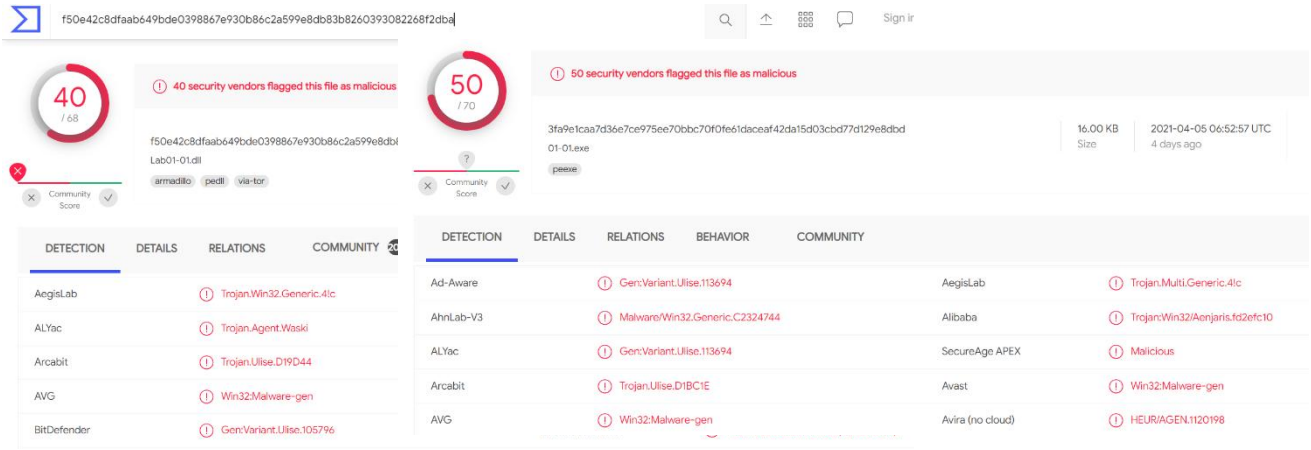
[Lab01-01~Lab01-04에 대한 기초 정적분석]

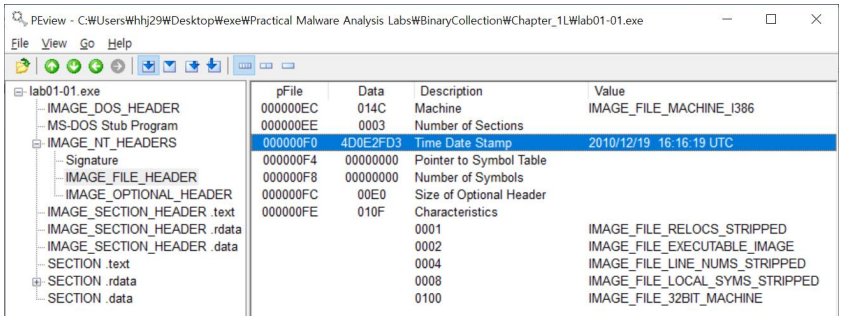
[lab01-01.exe/lab01-01.dll]

1. Virustotal에 업로드 하고 보고서를 확인하자 기존 안티바이러스 정의된 것과 일치하는가?

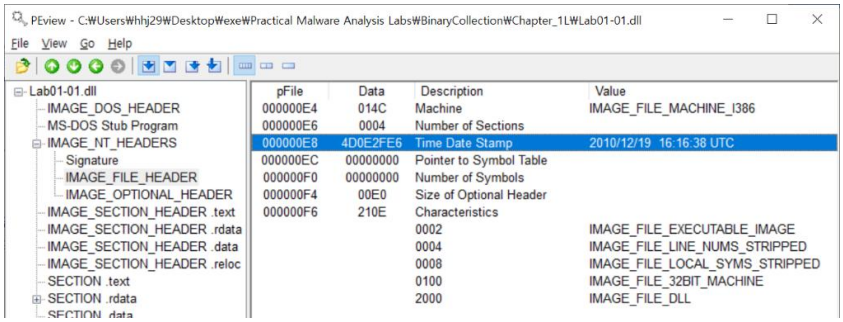


두 파일 모두 40~50개의 엔진에서 악성코드로 판별했고 윈도우 32비트에서 작동하며 Trojan이라는 명칭이 붙어 있다.

2. 이 파일은 컴파일 시점은

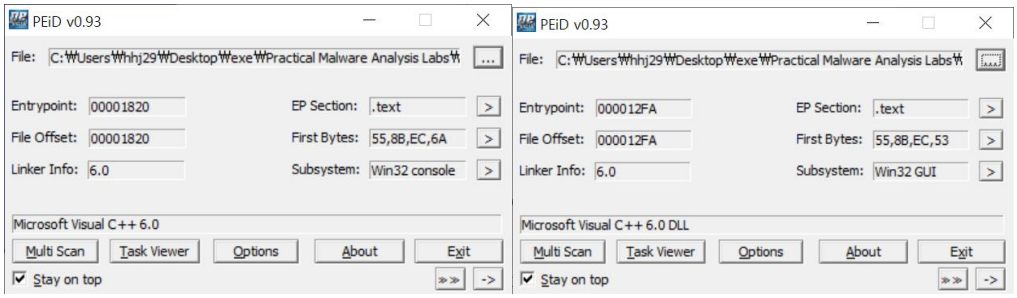


exe 파일은 2010년 12월 19일 16시16분 (UTC)에 컴파일 된 파일이다.



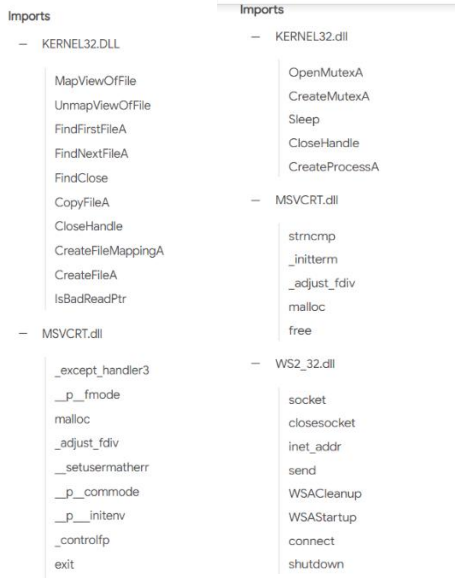
dll 파일도 2010년 12월 19일 16시16분 (UTC)에 컴파일 된 파일인 것으로 보아 두 파일은 같이 생성됐으며 exe 파일이 실행되고 잇따라 dll 파일이 실행된다는 것을 유추할 수 있다.

3. 패킹이나 난독화의 흔적이 있는가? 이유는



두 파일 모두 C++로 컴파일된 파일이며 EP Section이 text인 것으로 보아 패킹 되지 않은 파일이다.

4. 감염된 시스템에서 검색할 수 있는 다른 파일이나 호스트 기반의 증거가 존재하는가?



두 파일에서 본 imports API 함수를 살펴보았다.

*KERNEL32.dll (exe)

FindFirstFileA와 FindNextFileA는 파일 이름을 통해 파일을 검색하는 함수로 특정 파일을 찾는 것 같다.

CopyFileA는 기존 파일을 새 파일로 복사하는 함수로 찾은 파일을 복사하는 역할을 할 것 같다.

CreateFileA는 파일이나 I/O 장치를 만들거나 여는 함수

*KERNEL32.dll (dll)

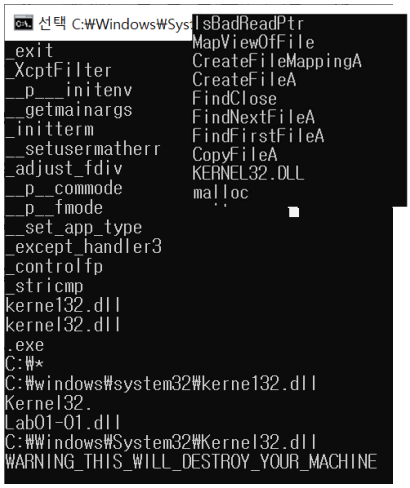
Sleep은 지정 시간만큼 대기하는 것으로 보아 백도어 기능을 수행하기 위한 함수인 것 같다. CreateProcessA 함수는 새 프로세스와 기본 스레드를 만드는 함수로 프로세스를 실행시키는 역할을 한다. 어떠한 프로세스를 실행시키는 것 같다.

MSVCRT.dll (exe, dll)

malloc은 메모리를 동적으로 할당시켜서 어떠한 프로세스를 실행하려는 것 같다.

WS2_32.dll

WS2_32는 Windows에서 제공하는 라이브러리로 Windows Socket을 사용하기 위한 함수라고 한다. socket, send, connect 등의 소켓 통신을 제공하는 함수들이 존재하는 것을 보아 통신으로 파일을 전송하는 등의 행동을 보일 것 같다.



차례대로 exe와 dll 파일을 strings 툴을 이용해 문자열들을 나열한 장면이다. exe 파일의 문자열들을 살펴보면 kerne132.dll과 kernel32.dll 파일이 존재하는 것을 볼 수 있는데 자세히 보면 1과 l의 차이가 존재한다. 즉 유사한 형태를 이용해 조작한 함수를 실행하도록 한 형태같다. 더불어 Lab01-01.dll 파일명도 존재한다. 위에서 봤던 파일을 검색하고 복사하는 함수에서 인자 값으로 들어가던 파일명으로 사용되거나 해당 파일을 실행하도록(CreateFileA) 사용될 것으로 추측된다.

dll 파일을 보면 Sleep과 CreateProcessA 함수를 볼 수 있는데 백도어로 대기하고 있다가 실행되는 것으로 예측된다.

5. 감염된 장비에서 이 악성코드를 발견하기 위해 사용한 네트워크 기반의 증거는 무엇인가?

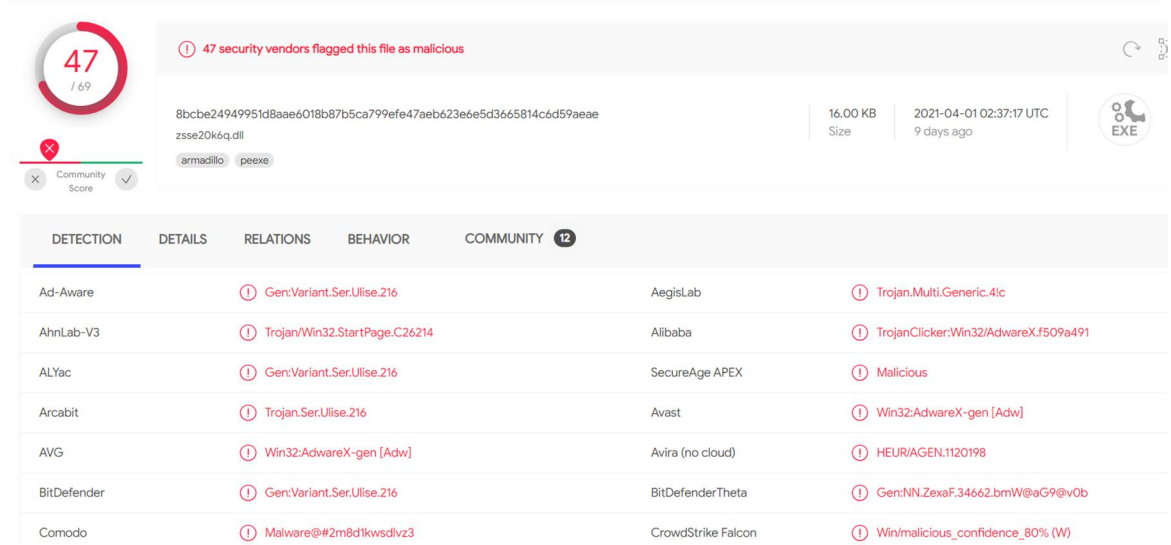
앞서 말했듯이 dll 파일에 import된 API 라이브러리인 WS2_32 내의 함수들이 네트워크를 기반으로 해서 통신을 하려는 것으로 보이고 127.26.152.13 아이피 값이 존재하는 것으로 보아 해당 주소로 통신을 시도해 파일을 전송하는 것이 아닐까 생각한다.

6. 이 파일의 목적은 무엇이라고 판단하였는가?

어떠한 파일을 검색하고 정해진 아이피로 소켓을 연결해 파일 통신하려는 용도로 제작된 악성코드라고 생각한다.

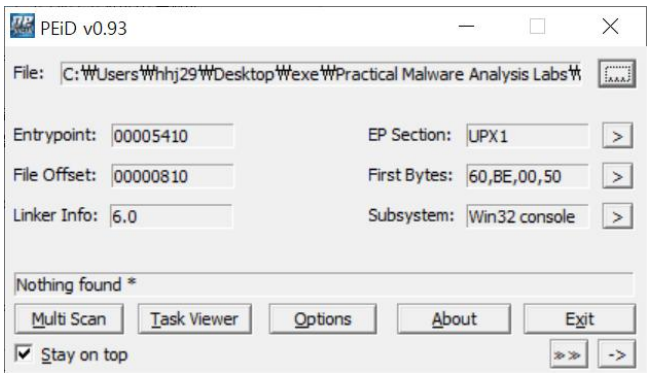
[lab01-02.exe]

1.Virustotal에 업로드 하고 보고서를 확인하자 기존 안티바이러스 정의된 것과 일치하는가?



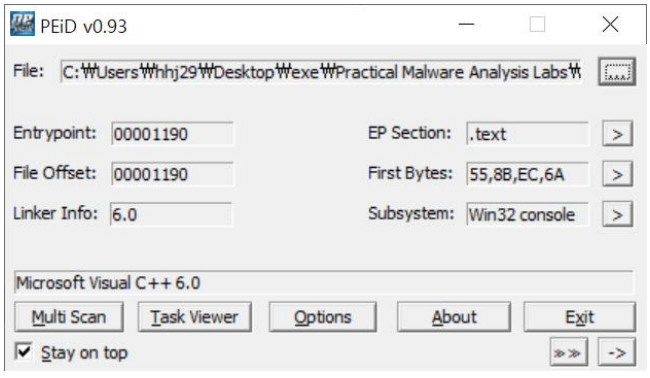
47개의 엔진에서 악성코드로 판별했고 윈도우 32비트에서 작동하며 Trojan이라는 명칭이 붙어 있다. 악성코드의 시그니처로 볼 수 있다.

2.패킹이나 난독화의 흔적이 있는가? 이유는



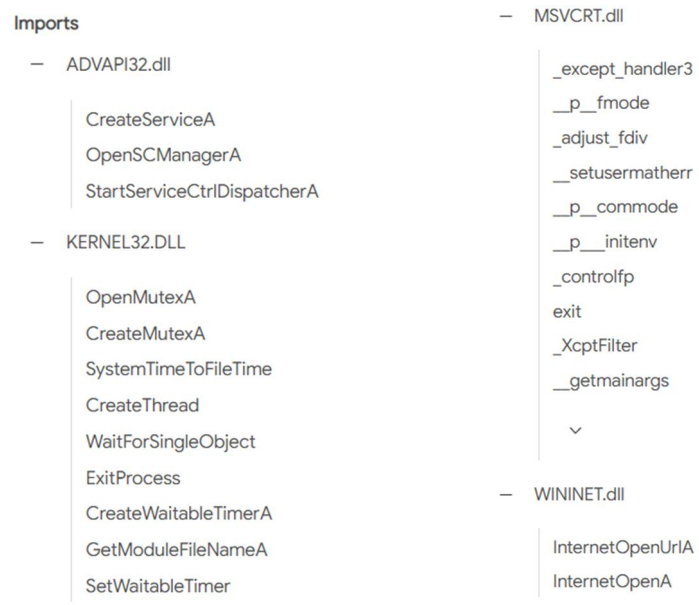
컴파일 된 언어가 나오지 않고 EP Section이 UPX1로 설정된 것으로 보아 UPX로 패킹된 파일이라는 것을 알 수 있다.

upx를 사용해 언패킹 해보았다.



C++로 컴파일 된 파일이며 EP Section이 text인 것을 확인할 수 있다.

3.임포트를 보고 악성코드의 기능을 알아 낼 수 있는가? 그렇다면 어떤 임포트를 보고 알 수 있었는가?



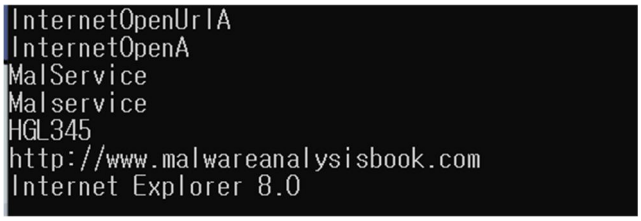
*ADVAPI32.dll - Windows 레지스트리, 시스템 다시 시작 및 종료, Windows 서비스 시작/중지 및 생성, 사용자 계정 관리와 같은 작업을 담당하는 API라고 한다.

OpenSCManagerA에서 SCM을 여는데 SCM은 Service Control Manager로 서비스 DB를 관리하는 시스템인데 여기서 서비스는 시스템 부팅 시 시작과 동시에 실행해야 되는 것을 의미하며 이러한 서비스 목록을 레지스트리에 저장한 것을 서비스 데이터 베이스라고 한다. 즉 여기서 서비스 DB를 관리하는 프로그램을 열고 CreateServiceA를 통해 어떠한 특정 서비스를 추가하는 것으로 예측된다.

*WININET.dll – 네트워크 기능, ftp, http, ntp 같은 프로토콜을 구현한 상위 수준의 네트워크 함수를 가지는 dll로 쉽게 네트워크에 접근 또는 연결할 수 있는 함수를 가졌다.

InternetOpenA은 Wininet 함수를 초기화하면서 인터넷 사용을 준비하는 역할을 하며,

InternetOpenUrlA에서 ftp또는 http url로 지정된 리소스를 여는 것을 보아 특정 url에 접근하려는 것으로 보인다.



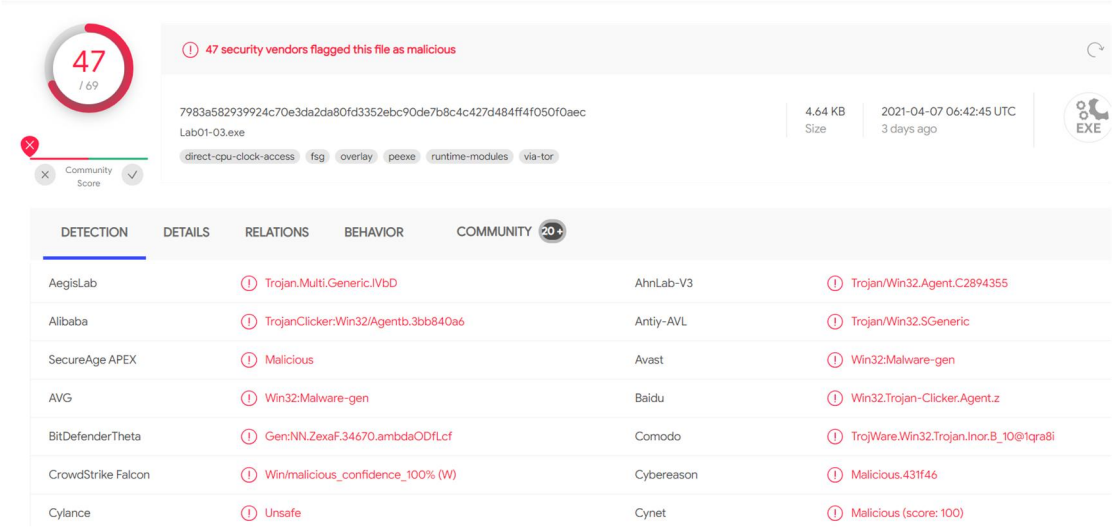
Strings를 통해 문자열을 확인해보니 MalService라는 특정 서비스 이름이 보인다. 해당 서비스를 CreateServiceA를 통해 추가하는 것 같고, url 주소도 발견할 수 있는데 해당 주소로 접근하려는 것임을 추측할 수 있다.

4.감염된 장비에서 이 악성코드를 발견하기 위해 사용한 네트워크 기반의 증거는 무엇인가?

WININET.dll의 함수 InternetOpenA와 InternetOpenUrlA를 보아 네트워크 작업을 하며 인터넷을 사용해 특정 사이트로 접근을 하려는 것 같다.

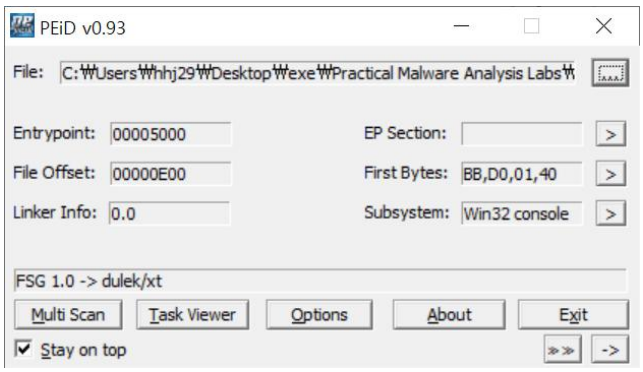
[lab01-03.exe]

1.Virustotal에 업로드 하고 보고서를 확인하자 기존 안티바이러스 정의된 것과 일치하는가?

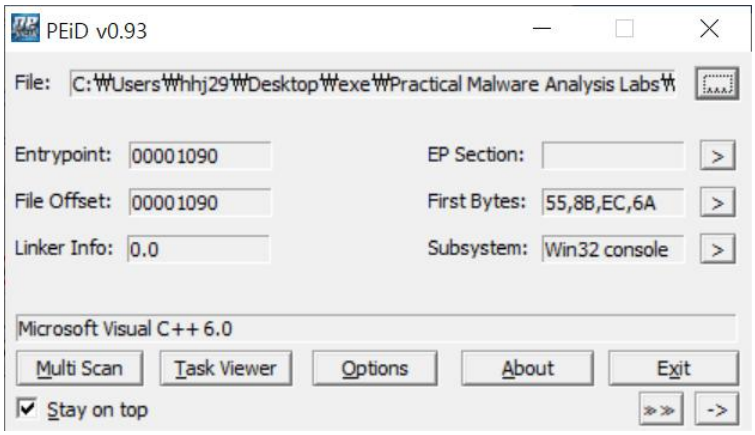


47개의 엔진에서 악성코드로 판별했고 윈도우 32비트에서 작동하며 Trojan이라는 명칭이 붙어 있다. 악성코드의 시그니처로 볼 수 있다.

2.패킹이나 난독화의 흔적이 있는가? 이유는



FSG 1.0 -> dulek/xt로 처음 보는 단어가 적혀 있고 EP Section이 표시되지 않은 것으로 보아 패킹 되어 있는 것 같다. FSG 언패킹 방법을 검색해보니 ollydbg로 oep를 찾아 덤프 하면 언패킹된다고 해서 따라 해보았다



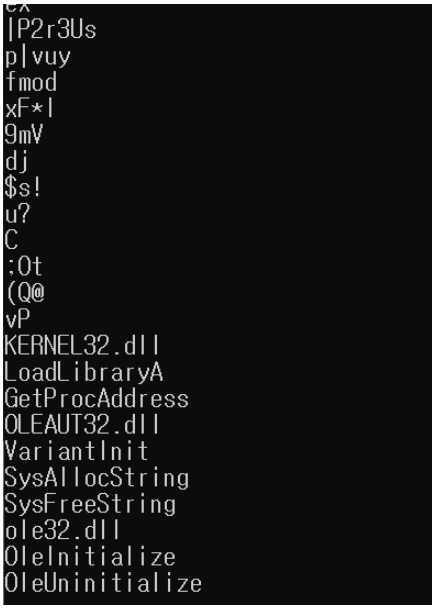
다음과 같이 덤프한 파일을 PEiD에 넣어 패킹 여부를 확인해보았다. C++로 컴파일 되었다는 것을 알 수 있었다.

3.임포트를 보고 악성코드의 기능을 알아 낼 수 있는가? 그렇다면 어떤 임포트를 보고 알 수 있었는가?

언패킹 된 파일을 virusTotal에 넣고 import api 함수를 확인했다.

*OLEAUT32.dll

- 우선 변수를 초기화 해서 SysAllocString으로 새 문자열을 할당하고 전달된 문자열을 여기에 복사한다. SysAllocString으로 할당된 문자열을 할당 해제한다. 어떤 특정 문자열을 복사하는 기능인 거 같다.....



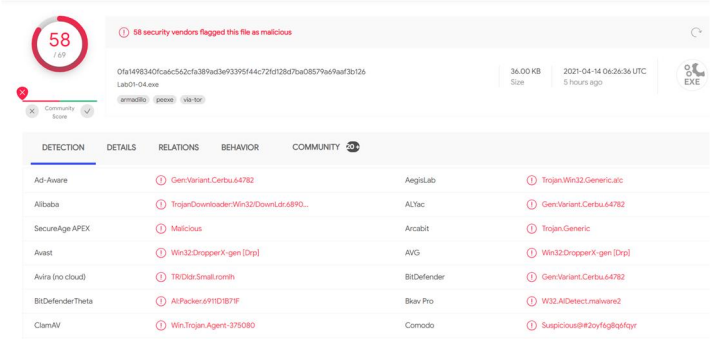
특별히 별다른 함수를 찾긴 어려웠고 대신 특정 url 주소를 발견했다.

4.감염된 장비에서 이 악성코드를 발견하기 위해 사용한 네트워크 기반의 증거는 무엇인가?

특정 인터넷 주소가 나와있는 것으로 보아 해당 url 주소로 연결되도록 짜여 진 코드일 것 같다는 추측이 든다.

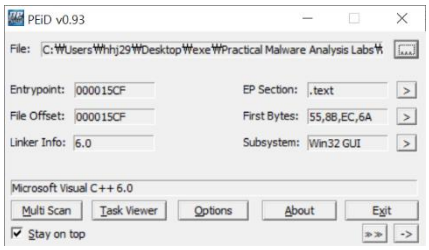
[lab01-04.exe]

1. Virustotal에 업로드 하고 보고서를 확인하자 기존 안티바이러스 정의된 것과 일치하는가?



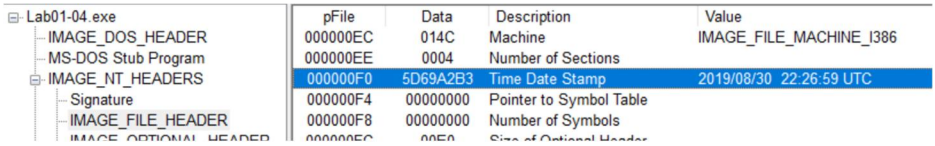
두 파일 모두 58개의 엔진에서 악성코드로 판별했고 윈도우 32비트에서 작동하며 Trojan이라는 명칭이 붙어 있다.

2.패킹이나 난독화의 흔적이 있는가? 이유는? 파일이 패킹되어 있다면 언패킹하라.



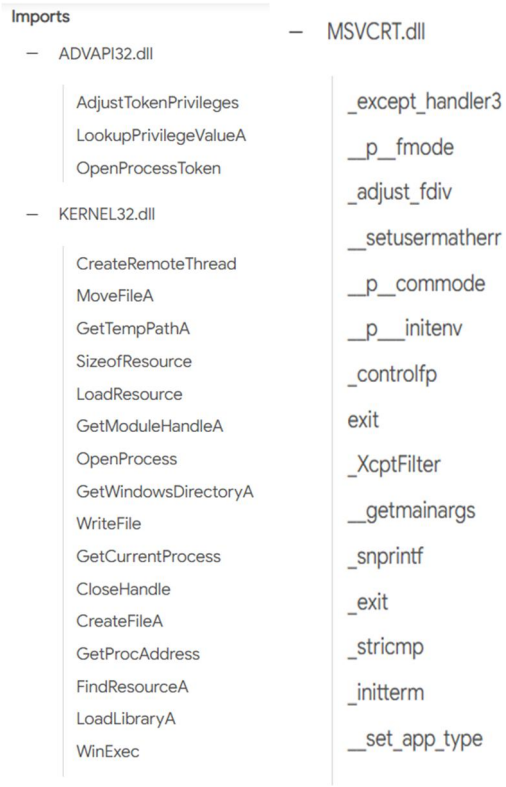
해당 파일은 C++로 컴파일 된 파일이며 EP Section이 text인 것으로 보아 패킹 되지 않은 파일이다.

3.이 프로그램은 언제 컴파일 됐는가?



PEview로 IMAGE_FILE_HEADER의 Time Date Stamp를 살펴보니 2019년 8월 30일에 22시 26분에 컴파일 된 파일이다.

4.임포트를 보고 악성코드의 기능을 알아 낼 수 있는가? 그렇다면 어떤 임포트를 보고 알 수 있었는가?



-ADVAPI32.dll

AdjustTokenPrivileges, LookupPrivilegeValueA, OpenProcessToken 이 세 가지 함수로 보호된 프로세스의 권한을 재설정하여 권한 상승을 얻는 것에 쓰인다. 아마도 접근 권한이 필요한 프로세스에 접근하는 용도로 사용되는 함수가 아닐까 하는 생각이 들었다.

-KERNEL32.dll

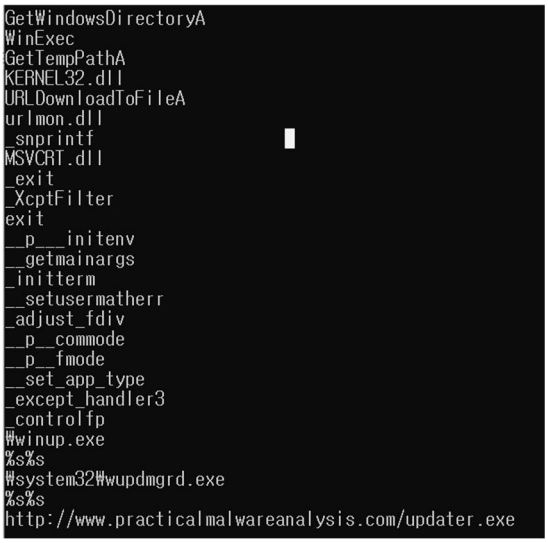
WinExec: 지정된 애플리케이션을 실행한다.

GetWindowsDirectoryA: Windows 디렉터리의 경로를 검색한다.

FindResourceA, LoadResource,OpenProcess 등 리소스를 찾고 로드하고 프로세스를 실행하는 함수들이 존재한다.

5.감염된 장비에서 이 악성코드를 발견하기 위해 사용한 네트워크 기반의 증거는 무엇인가?

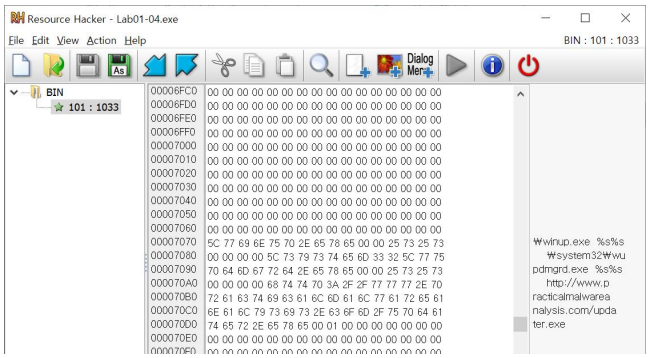
네트워크와 관련된 함수 정보를 찾기 어려워 Strings를 이용해 문자열을 탐색해보았다.



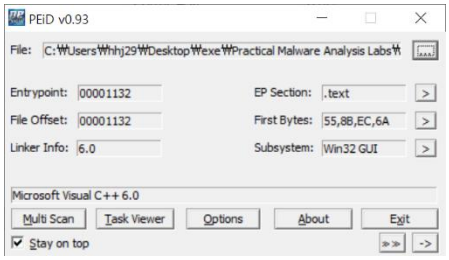
Strings를 살펴보면 virusTotal에서 찾은 Api 함수 목록들을 볼 수 있었는데 조금 더 내려보니 비슷한 유형의 함수 목록들이 더 존재하는 것을 확인할 수 있었다. 더불어 url 주소와 함께 URLDownloadToFileA라는 함수, 그리고 urlmon.dll을 발견할 수 있었다. 인터넷에 검색해보니 URLDownloadToFileA는 urlmon.dll에 있는 함수로 해당 url을 이용해 인터넷에서부터 파일을 다운

받는 기능이 있다. 따라서 인터넷에 접속을 통해 해당 url주소로부터 어떠한 파일을 다운받는 것이 목적인 악성코드인 것 같다.

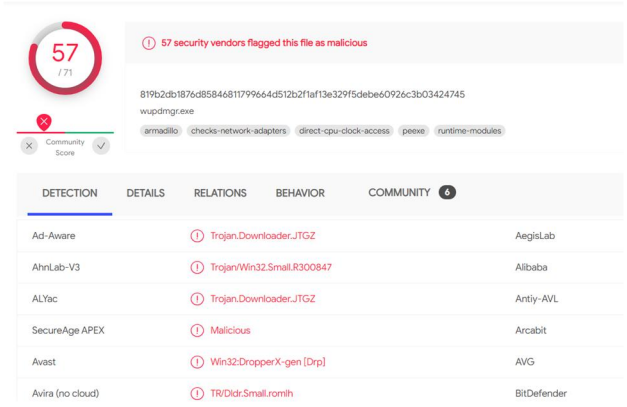
6.이 파일은 리소스 섹션에 하나의 리소스를 Resource Hacker를 이용해 리소스를 점검하고 추출해 보자. 리소스로부터 무엇을 알 수 있는가?



lab01-04 파일을 resource Hacker를 통해 열고 내부에 숨겨진 악성코드 파일을 추출해 보았다.



PEiD를 통해 패킹 여부를 확인해보았고 패킹 되어있지 않은 파일이었다.



Imports

- KERNEL32.dll
  - GetWindowsDirectoryA
  - GetTempPathA
  - WinExec
- MSVCRT.dll
  - _except_handler3
  - __p__fmode
  - _initterm
  - _adjust_fdiv
  - __setusermatherr
  - __p__commode
  - __p___initenv
  - exit
  - _XcptFilter
  - __getmainargs
- urlmon.dll
  - URLDownloadToFileA

HTTP Requests

+ http://www.practicalmalwareanalysis.com/updater.exe
+ http://www.practicalmalwareanalysis.com

virulTotal에서 파일을 확인해보니 악성코드가 맞으며 urlmon.dll의 함수 URLDownloadToFileA를 사용해 해당 주소로의 접근이 있다는 사실을 확인할 수 있다.