Reverse Engineering

Winlab 02

Timo Lehosvuo, M3426

Report
Reverse Engineering, Marko Silokunnas
20.3.2021
ICT

jamk | Jyväskylän ammattikorkeakoulu
University of Applied Sciences

Sisällys

1 Winlab 02

First, I opened the “winlab02” using PEview and found some interesting text from the “.data” section that gave me the idea that this is ransomware that most likely locks the user’s files/directories:

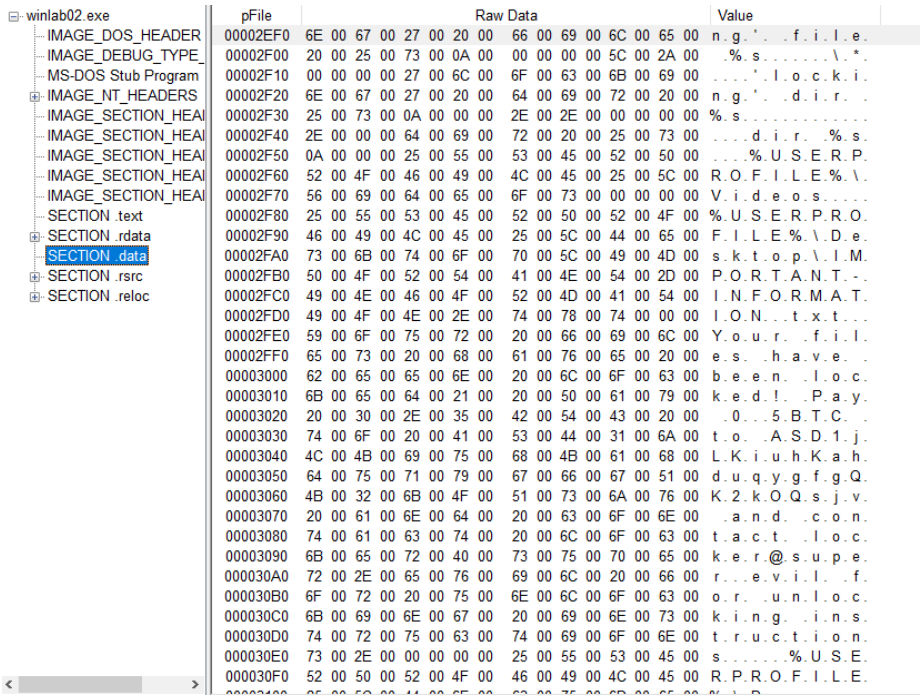


Figure 1: PEview

I took a snapshot of my virtual machine and executed the “winlab02” file. It created a file on my desktop, and it appears that the malware is indeed a ransomware malware:

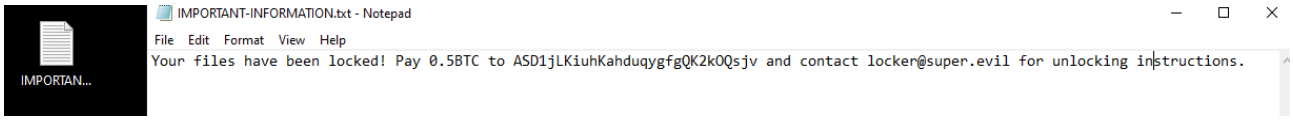
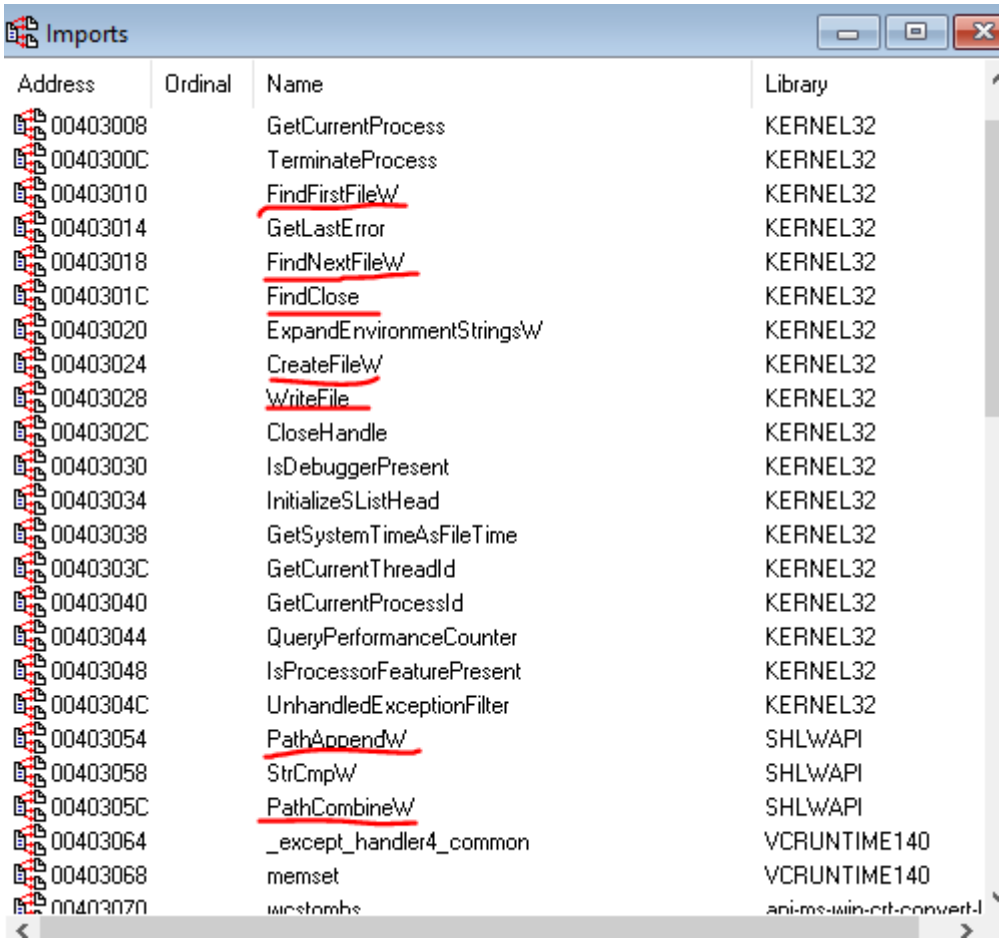


Figure 2: Ransomware

I started looking for proof of this in the code using IDA and I opened the imports and found some interesting functions:



Imports

Address	Ordinal	Name	Library
00403008		GetCurrentProcess	KERNEL32
0040300C		TerminateProcess	KERNEL32
00403010		FindFirstFileW	KERNEL32
00403014		GetLastError	KERNEL32
00403018		FindNextFileW	KERNEL32
0040301C		FindClose	KERNEL32
00403020		ExpandEnvironmentStringsW	KERNEL32
00403024		CreateFileW	KERNEL32
00403028		WriteFile	KERNEL32
0040302C		CloseHandle	KERNEL32
00403030		IsDebuggerPresent	KERNEL32
00403034		InitializeSListHead	KERNEL32
00403038		GetSystemTimeAsFileTime	KERNEL32
0040303C		GetCurrentThreadId	KERNEL32
00403040		GetCurrentProcessId	KERNEL32
00403044		QueryPerformanceCounter	KERNEL32
00403048		IsProcessorFeaturePresent	KERNEL32
0040304C		UnhandledExceptionFilter	KERNEL32
00403054		PathAppendW	SHLWAPI
00403058		StrCmpW	SHLWAPI
0040305C		PathCombineW	SHLWAPI
00403064		_except_handler4_common	VCRUNTIME140
00403068		memset	VCRUNTIME140
00403070		wcstombs	api-ms-win-crt-convert-l

Figure 3: IDA imports

The find file functions were interesting since it indicates that the malware searches for files which it does since its locking certain file/directories. Looking further down the code I found clear indicator of locking directories:

```
loc_401E53:
lea     ecx, [ebp+var_48]
mov     [ebp+var_2A8], ecx
mov     edx, [ebp+var_2A0]
mov     [ebp+var_2A4], edx
mov     [ebp+var_268], offset aUserprofileDoc ; "%USERPROFILE%\\Documents"
mov     [ebp+var_264], offset aUserprofilePic ; "%USERPROFILE%\\Pictures"
mov     [ebp+var_260], offset aUserprofileMus ; "%USERPROFILE%\\Music"
mov     [ebp+var_25C], offset aUserprofileVid ; "%USERPROFILE%\\Videos"
mov     [ebp+var_258], offset aUserprofileDow ; "%USERPROFILE%\\Downloads"
mov     [ebp+var_254], 0
lea     eax, [ebp+var_268]
mov     [ebp+var_280], eax
jmp     short loc_401EC1
```

Figure 4: Directories to lock

```
call    sub_4014D0
push    offset asc_40410C ; "\\*"
push    260
lea     edx, [ebp+FileName]
push    edx
call    sub_401530
mov     eax, [ebp+lpszDir]
push    eax
push    offset aLockingDirS ; "'locking' dir %s\n"
call    sub_401F60
```

Figure 5: locking dir

The malware looks for files in the directory and locks them using some kind of algorithm:

```
push    edx
push    260
lea     eax, [ebp+pMore]
push    eax
call    sub_401530
push    0               ; lpszFile
mov     ecx, [ebp+lpszDir]
push    ecx             ; lpszDir
lea     edx, [ebp+FileName]
push    edx             ; szDest
call    ds:PathCombineW
lea     eax, [ebp+pMore]
push    eax             ; pMore
lea     ecx, [ebp+FileName]
push    ecx             ; pszPath
call    ds:PathAppendW
lea     edx, [ebp+FileName]
push    edx
mov     eax, [ebp+var_974]
mov     ecx, [eax]
push    ecx
push    offset aLookingForSFil ; "Looking for %s files (%s)\n"
call    sub_401F60
add     esp, 12
lea     edx, [ebp+FindFileData]
push    edx             ; lpFindFileData
lea     eax, [ebp+FileName]
push    eax             ; lpFileName
call    ds:FindFirstFileW
mov     [ebp+hFindFile], eax
cmp     [ebp+hFindFile], 0FFFFFFFFh
jnz     short loc_4018D6
```

Figure 6: Looking for files

```
lea     ecx, [ebp+pszPath]
push    ecx             ; pszPath
call    ds:PathAppendW
mov     edx, [ebp+arg_4]
push    edx
mov     eax, [ebp+arg_0]
push    eax
lea     ecx, [ebp+var_720]
push    ecx
call    sub_401330
add     esp, 12
lea     edx, [ebp+pszPath]
push    edx
push    offset aLockingFileS ; "  'locking' file %s\n"
call    sub_401F60
add     esp, 8
push    0               ; size_t
lea     eax, [ebp+pszPath]
push    eax             ; wchar_t *
push    0               ; char *
call    ds:wcstombs
add     esp, 0Ch
mov     [ebp+var_980], eax
mov     ecx, [ebp+var_980]
add     ecx, 1
push    ecx             ; size_t
call    ds:malloc
add     esp, 4
mov     [ebp+var_978], eax
mov     edx, [ebp+var_980]
add     edx, 1
push    edx             ; size_t
lea     eax, [ebp+pszPath]
push    eax             ; wchar_t *
mov     ecx, [ebp+var_978]
push    ecx             ; char *
```

Figure 7: Locking files

Once the malware has found and lock all the files it creates a text file on the victim's desktop:

```
mov     ebp, esp
sub     esp, 21Ch
mov     eax, dword_404004
xor     eax, ebp
mov     [ebp+var_4], eax
push    260             ; nSize
lea     eax, [ebp+FileName]
push    eax             ; lpDst
push    offset Src      ; "%USERPROFILE%\\Desktop\\IMPORTANT-INFORMA"...
call    ds:ExpandEnvironmentStringsW
push    0               ; hTemplateFile
push    128             ; dwFlagsAndAttributes
push    2               ; dwCreationDisposition
push    0               ; lpSecurityAttributes
push    0               ; dwShareMode
push    1073741824      ; dwDesiredAccess
lea     ecx, [ebp+FileName]
push    ecx             ; lpFileName
call    ds:CreateFileW
mov     [ebp+hObject], eax
mov     [ebp+NumberOfBytesWritten], 0
mov     [ebp+var_218], offset aYourFilesHaveB ; "Your files have been locked! Pay 0.5BTC"...
lea     edx, [ebp+nNumberOfBytesToWrite]
push    edx
push    400h
mov     eax, [ebp+var_218]
push    eax
call    sub_401590
```

Figure 8: IMPORTANT-INFORMATION file

And writes to the file the text: “Your files have been locked! Pay 0.5BTC to ASD1jLKiuhKahduqyqfgQk2k0Qsjv and contact locker@super.evil for unlocking instructions.”:

```
mov     [ebp+NumberOfBytesWritten], 0
mov     [ebp+var_218], offset aYourFilesHaveB ; "Your files have been locked! Pay 0.5BTC"...
lea     edx, [ebp+nNumberOfBytesToWrite]
push    edx
push    400h
mov     eax, [ebp+var_218]
push    eax
call    sub_401590

loc_401B9B:
mov     ecx, [ebp+NumberOfBytesWritten]
cmp     ecx, [ebp+nNumberOfBytesToWrite]
jnb     short loc_401BD8

push    0               ; lpOverlapped
lea     edx, [ebp+NumberOfBytesWritten]
push    edx             ; lpNumberOfBytesWritten
mov     eax, [ebp+nNumberOfBytesToWrite]
push    eax             ; nNumberOfBytesToWrite
mov     ecx, [ebp+NumberOfBytesWritten]
mov     edx, [ebp+var_218]
lea     eax, [edx+ecx*2]
push    eax             ; lpBuffer
mov     ecx, [ebp+hObject]
push    ecx             ; hFile
call    ds:WriteFile
jmp     short loc_401B9B

loc_401BD8:
mov     edx, [ebp+hObject]
push    edx             ; hObject
call    ds:CloseHandle
mov     ecx, [ebp+var_4]
xor     ecx, ebp
call    sub_401FAA
mov     esp, ebp
pop     ebp
retn
sub_401B20 endp
```

Figure 9: Write to file

I also found a long list on numbers and I thought that it might be the key for the encryption that which the malware locks the files/directories:

```
mov     byte ptr [ebp+var_48], 170
mov     byte ptr [ebp+var_48+1], 197
mov     byte ptr [ebp+var_48+2], 215
mov     byte ptr [ebp+var_48+3], 156
mov     [ebp+var_44], 254
mov     [ebp+var_43], 196
mov     [ebp+var_42], 205
mov     [ebp+var_41], 207
mov     [ebp+var_40], 191
mov     [ebp+var_3F], 141
mov     [ebp+var_3E], 213
mov     [ebp+var_3D], 138
mov     [ebp+var_3C], 167
mov     [ebp+var_3B], 131
mov     [ebp+var_3A], 158
mov     [ebp+var_39], 155
mov     [ebp+var_38], 182
mov     [ebp+var_37], 0C8h
mov     [ebp+var_36], 0CCh
mov     [ebp+var_35], 8Ah
mov     [ebp+var_34], 0FEh
mov     [ebp+var_33], 0CCh
mov     [ebp+var_32], 0CCh
mov     [ebp+var_31], 8Ah
mov     [ebp+var_30], 0FEh
mov     [ebp+var_2F], 0C0h
mov     [ebp+var_2E], 0DFh
mov     [ebp+var_2D], 81h
mov     [ebp+var_2C], 0A7h
mov     [ebp+var_2B], 8Dh
mov     [ebp+var_2A], 0D2h
mov     [ebp+var_29], 86h
mov     [ebp+var_28], 0B5h
mov     [ebp+var_27], 0C8h
mov     [ebp+var_26], 9Eh
mov     [ebp+var_25], 86h
mov     [ebp+var_24], 0AAh
```

Figure 10: Key for encryption

Also found the possible encryptor but I am not sure how it works in detail:

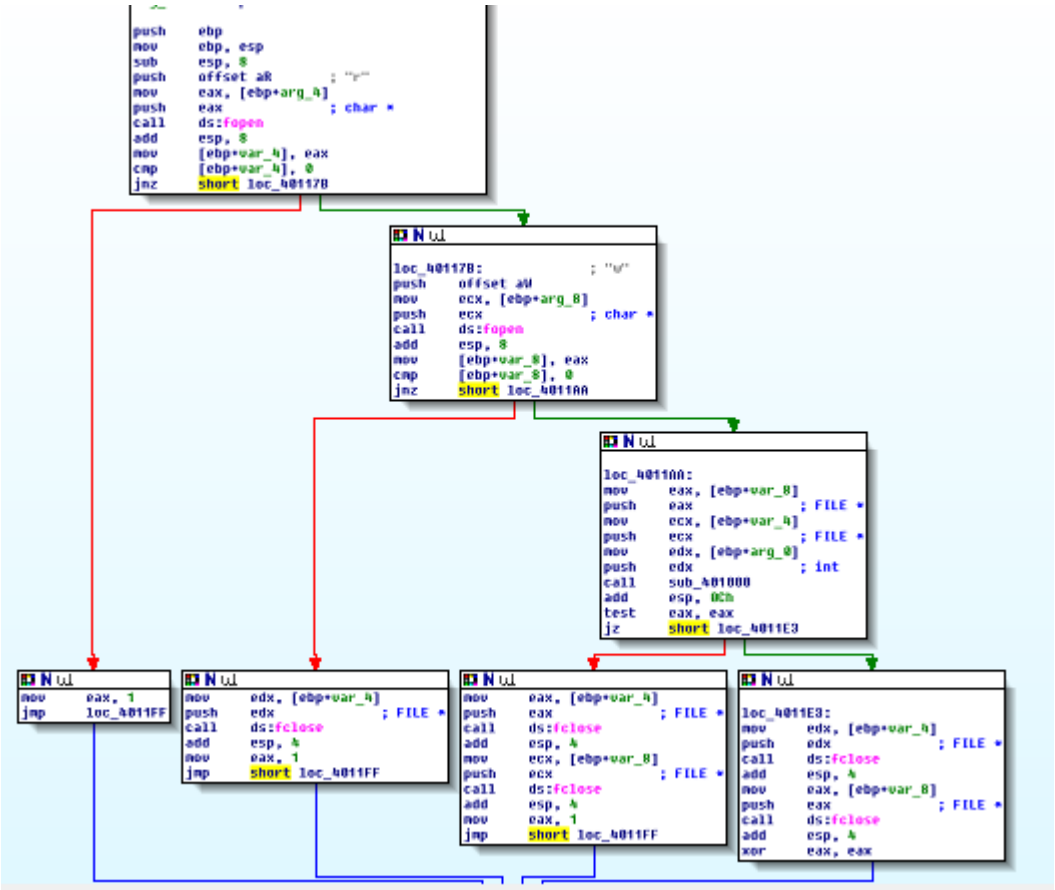


Figure 11: Possible encryptor

This file also has a trap to debugger which is kind of odd if this were to be a legitime program:

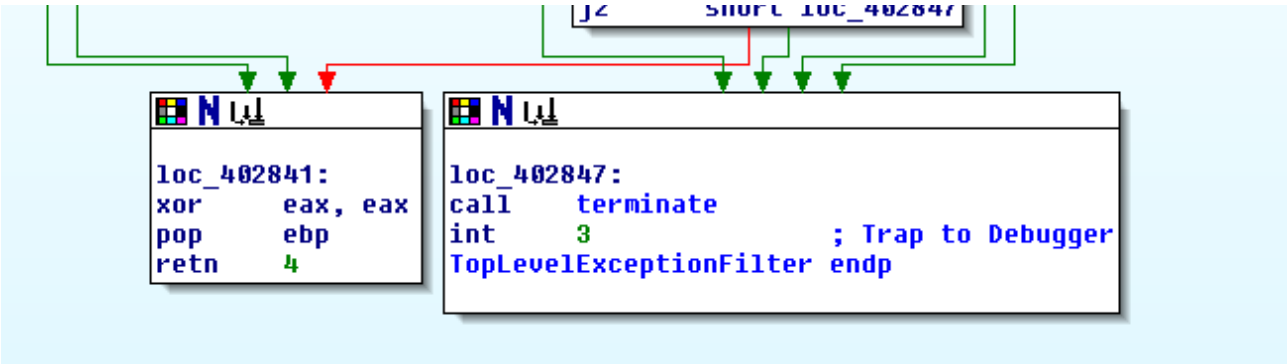


Figure 12: Trap to debugger

2 Summary

This is a ransomware that locks the victim's "Documents", "Pictures", "Music", "Videos" and "Downloads" directories and ask for 0.5BTC (Bitcoin) for unlocking the files/directories

3 Indicators of compromise

- Executed the file and it locked my files
- PEview shows some alarming text on the .data section
- Code contains alarming functions (imports)
- Code contains cleartext indicators what the malware does like "locking dir"
- Simply just solving what the code does

4 Timetable

Report:	0.5 H
Solving the lab:	2 H
Total:	2.5 H