

악성코드 분석 결과보고

dropper.exe 분석 보고서

김 진 환

24.07.24

목차

1. 개요

1.1 분석 환경

1.2 분석 샘플

2. 기초 분석

2.1 VirusTotal

3. 기초 정적 분석

3.1 HashCalc

3.2 Exeinfo PE

3.3 PE view

3.4 Dependency Walker

3.5 Strings

4. 기초 동적 분석

4.1 SysAnalyzer

5. 고급 정적 분석

5.1 IDA pro

6. 고급 동적 분석

6.1 OllyDbg

7. 분석 결론 및 대응 방안

7.1 악성파일의 분석 결론

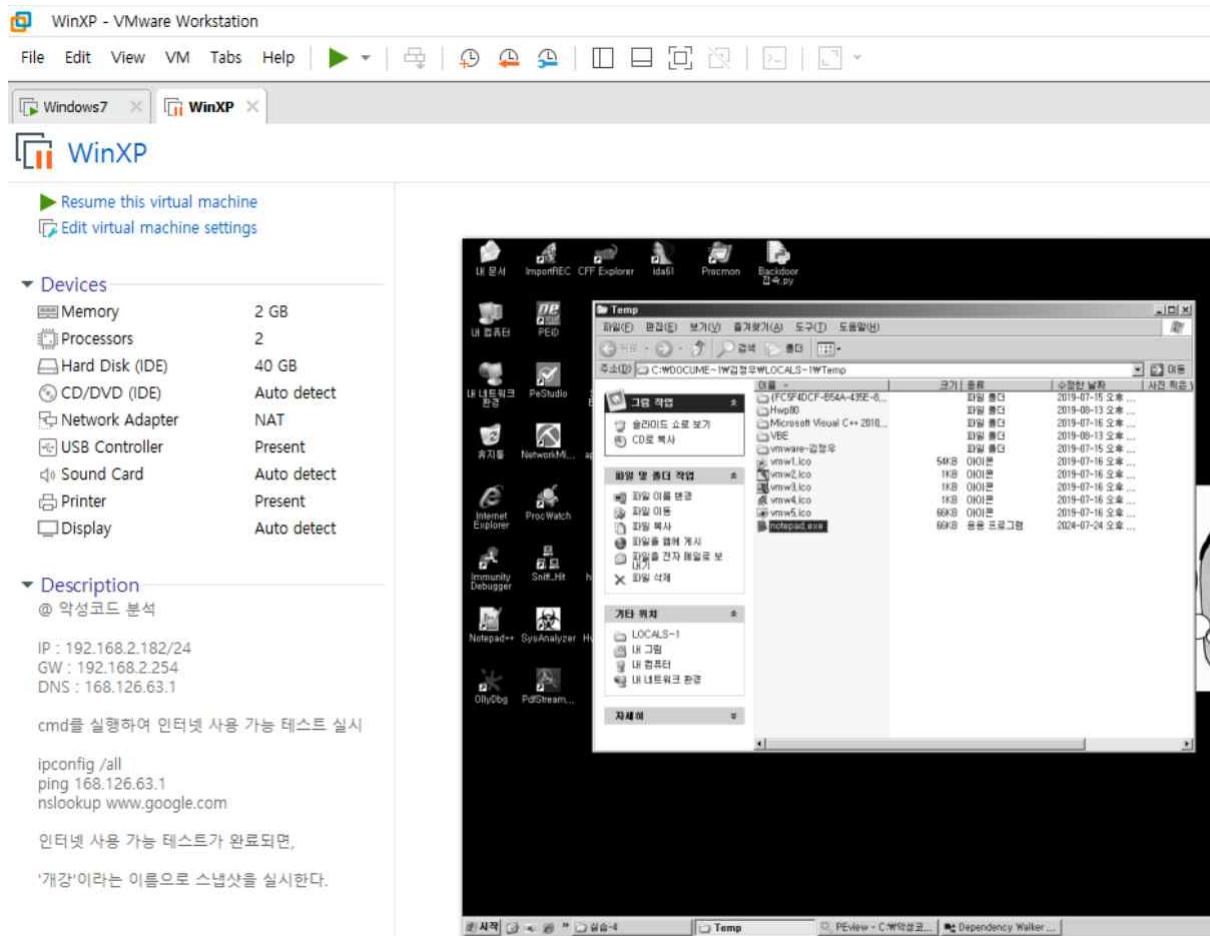
7.2 대응 방안

1. 개요

1.1 분석 환경

가상 환경	VMware Workstation Pro
윈도우 버전	Window XP(32bit)
분석 도구	hashcalc, exeinfope, PEview, Dependency Walker, strings, PeStudio, SysAnalyzer, IDA Pro, OllyDbg

<표 1>



<그림 1> VMware Workstation Pro 가상환경

1.2 분석 샘플

파일 이름	dropper.exe
파일 형식	.exe 실행 파일
기초 분석 결과	AhnLab-V3 : Malware/Gen.Generic.C3355284 Avast : Win32:Malware-gen Antiy-AVL : Trojan/Win32.Swisyn

<표 2>

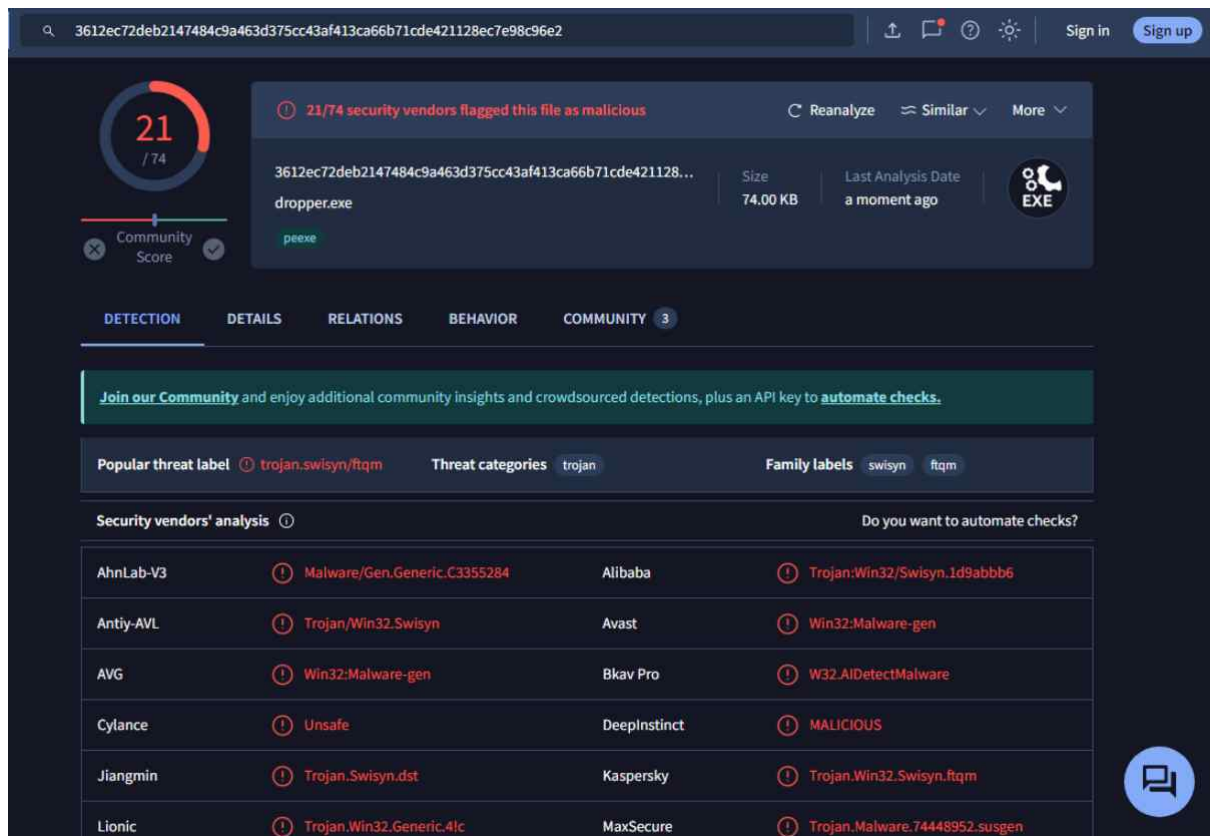


dropper.exe

<그림 2> dropper.exe 이미지

2. 기초분석

2.1 VirusTotal



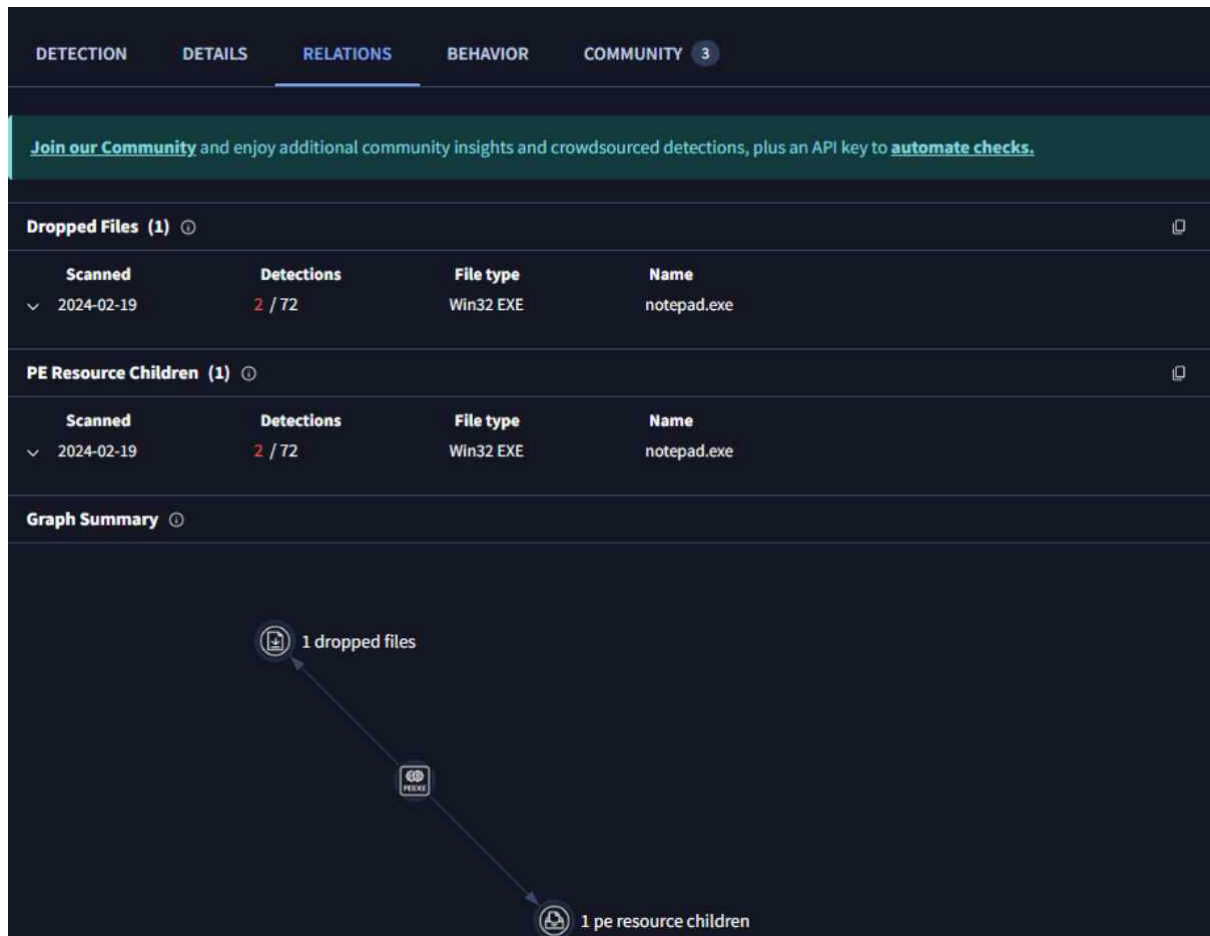
<그림 3> VirusTotal 기초분석 결과

74개의 백신 중 21개의 백신에서 downloader.exe 파일의 악성 여부를 발견 했습니다. AhnLab-V3 를 포함한 여러 백신들에서 MalwareX 이 발견되어 파일이 악성코드를 가지고 있음을 알 수 있으며 Trojan 이 발견되는 점을 보아 트로이목마 기능이 있는 악성 파일임을 확인했습니다.

Basic properties ⓘ	
MD5	5d67f2d24326c550898a42959379aaaf
SHA-1	46b0d740034eb02c4ffe512e8db132050cf7610b
SHA-256	3612ec72deb2147484c9a463d375cc43af413ca66b71cde421128ec7e98c96e2
Vhash	074046551d156az18nz2az1c1z
Authentihash	daac42c226cf99d200a8775025b12b174000a3bfef387fa1fc38976489a6dd89
Imphash	bb852d8de285cdb6afc8d7a5d81dec95
Rich PE header hash	6eaf77033ec3d8595257754146556da0
SSDEEP	1536:j3OowOnbNQKLjWdy1o5l0dJUEbooPRrKKRjMI8j:NNQKPWDyDI0dJltZrpRjMla
TLSH	T1AE735B09A386F0AAD451843012E69BA24F399E306E0B53CFB7707B1F9D316DEEB25305
File type	Win32 EXE executable windows win32 pe peexe
Magic	PE32 executable (GUI) Intel 80386, for MS Windows
TrID	InstallShield setup (34.2%) Win32 Executable MS Visual C++ (generic) (24.8%) Microsoft Visual C++ compiled execu...
DetectItEasy	PE32 Compiler: EP:Microsoft Visual C/C++ (2008-2010) [EXE32] Compiler: Microsoft Visual C/C++ (15.00.30729) [LTC...
Magika	PEBIN
File size	74.00 KB (75776 bytes)
History ⓘ	
Creation Time	2019-07-19 06:11:46 UTC
First Submission	2019-07-24 19:07:04 UTC
Last Submission	2024-07-24 07:16:57 UTC
Last Analysis	2024-07-24 07:17:03 UTC
Names ⓘ	
dropper.exe	

<그림 4> VirusTotal - Details

해당 악성 파일이 Windows 32bit 운영체제에서 실행되는 파일이며, Microsoft Visual C++ 을 사용하여 제작되고, 컴파일 된것을 확인 할 수 있습니다. 파일의 생성 날짜는 2019년 7월 19일 6시11분46초이며 최초 발견일은 2019년 7월 24일 19시7분4초 인것으로 확인 됩니다.



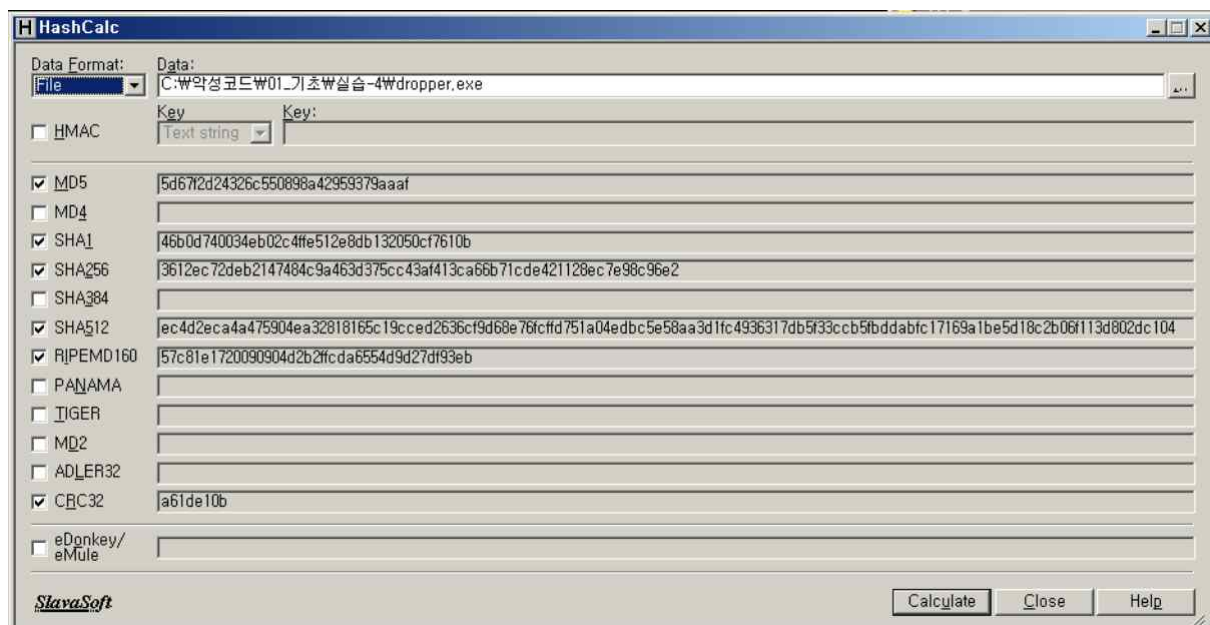
<그림 5> VirusTotal - RELATIONS

해당 파일은 Dropper 파일이며 notepad.exe 파일을 Drop 하는 것으로 보여집니다.

위와 같이 VirusTotal 에서 많은 정보를 얻을 수 있었습니다. 하지만 VirusTotal 의 정보가 항상 옳다고 볼 수 없기에 여러 프로그램을 추가적으로 사용하여 분석한 뒤 정보를 교차 검증하여 해당 파일에 대한 정보를 분석해낼 필요가 있습니다.

3. 기초 정적 분석

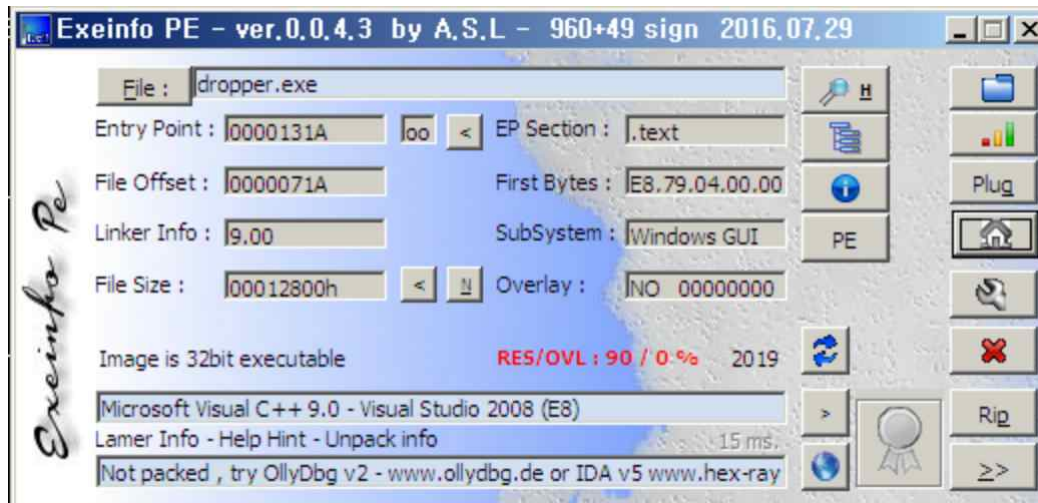
3.1 HashCalc



<그림 6> HashCalc

HashCalc 프로그램을 이용하여 dropper.exe 의 MD5, SHA256 등 해시값을 확인 할 수 있습니다.

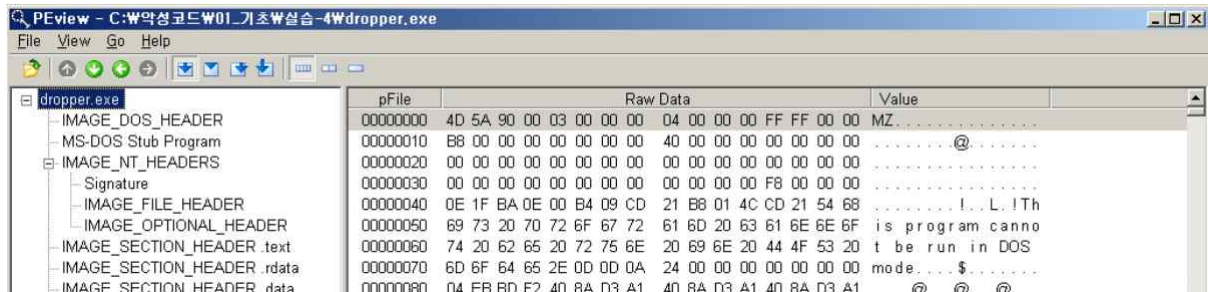
3.2 Exeinfo PE



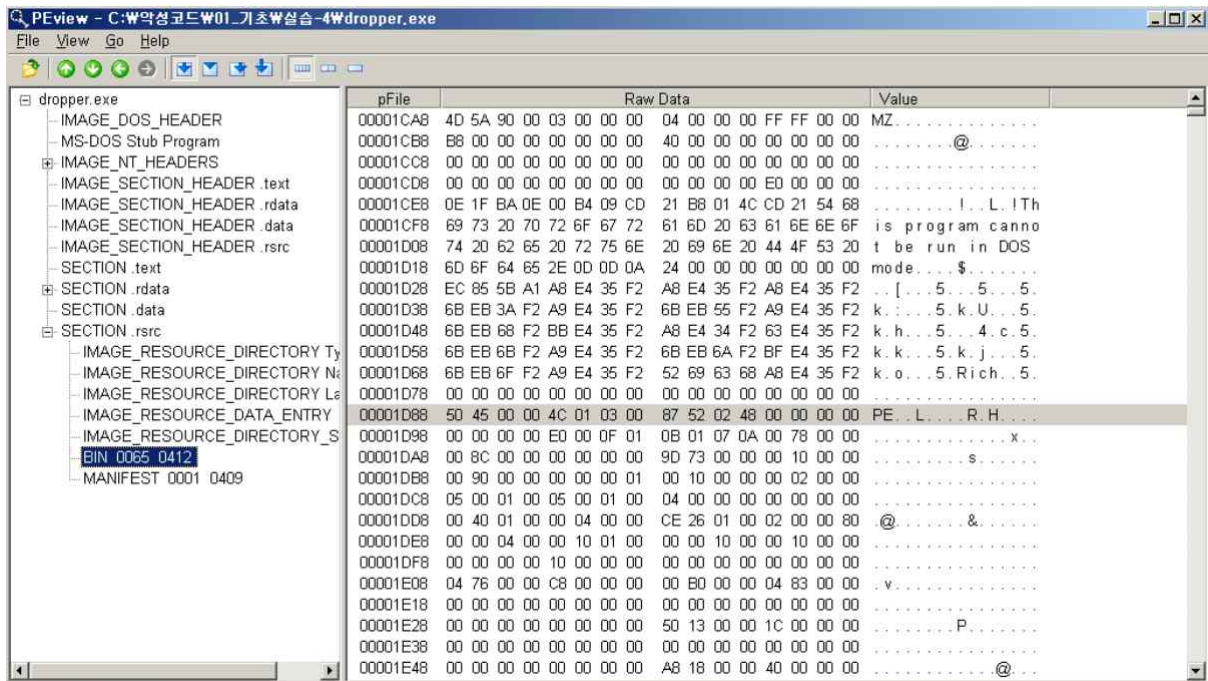
<그림 7> Exeinfo PE - ver.0.0.4.3

Exeinfo PE 분석 결과 Microsoft Visual C++ 9.0 - Visual Studio 로 만들어진 파일이며, 패킹은 되지 않음을 알 수 있었습니다. 추가적으로 OllyDbg , IDA v5 등 의 분석 도구를 제안 받음을 알 수 있습니다.

3.3 PE View



<그림 8> PE View - dropper.exe

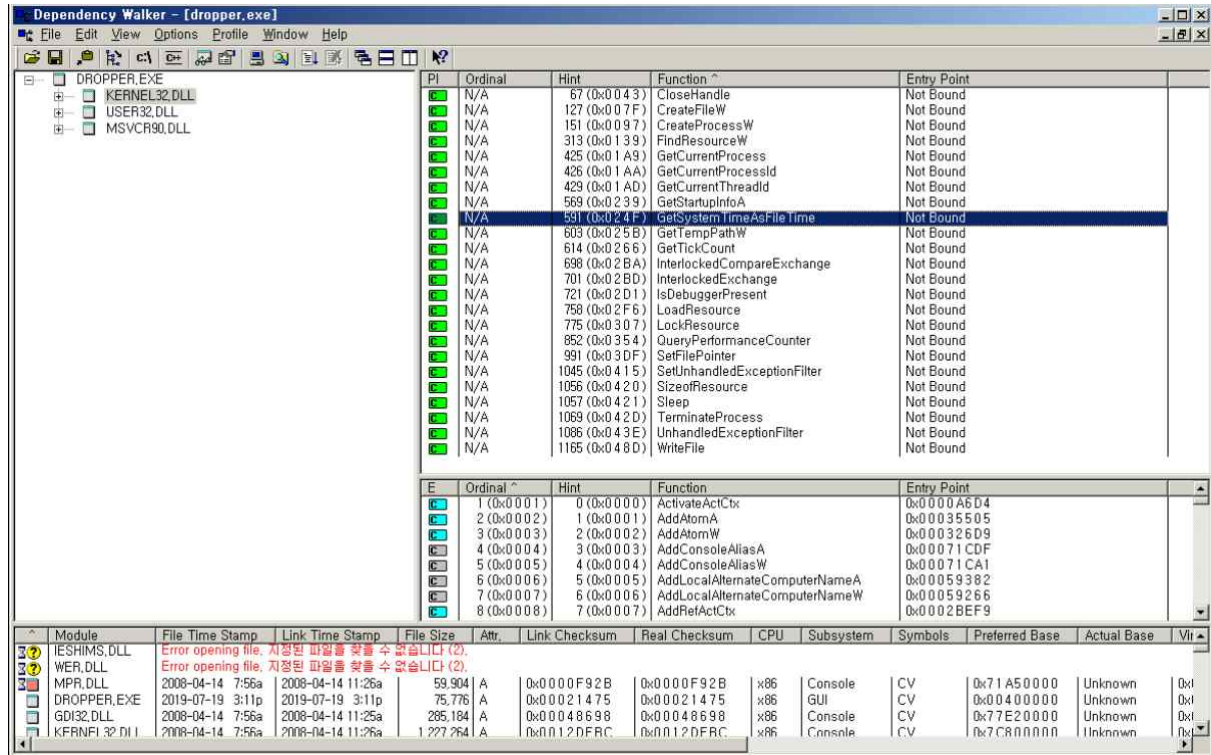


<그림 9> PE View - BIN 0065 0412

PE View 분석을 통해 해당 파일의 MZ 칸의 4D 5A 데이터 구조를 보아 .exe 실행 파일 인 점을 확인 가능하며 , This program cannot be run in DOS mode 문자열을 확인하여 DOS 모드에서는 실행 할 수 없는 것으로 보입니다. IMAGE_FILE_HEADER 탭을 통해 제작 날짜는 2019년 7월 19일로 확인 할 수 있으며, Body section 을 통해 IAT 정보 확인 결과 KERNEL32.dll , USER32.dll , MSVCRT90.dll 를 참조함 을 알 수 있습니다.

중요한 점으로 해당 파일 안에 BIN 탭을 확인하여 또다른 파일이 숨겨져 있음을 확인 했고 숨겨져 있는 파일 또한 위와 같이 4D 5A 데이터 구조로 보아 .exe 실행 파일인것 을 확인 했습니다.

3.4 Dependency Walker



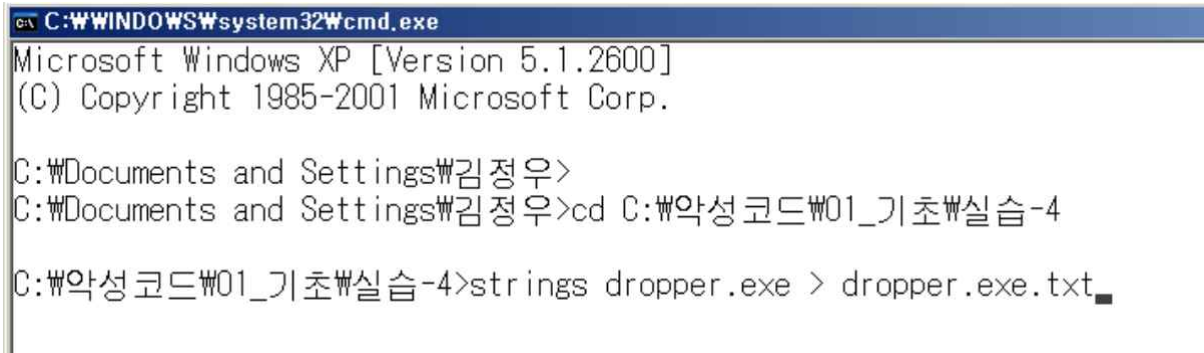
<그림 10> dependency Walker

Dependency Walker 를 통해 파일/디렉토리 관련 DLL를 확인 및 분석한 결과 해당 파일에서 사용하고 있는 함수를 확인 할 수 있었습니다.

'PE View' 를 이용하여 리소스 영역에 PE 파일이 있음을 확인 했었고, Dropper 로 의심되는 API (하단 정리) 들이 있고 ,

GetTempPathW 를 통해 Temp(임시파일) 의 경로를 찾으려 하며, Create File , FindResource , SizeofResource, CreateProcess, LoadResource, WriteFile 등 함수를 통해 해당 파일 안에 숨겨놓은 PE 파일을 시스템 프로세스에 불러오며 새로운 파일을 생성하여 Temp 경로에 저장하고, 실행 함 을 알 수 있었습니다.

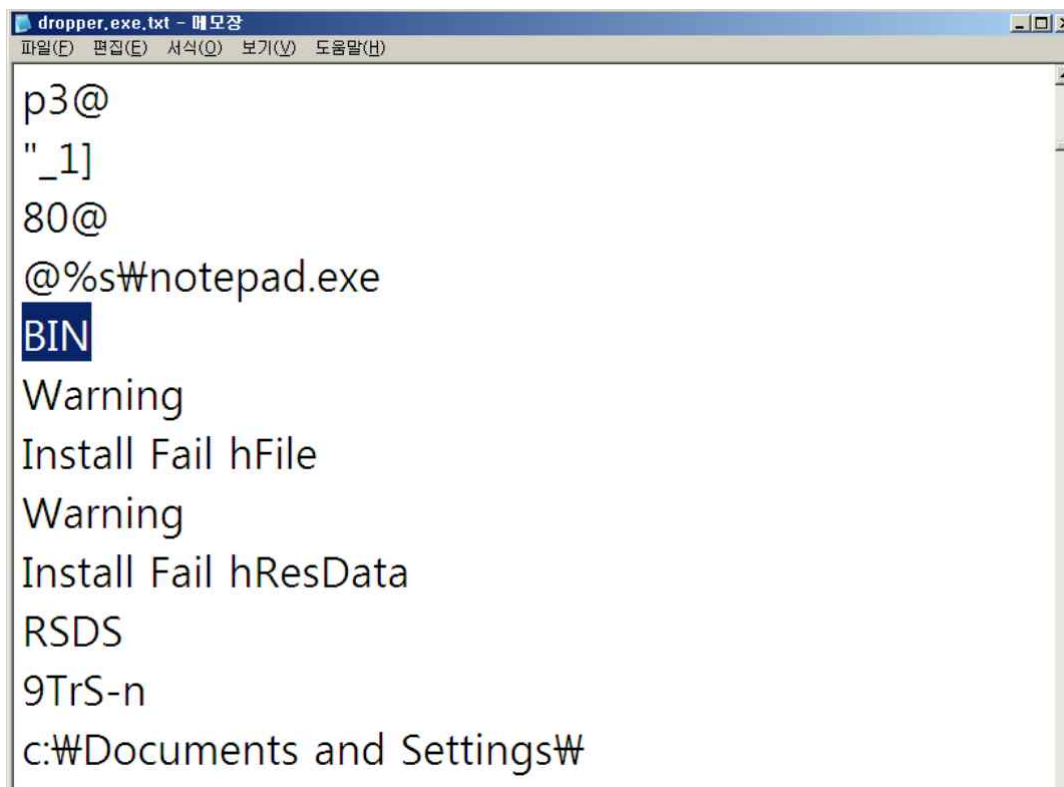
3.5 Strings



```
C:\WINDOWS\system32\cmd.exe
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings\김정우>
C:\Documents and Settings\김정우>cd C:\악성코드\01_기초\실습-4
C:\악성코드\01_기초\실습-4>strings dropper.exe > dropper.exe.txt
```

<그림 11> Strings - cmd



```
dropper.exe.txt - 메모장
파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)

p3@
"_1]
80@
@%s\notepad.exe
BIN
Warning
Install Fail hFile
Warning
Install Fail hResData
RSDS
9TrS-n
c:\Documents and Settings\
```

<그림 12> Strings - 메모장

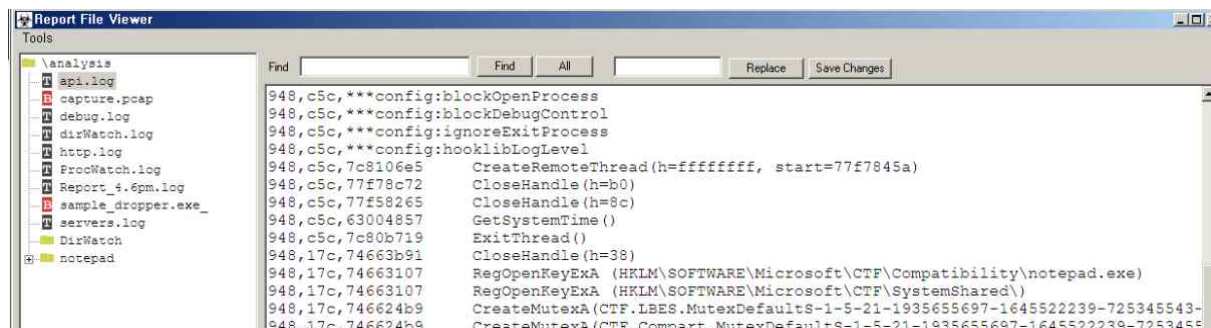
Strings 명령어를 통해 해당 파일의 문자열을 메모장에 저장하여 확인하고 PE View 프로그램에서 확인했던 BIN 타입을 검색 하니 해당 파일에 숨겨져 있는 .exe 파일이 notepad.exe 파일 이라는 것을 알아 낼 수 있었습니다.

4. 기초 동적 분석

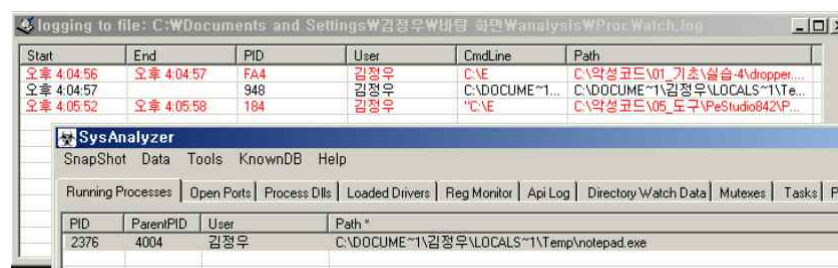
4.1 SysAnalyzer



<그림 13> SysAnalyzer 1



<그림 14> SysAnalyzer 2

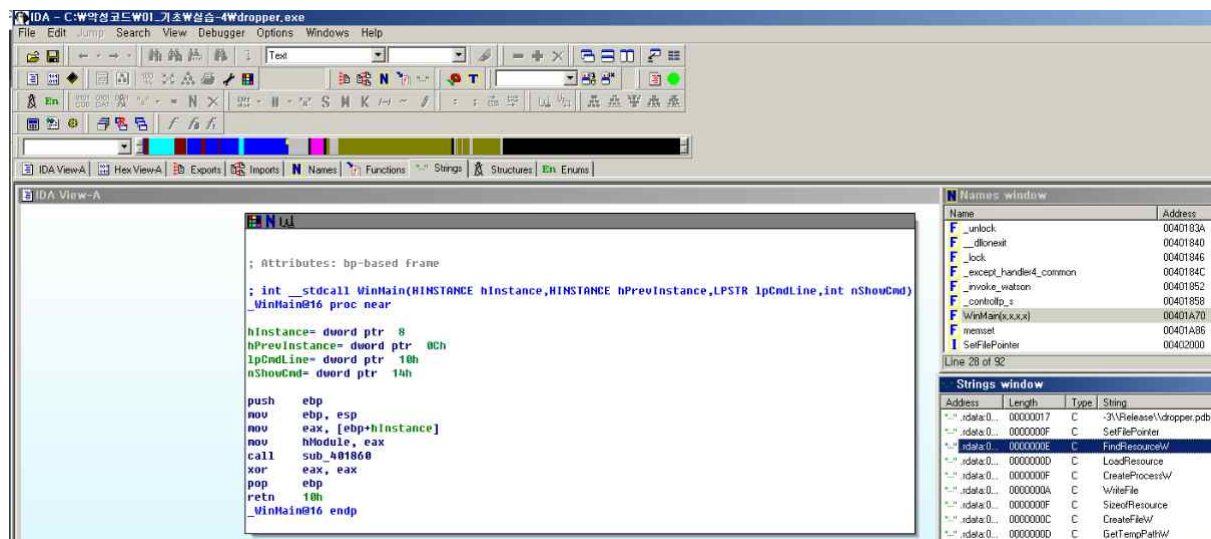


<그림 15> SysAnalyzer 3

SysAnalyzer 분석 결과 네트워크 관련된 설정이 없으므로, WireShark 분석은 필요 없음을 알 수 있으며, 또한 <그림15> 를 참조하여 PID 확인 결과 부모의 PID가 0xFA4(10진수 4004) 로 보이고 해당 파일로 인해 자식파일인 notepad.exe 가 실행 됨을 다시 확인 할 수 있었습니다.

5. 고급 정적 분석

5.1 IDA Pro



<그림 16> IDA Pro - main function

IDA Pro 프로그램 분석을 통해 메인 함수의 주소값 401A70 을 알아 낼 수 있으며, 해당 메인 함수가 주소값 401860에 저장되어있는 함수를 실행 시킴을 알 수 있습니다. 따라서 주소값 401860에 저장되어있는 함수에 대한 분석이 필요합니다.

```

nNumberOfBytesToWrite= dword ptr -42Ch
CommandLine= word ptr -428h
var_4= dword ptr -4

push    ebp
mov     ebp, esp
sub     esp, 680h
mov     eax, dword_403000
xor     eax, ebp
mov     [ebp+var_4], eax
mov     [ebp+hResInfo], 0
mov     [ebp+hResData], 0
mov     [ebp+lpBuffer], 0
mov     [ebp+hObject], 0
lea     eax, [ebp+Buffer]
push    eax                ; lpBuffer
push    105h               ; nBufferLength
call    ds:GetTempPathW
lea     ecx, [ebp+Buffer]
push    ecx
push    offset a$Notepad_exe ; "%s\\notepad.exe"
lea     edx, [ebp+CommandLine]
push    edx                ; LPWSTR
call    ds:wsprintfW
add     esp, 0Ch

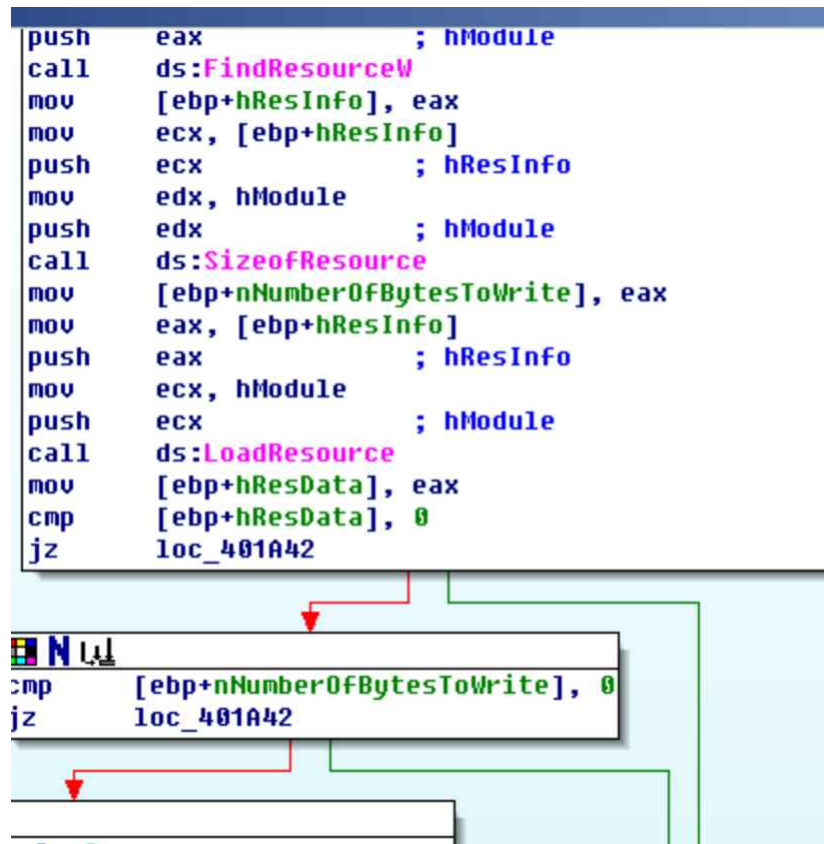
```

<그림 17> IDA Pro - 401860 function 1

주소값 401860의 함수에 접근하여 가장 먼저 GetTempPathW 와 wsprintfw API 를 확인 하였습니다. 그리고 각 API 들의 역할을 분석 합니다.

1. GetTempPathW 를 통해 Temp 디렉토리를 찾는다.
2. wsprintfw 를 사용하여 'C:\DOCUMENT~1\김정우\LOCALS~1\Temp\notepad.exe' 문자열을 생성한다.

이후 함수를 계속해서 확인 해 나갑니다.



<그림 18> IDA Pro - 401860 function 2

FindResourceW , SizeofResource , LoadResource 세가지의 API를 확인 할 수 있으며 따라서 해당 파일은 Dropper 파일 이라는것을 인지 할 수 있습니다.

각 API 들의 역할은

3. FindResourceW 를 통해 해당 파일 내부에 숨겨놓은 'BIN 101' PE 파일 찾기

4. SizeofResource 를 통해 'BIN 101' PE 파일의 크기 계산

5. LoadResource 를 통해 'BIN 101' PE 파일을 메모리에 로딩

을 알 수 있습니다.


```

push    40000000h        ; dwDesiredAccess
lea     eax, [ebp+CommandLine]
push    eax              ; lpFileName
call    ds:CreateFileW
mov     [ebp+hObject], eax
cmp     [ebp+hObject], 0
jz      loc_401A2C

```

<그림 19> IDA Pro - 401860 function 2

```

push    0                ; lpOverlapped
lea     edx, [ebp+NumberOfBytesWritten]
push    edx              ; lpNumberOfBytesWritten
mov     eax, [ebp+nNumberOfBytesToWrite]
push    eax              ; nNumberOfBytesToWrite
mov     ecx, [ebp+lpBuffer]
push    ecx              ; lpBuffer
mov     edx, [ebp+hObject]
push    edx              ; hFile
call    ds:WriteFile
mov     eax, [ebp+hObject]
push    eax              ; hObject
call    ds:CloseHandle
mov     [ebp+StartupInfo.cb], 0
push    40h              ; size_t
push    0                ; int
lea     ecx, [ebp+StartupInfo.lpReserved]
push    ecx              ; void *

```

<그림 20> IDA Pro - 401860 function 3

```

push    0                ; bInheritHandles
push    0                ; lpThreadAttributes
push    0                ; lpProcessAttributes
lea     ecx, [ebp+CommandLine]
push    ecx              ; lpCommandLine
push    0                ; lpApplicationName
call    ds:CreateProcessW
jmp     short loc_401A40

```

<그림 21> IDA Pro - 401860 function 4

이어서 중요 API 들을 확인하여 CreateFileW, WriteFile, CloseHandle, CreateProcessW 의 역할을 분석합니다.

6. CreateFileW 를 통해 경로를 찾아 놓았던 Temp 디렉토리에 0바이트 파일을 생성합니다.

7. WriteFile 을 통해 메모리에 로딩된 'BIN 101' PE 파일의 바이너리를 위에 생성해 두었던 0바이트 파일에 씁니다.

8. CloseHandle 을 실행하면 notepad.exe 파일이 완성됩니다.

9. CreateProcessW 를 통해 완성된 notepad.exe 파일을 실행 합니다.

위 분석을 통해 해당 파일은 dropper 파일 이며, 내부에 숨겨둔 notepad.exe 파일을 temp(임시파일) 경로에 생성 및 실행 하는 파일인 것을 알 수 있습니다.

6. 고급 동적 분석

6.1 OllyDbg

```

00401A6E CC      INT3
00401A6F CC      INT3
00401A70 55      PUSH EBP
00401A71 8BEC    MOV EBP,ESP
00401A73 8B45 08 MOV EAX,DWORD PTR SS:[EBP+8]
00401A76 A3 70334000 MOV DWORD PTR DS:[403370],EAX
00401A7B E8 E0FDFFFF CALL dropper.00401860
00401A80 33C0    XOR EAX,EAX
00401A82 5D      POP EBP
00401A83 C2 1000 RETN 10
00401A86 FF25 D0204000 JMP DWORD PTR DS:[<MSVCR90.memset>]
00401A8C 00      DB 00
00401A8D 00      DB 00
  
```

<그림 22> OllyDbg - main function

OllyDbg를 통하여 고급 동적 분석을 실시합니다.

IDA Pro를 이용하여 알아 두었던 메인함수의 주소값 401A70으로 이동하고 중요 함수였던 401860으로 접근하여 주요 API들을 확인하고 어셈블리어로 되어있는 프로그램을 한 단계씩 실행시키며 동적 분석을 실시합니다.

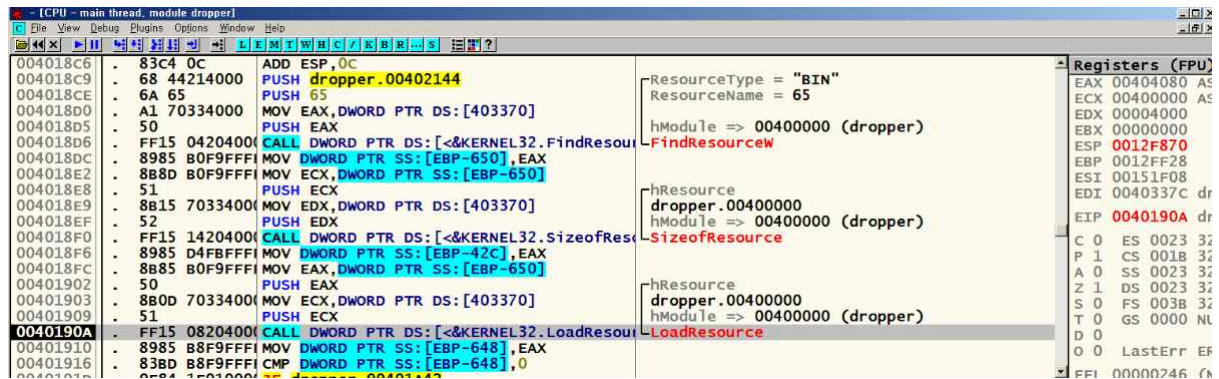
```

0040187D C785 B8F9FFFF MOV DWORD PTR SS:[EBP-648],0
00401887 C785 BCF9FFFF MOV DWORD PTR SS:[EBP-644],0
00401891 C785 B4F9FFFF MOV DWORD PTR SS:[EBP-640],0
0040189B 8D85 C0F9FFFF LEA EAX,DWORD PTR SS:[EBP-640]
004018A1 50      PUSH EAX
004018A2 68 05010000 PUSH 105
004018A7 FF15 1C204000 CALL DWORD PTR DS:[<&KERNEL32.GetTempPathW>]
004018AD 8D8D C0F9FFFF LEA ECX,DWORD PTR SS:[EBP-640]
004018B3 51      PUSH ECX
004018B4 68 24214000 PUSH dropper.00402124
004018B9 8D95 D8FBFFFF LEA EDX,DWORD PTR SS:[EBP-428]
004018BF 52      PUSH EDX
004018C0 FF15 DC204000 CALL DWORD PTR DS:[<&USER32.wsprintfw>]
004018C6 83C4 0C  ADD ESP,0C
  
```

<그림 23> OllyDbg - 401860 function 1

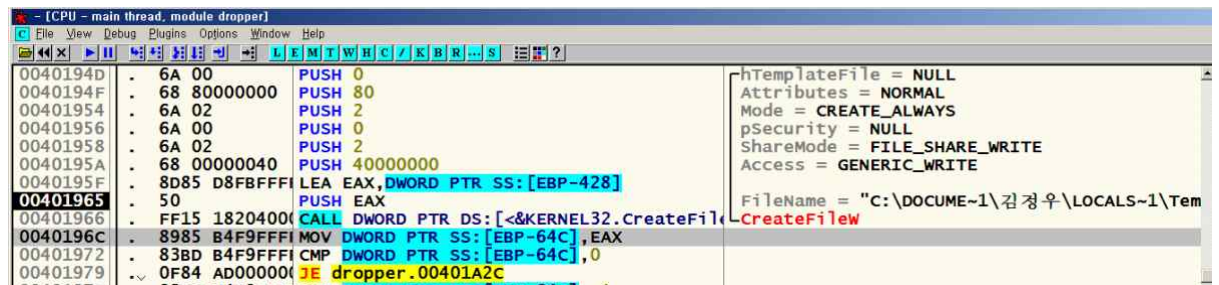
GetTempPathW 함수를 호출하여 Temp디렉토리의 경로를 찾고, ECX에 저장합니다.

그 후 sprintfw 함수에서 인자로 사용하여 "%s\notepad.exe" 문자열을 생성하고 추후에 드롭할 파일의 경로 및 이름으로 사용할 것으로 추측할 수 있습니다.

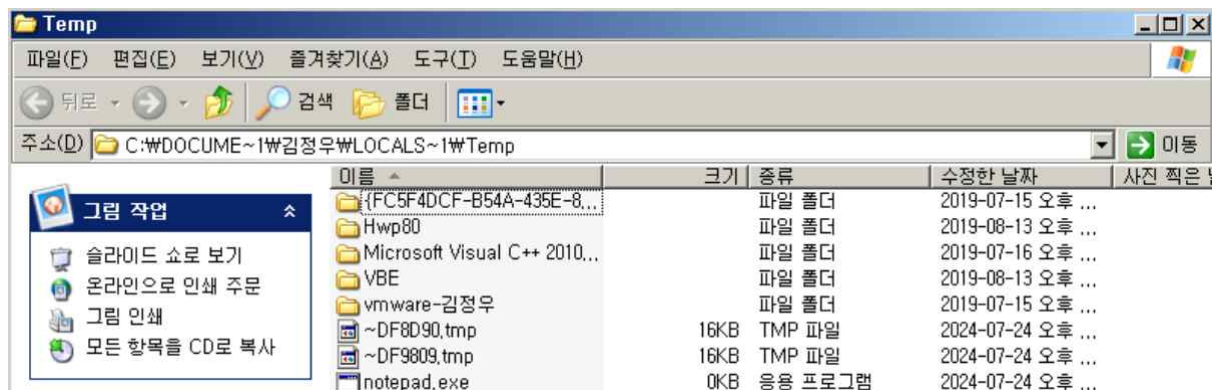


<그림 24> OllyDbg - 401860 function 2

Dropper 파일의 3가지 API (FindResourceW, SizeofResource, LoadResource) 를 호출하여 내부의 'BIN 101' PE 파일 찾기, 크기 계산, 메모리에 로딩 을 실시합니다.

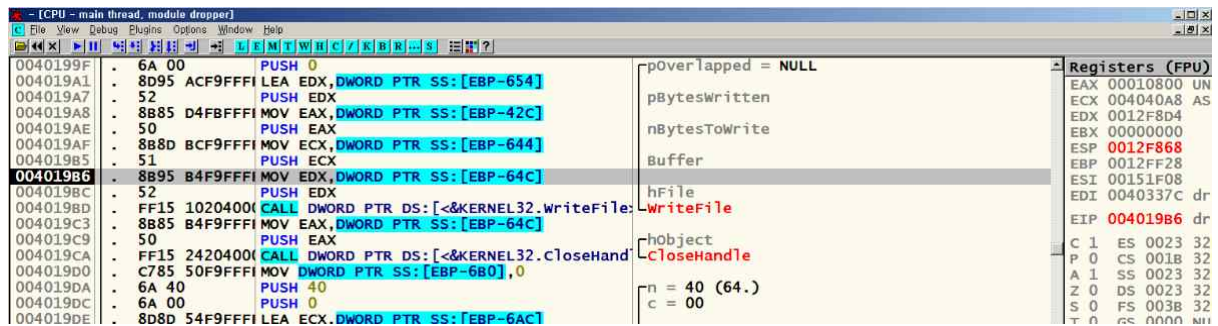


<그림 25> OllyDbg - 401860 function 3

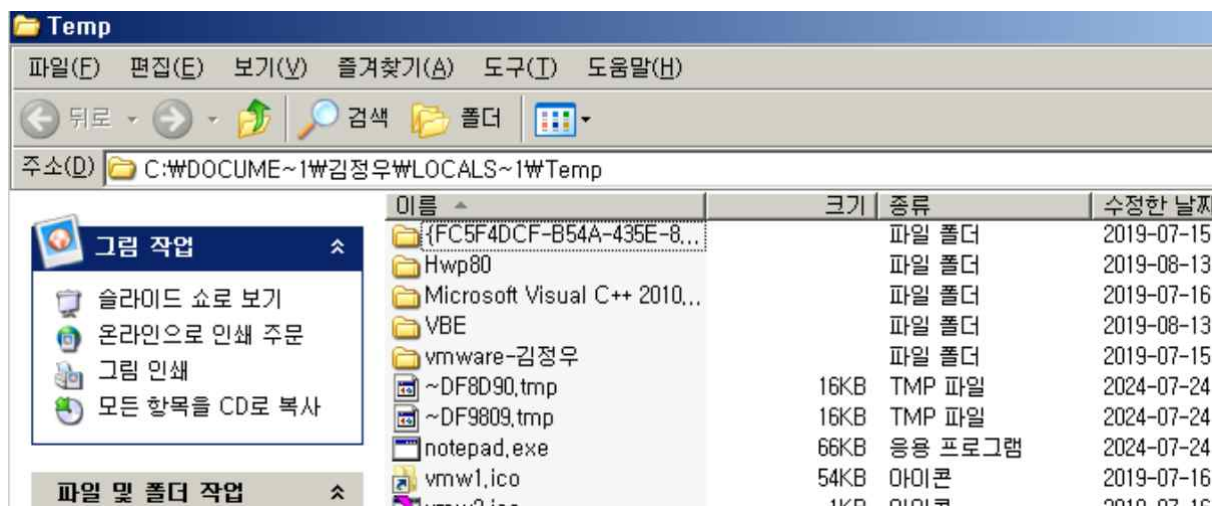


<그림 26> Temp 1

CreateFileW 를 호출하여 Temp 디렉토리에 notepad.exe 0바이트 파일을 생성 한 것을 확인합니다.



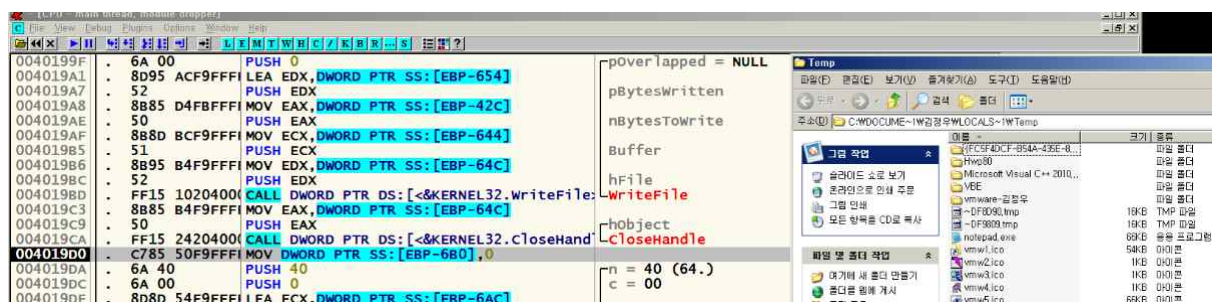
<그림 27> OllyDbg - 401860 function 4



<그림 28> Temp 2

WriteFile 함수 호출시 notepad.exe 0바이트 파일이 66KB로 쓰기 완료 됨을 확인 합니다.

아직 CloseHandle 함수를 호출하지 않았기 때문에 파일이 완성된 것은 아닙니다. 따라서 실행이 되지 않으며, 아이콘도 활성화되지 않음을 알 수 있습니다.



<그림 29> OllyDbg - 401860 function 5

CloseHandle 함수를 호출하면 notepad.exe 파일이 완성되며 아이콘이 활성화 되어

파일을 실행 할 준비를 마쳤음을 알 수 있습니다.

Address	Disassembly	Comment
004019F7	MOV DWORD PTR SS:[EBP-684],0	
00401A01	LEA EDX,DWORD PTR SS:[EBP-664]	
00401A07	PUSH EDX	
00401A08	LEA EAX,DWORD PTR SS:[EBP-680]	
00401A0E	PUSH EAX	
00401A0F	PUSH 0	
00401A11	PUSH 0	
00401A13	PUSH 0	
00401A15	PUSH 0	
00401A17	PUSH 0	
00401A19	PUSH 0	
00401A1B	LEA ECX,DWORD PTR SS:[EBP-428]	
00401A21	PUSH ECX	
00401A22	PUSH 0	
00401A24	CALL DWORD PTR DS:[<KERNEL32.CreateProcessW>]	
00401A2A	JMP SHORT dropper.00401A40	

<그림 30> OllyDbg - 401860 function 6

CreateProcessW 함수를 호출하면 완성된 notepad.exe 파일이 실행됩니다.

Address	Disassembly	Comment
00401A6D	INT3	
00401A6E	INT3	
00401A6F	INT3	
00401A70	PUSH EBP	
00401A71	MOV EBP,ESP	
00401A73	MOV EAX,DWORD PTR SS:[EBP+8]	
00401A76	MOV DWORD PTR DS:[403370],EAX	
00401A7B	CALL dropper.00401860	
00401A80	XOR EAX,EAX	
00401A82	POP EBP	
00401A83	RETN 10	
00401A86	JMP DWORD PTR DS:[<MSVCR90.memset>]	MSVCR90.memset

<그림 31> OllyDbg - main function 2

그 후 메인함수로 복귀하며 메인함수도 RETN 하여 종료됩니다.

7. 분석 결론 및 대응 방안

7.1 악성 파일의 분석 결론

기초 분석

VirusTotal 을 통해 기초 분석을 수행하여 여러 백신에서 dropper.exe 파일이 Trojan 성질 및 Dropper 기능을 하는 악성 코드임을 확인 했습니다. 해당 파일은 Windows 32bit 운영 체제에서 작동되는 실행 파일이며, 패킹되지 않은 악성파일이기때문에 언패킹 할 필요 없이 분석 하였습니다. 추가 정보로 파일 생성 날짜, 최초 발견일, notepad.exe 파일을 drop하는 악성 코드 인 점을 파악하였습니다.

이후 정적 분석을 통해 교차 검증이 필요했기에 다양한 프로그램을 통해 분석을 실시하였습니다.

정적 분석

HashCalc 프로그램을 통해 파일의 해시값을 파악하고, Exeinfo PE 프로그램으로 해당 파일이 C++ 언어로 만들어지고, 패킹되지 않은 파일인것을 알 수 있었습니다. PE view 를 사용하여 해당 파일이 .exe 실행 파일인것, 참조하고 있는 IAT 정보와 추가적으로 BIN타입의 .exe 실행파일을 내장하고 있다는 것을 알 수 있었습니다. 이후 Dependency Walker 와 Strings 을 사용하여 파일의 API 들을 파악하여 dropper파일인것을 교차 검증 하고 마지막으로IDA Pro 를 사용하여 상세한 정적 분석을 통해 파일 내부의 함수들을 하나씩 살펴보면 해당 파일이 notepad.exe 파일을 temp 디렉토리에 저장하고, 실행시키는 dropper 형식 악성 코드인것으로 결과를 얻어냈습니다.

동적 분석

기초, 정적 분석을 통해서 dropper.exe 파일이 내부에 숨겨놓은 notepad.exe 파일을 temp 디렉토리에 저장 및 실행 시키는 dropper 형식 악성 코드인 점을 확인 하였고, OllyDbg 분석도구를 이용해 파일 동작 순서를 상세하게 동적 분석 하여 기초, 정적 분석한 내용을 검증하였습니다.

7.2 대응 방안

종합적인 분석을 통하여 dropper.exe 파일은 트로이목마 성질을 지니고 있는 dropper 형식의 악성코드라는 점을 확인 했습니다. 파일 내부에 숨겨져 있던 파일이 notepad.exe 가 아닌 위험한 공격 프로그램이었다면 PC사용자가 알 수 없게 내부에 설치 및 실행되어 개인정보 유출, 해킹 등의 피해가 발생할 수 있습니다.

dropper 형식 악성코드에 대응하기 위해서는 다음과 같은 대응 방안을 숙지 할 필요성이 있습니다.

1. 소프트웨어를 주기적으로 업데이트하여 해커가 이전 버전의 소프트웨어 취약점을 활용할 수 없도록 합니다.

2. dropper 형식의 악성코드는 사용자도 인지하지 못한 채 당할 수 있는 Malware유형의 공격이기 때문에 특정 보안 솔루션으로 방어하는 계획 보다는 상식적인 방법으로 이를 예방하는 것이 더 효과적인 대응 방안입니다. 개인, 사내 등 어디에서나 메일, 사이트 등을 사용함에 있어서 의심이 가거나 안전하지 않은 잘 알지 못하는 파일과 프로그램을 다운 및 실행 시키는 행위는 의식적으로 조심하며 링크나 첨부파일을 함부로 클릭 및 실행 시키는 행동을 해서는 안됩니다.

3. AhnLab-V3 등 위협 방지 백신 소프트웨어들을 설치하여 악성코드들의 접근을 방지합니다.