018FF24  00230608  RETURN to 00230608
0018FF28  0018FF3C
0018FF2C  00400000  ac0d0cf7.00400000
0018FF30  0008E3C8
0018FF34  00000000
0018FF38  00240000  ASCII "MZP"
0018FF3C  002300A4  RETURN to 002300A4 from 00230519
0018FF40  00405FA7  RETURN to ac0d0cf7.00405FA7
0018FF44  0018FF88
0018FF48  75E549D7  kernel32.LoadLibraryA
0018FF4C  00000000
0018FF50  00000000
0018FF54  00000000
0018FF58  00000000
0018FF5C  00000000
0018FF60  00000000
0018FF64  00000000
0018FF68  00000000
0018FF6C  00000000
0018FF70  00000000
0018FF74  00000000
```

Here is the decryption loop code:

```
002B0554   3B75 64        CMP ESI,DWORD PTR SS:[EBP+64]
002B0557   75 0D          JNZ SHORT 002B0566
002B0559   0375 68        ADD ESI,DWORD PTR SS:[EBP+68]
002B055C   037D 68        ADD EDI,DWORD PTR SS:[EBP+68]
002B055F   2B4D 68        SUB ECX,DWORD PTR SS:[EBP+68]
```

```
002B0562   85C9            TEST ECX,ECX
002B0564   74 13          JE SHORT 002B0579
002B0566   AD             LODS DWORD PTR DS:[ESI]
002B0567   50             PUSH EAX
002B0568   83E8 0A         SUB EAX,0A
002B056B   35 AC324196      XOR EAX,964132AC
002B0570   2BC2           SUB EAX,EDX
002B0572   5A            POP EDX
002B0573   AB            STOS DWORD PTR ES:[EDI]
002B0574   83E9 03         SUB ECX,3
002B0577  ^E2 DB           LOOPD SHORT 002B0554
002B0579   61              POPAD
```

It decrypt’s an executable file SHA 1: bb5d9b8aee93ef92cea52849b17f57bb4e3c0306

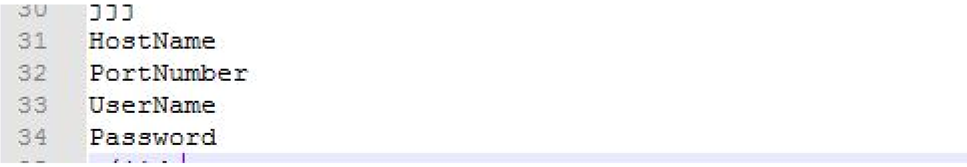
Behavior of the base file is suspicious itself, which is dropping another executable into the memory dump.

FareIt Dynamic analysis:

I have executed the file into safe environment and found some results:

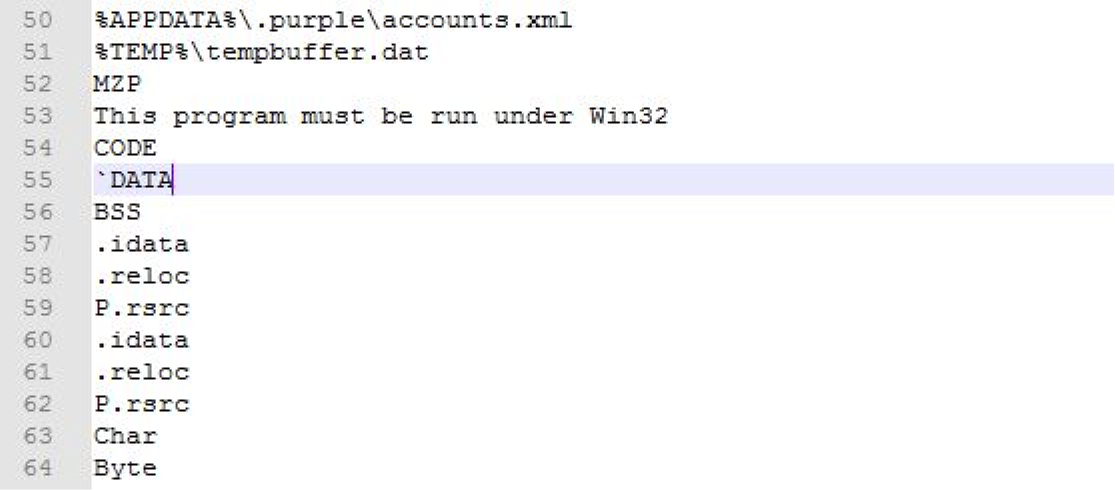
After execution I have gathered data of file’s Properties -> Strings -> strings and found some suspicious looking string form there:

- String Related to Passwords, Hostname



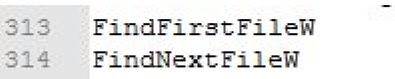
```
]]]
HostName
PortNumber
UserName
Password
```

- Strings related to embedding a executable file in tempbuffer.dat



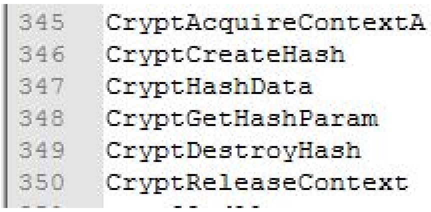
```
%APPDATA%\.purple\accounts.xml
%TEMP%\tempbuffer.dat
MZP
This program must be run under Win32
CODE
`DATA
BSS
.idata
.reloc
P.rsrc
.idata
.reloc
P.rsrc
Char
Byte
```

- Found Functions related to Find first file, Excessive number of FindFirstFile calls (suspicious)



```
FindFirstFileW
FindNextFileW
```

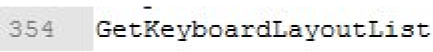
- Found functions related to Hashing. (suspicious)



```
CryptAcquireContextA
CryptCreateHash
CryptHashData
CryptGetHashParam
CryptDestroyHash
CryptReleaseContext
```

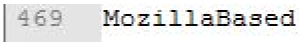
The CryptCreateHash function initiates the hashing of a stream of data.

- Function to get keyboard layout.

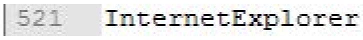


```
GetKeyboardLayoutList
```

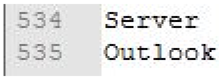
- Having Strings related to Browsers, email applications, Chat applications.



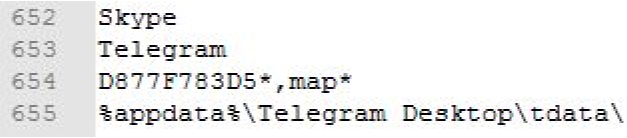
```
MozillaBased
```



```
InternetExplorer
```

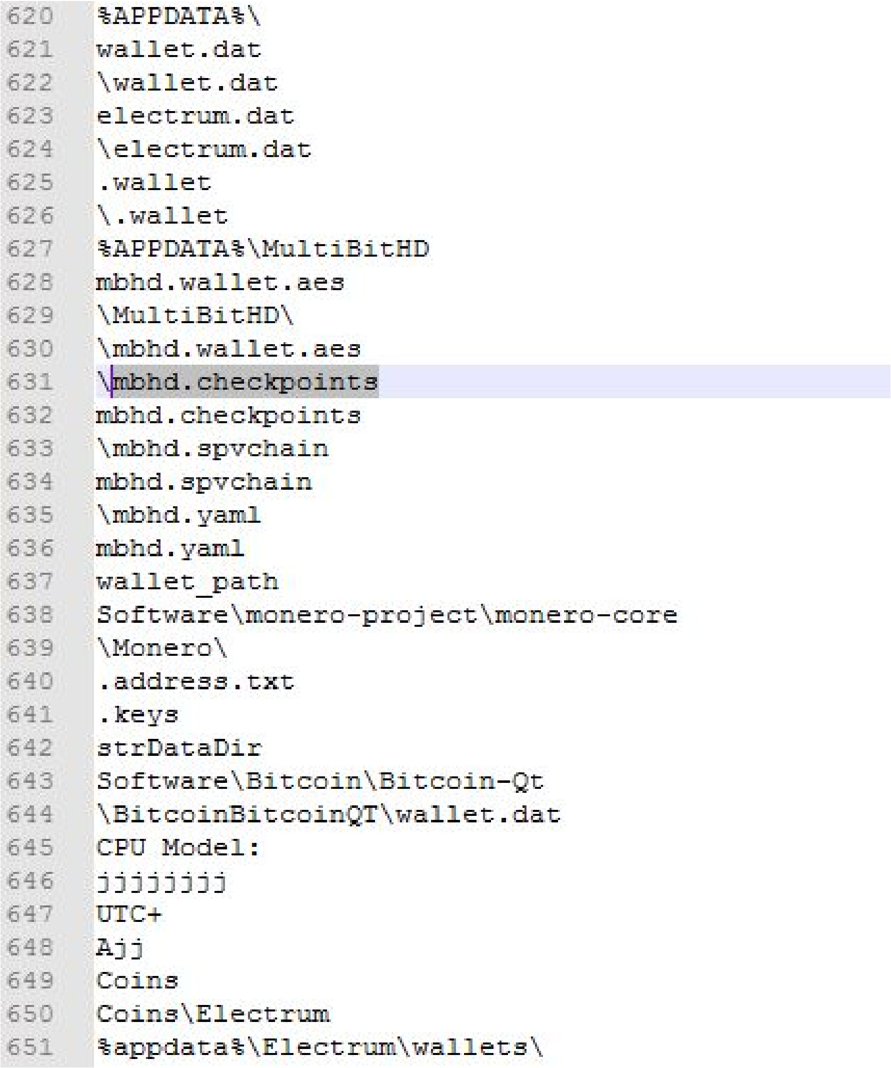


```
Server
Outlook
```



```
Skype
Telegram
D877F783D5*,map*
%appdata%\Telegram Desktop\tdata\
```

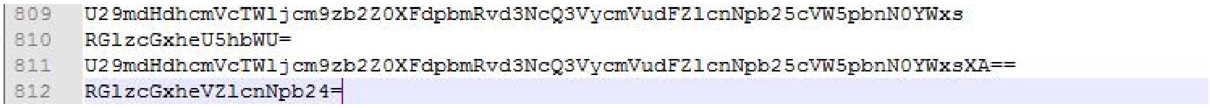
- Having Strings Related to bit coin wallet related keywords.



```
%APPDATA%\
wallet.dat
\wallet.dat
electrum.dat
\electrum.dat
.wallet
\.wallet
%APPDATA%\MultiBitHD
mbhd.wallet.aes
\MultiBitHD\
\mbhd.wallet.aes
\mbhd.checkpoints
mbhd.checkpoints
\mbhd.spvchain
mbhd.spvchain
\mbhd.yaml
mbhd.yaml
wallet_path
Software\monero-project\monero-core
\Monero\
.address.txt
.keys
strDataDir
Software\Bitcoin\Bitcoin-Qt
\BitcoinBitcoinQT\wallet.dat
CPU Model:
jjjjjjjj
UTC+
Ajj
Coins
Coins\Electrum
%appdata%\Electrum\wallets\
```

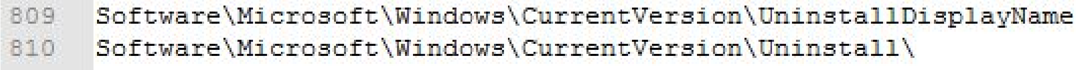
- Having some base64 encoded strings:

Before decryption



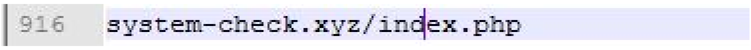
```
U29mdHdhcmVcTWljcm9zb2Z0XFdpbmRvd3NcQ3VycmVudFZlcnNpb25cVW5pbnN0YWxs
RGlzcGxheU5hbWU=
U29mdHdhcmVcTWljcm9zb2Z0XFdpbmRvd3NcQ3VycmVudFZlcnNpb25cVW5pbnN0YWxsXA==
RGlzcGxheVZlcnNpb24=
```

After decryption



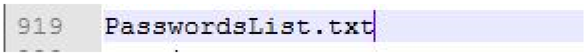
```
Software\Microsoft\Windows\CurrentVersion\UninstallDisplayName
Software\Microsoft\Windows\CurrentVersion\Uninstall\
```

- Got a suspicious URI(which is malware repo categorized) (malicious connection)



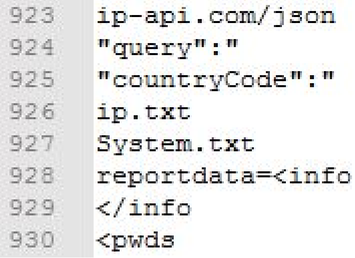
```
system-check.xyz/index.php
```

- Probably saving all credentials to below file.



```
PasswordsList.txt
```

- Probably creating a JSON object to post data through the above URI



```
ip-api.com/json
"query":"
"countryCode":"
ip.txt
System.txt
reportdata=<info
</info
<pwds
```

From the above Strings, it does have a credential stealing + bitcoin related data stealing properties.
Those are same as Fareit malware family.

Because of its anti-analysis feature I am not able to execute the malware to its full potential.
But above analysis is enough to prove sample as a Fareit malware family.