기술문서	`08.10.25. 작성

악성코드 분석

작성자 : 영남대학교 정보보호 연구학회 @Xpert
김슬예나 prehea@ynu.ac.kr

목차

1. 소개

인터넷 통계 정보 시스템(ISIS)에 따르면 전세계 인터넷 이용자수는 2007년 기준으로 1.467,040천명, 약 15억 명 가까이 된다고 한다. 그에 따라 웹 서비스도 증가하고 있으며 더불어 악성코드의 종류와 그 수도 꾸준히 증가하고, 어딘가에서 끊임없이 생성되고 있다. 한 달에도 평균 400건의 악성코드 감염 사례가 발견되고 있으니 악성코드의 위험은 그냥 넘어갈 문제가 아니다. 따라서 필자는 이 문서를 통해 악성코드 분석기법을 알아보고자 한다.

가. 악성코드란?

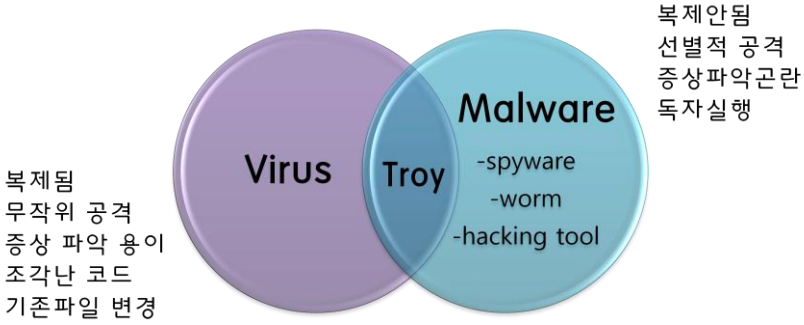
악성코드는 영어사전에 멀웨어(Malware)라고 표기되어있다. 이 단어는 "Malicious Software"의 약자로 악의적인 소프트웨어라는 의미를 가지고 있다. 이처럼 악성코드는 말 그대로 컴퓨터에 악영향을 끼치는 소프트웨어를 총칭하는 단어이다.

악성코드의 역사를 말하자면 끝이 없을 정도로 길다. 컴퓨터 바이러스란 용어가 정립된 80년대 중반 이전에도 이미 바이러스, 웜, 트로이목마에 해당하는 악성코드가 존재했다고 할 정도로 악성코드 역사는 오래되었다. 그 이후로 미켈란젤로 바이러스, 코드 레드 웜 등의 크고 작은 감염 사례가 발생하였고 플래쉬 파일 감염, 이메일을 통한 감염, IRC 봇을 이용한 감염 등 악성코드 감염 방법도 다양하게 발전하였다. 이처럼 악성코드는 사용자의 의사와 이익에 반해 정보 등을 불법으로 유출시키는 작용을 하는 것을 이른다.

나. 악성코드와 바이러스

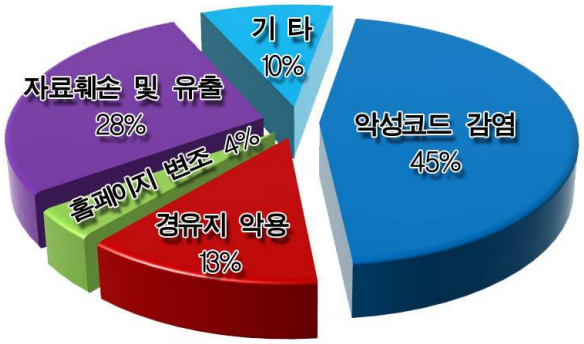
악성코드는 감염 방법과 증상 등에 따라 바이러스, 웜, 트로이목마 등으로 구분되므로 악성코드를 바이러스보다 큰 개념으로 이해하면 될 것 같다.

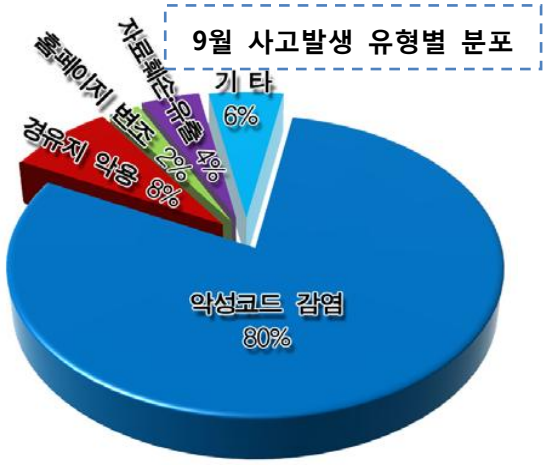
스파이 웨어(Spyware)는 특정 소프트웨어에 포함되어 설치된 후 사용자 정보를 외부로 유출하는 악성코드를 말하며 웜(Worm)은 LAN또는 인터넷을 통해 전파되는 프로그램을 말한다. 이 외에도 해킹 툴이나 애드웨어 등도 흔히 악성코드라고 부른다. 트로이 목마(Trojan)는 사용자의 인지나 허가없이 실행되는 다양한 프로그램 변종을 일컫는 말이다. 데이터를 수집하여 사이버 범죄자에게 전송하거나, 악의적 목적으로 데이터를 파괴 및 변조할 때, 악의적 혹은 범죄의 목적으로 PC를 사용할 때 많이 이용되며, 스팸 발송용으로 자주 이용된다.



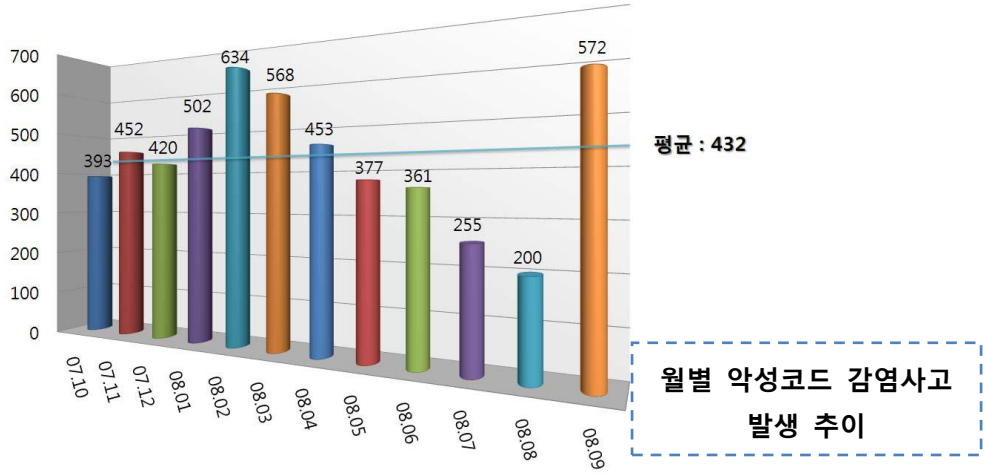
다. 악성코드 감염사고 발생현황

국가 사이버 안전 센터에서 발간하는 'Monthly 사이버 시큐리티'를 참고하면 지난 7월 사고발생





총 486건 중 255건(52%)이, 8월 사고발생 총 439건 중 200건(45%)이 악성코드 감염 사례인 것으로 나타났으며 최근에 발행된 10월호에서는 지난 9월 사고발생 총 716건 중 572건(80%)이 악성코드 감염 사례로 보고되었다. 이는 전월 대비 186%(572건)나 증가한 수치이다.



2. 준비

악성코드를 분석하기 위해 필요한 준비를 살펴보자.

가. 필요한 사전 지식

1) 어셈블리어

어셈블리어는 기계어와 1:1대응으로 이루어진 저급언어(Low-level Language)이다. 보통의 악성코드 실행파일들은 프로그램 소스가 공개되어 있지 않으므로 악성코드 파일을 Ollydbg, IDA등의 프로그램을 사용하여 Disassemble한 코드(어셈블리어코드)를 바탕으로 악성코드를 분석하여야 한다.

2) PE파일 구조

PE파일 포맷(Portable Executable File Format)이란 Win32운영체제 시스템에서 실행 가능한 파일 형식이다. 파일에 담겨 다른 곳에 옮겨져도 실행시킬 수 있도록 규정한 형식이라는 뜻이다. 우리가 평소에 자주 접하는 EXE, DLL파일 모두가 PE파일 구조를 가지고 있다. 악성코드 파일도 EXE파일이 많으므로 PE파일 구조를 가지고 있다고 할 수 있다.

3) 그 외..

이 외에도 악성코드가 작동하는 운영체제에 대한 지식, 악성코드가 번식하거나 작동하는 데 사용되는 네트워크에 관한 지식, API(Application programming interface)에 대한 지식 등..들도 악성코드 분석에 도움을 줄 것이다.

나. 분석 환경

악성코드를 분석해 보려면 우선 실행을 시켜보아야 한다. 악성코드를 실행시킬 환경은 당연히 외부와 연결이 되지 않은 통제된 환경이야 할 것이다. 그래서 필자는 VMware라는 가상 머신을 2개 구축하여 사용하였다.

위 그림처럼 가상 머신 2개를 GUEST와 HOST로 나누었다. HOST는 GUEST의 DNS로 설정하고 두 머신의 네트워크 연결방식은 Host-Only로 설정하였다. Host-Only는 GUEST와 HOST간의 통신만 가능하게 해주는 것인데, Host PC만 네트워크 사용을 가능하게 해주는 것이다. GUEST에서 작동하고 있는 악성코드가 외부로 패킷을 보내려고 할 때에 HOST에서 그 패킷을 캡쳐하기 위하여 Host-Only모드를 사용, HOST머신을 DNS로 설정하였다.

다. 툴 소개

악성코드를 분석할 때 필자가 사용한 도구들은 다음과 같다.

1) 모니터링 툴

- Autoruns : 윈도우 시작프로그램 감시.
- Filemon : 실시간으로 현재 프로그램이 읽고 쓰는 파일을 탐지.
- Proccess Explorer : 실시간으로 현재 프로세스의 목록 감시.
- Regmon : 실시간으로 현재 프로그램이 읽고 쓰는 레지스트리를 탐지.
- TCPview : 현재 주고받는 TCP 패킷 탐지, 네트워크 연결정보 확인.
- WinAlysis : 현재 시스템 상황을 스냅샷을 찍어 다음 상황과 비교.
- WireShark : 많은 사람들이 사용하는 패킷 분석기. 네트워크 프로토콜 해석기.

2) 코드분석 툴

- OllyDbg : 프로그램을 디버깅하고 분석할 때 유용한 도구.

- IDA : 다양한 기능을 가지고 있는 디스어셈블러
- BinText : 파일의 String추출.

3) 분석 툴

- PEiD : PE파일의 packing여부 탐지.

여러 가지 도구들을 소개했지만 꼭 이 도구만 사용하라는 법은 없다. 이 외에도 다른 도구들이 많이 있으니 사용하면 도움이 될 것이다.

3. 분석 절차 소개

본격적으로 악성코드 분석에 들어가기 전에 간단하게 분석 절차를 알아보자. 분석단계는 필자가 여러 문서들을 참고, 임의로 적은 것이니 분석단계의 정석이라고 할 수는 없다.

가. 초기 분석

초기분석은 말 그대로, 본격적으로 악성코드를 분석하기 전에 악성코드의 외형을 분석하는 단계이다. 악성코드의 외형이라 함은 악성코드를 실행하지 않고도 알 수 있는 것을 말한다. 예를 들면, 악성코드 파일의 용량과 실행압축이라고 부르는 패킹여부 등을 말한다. 이 밖에도 악성코드의 외형뿐만 아니라 인터넷을 참조, 여러 가지 정보를 수집하여 분석할 악성코드의 시나리오를 예상해 본다.

나. 동적 분석

악성코드의 코드를 분석하기 전에 악성코드를 직접 실행시켜보는 단계이다. 가상 환경을 구축한 후 그 안에서 악성코드를 직접 실행시켜보고 여러 가지 모니터링 툴들을 이용하여 악성코드의 동작 방식을 분석한다. 이 과정에서 필자는 VMware의 기능 중 하나인 'SnapShot'기능을 매우 유용하게 사용하였다.

다. 정적 분석

악성코드를 디버깅하고 코드를 분석하는 단계이다. 앞서 설명한 OllyDbg나 IDA등의 도구들을 사용하여 악성코드의 실행파일을 디스어셈블하여 정밀 조사가 필요한 악성코드의 핵심 부분을 세밀하게 분석한다.

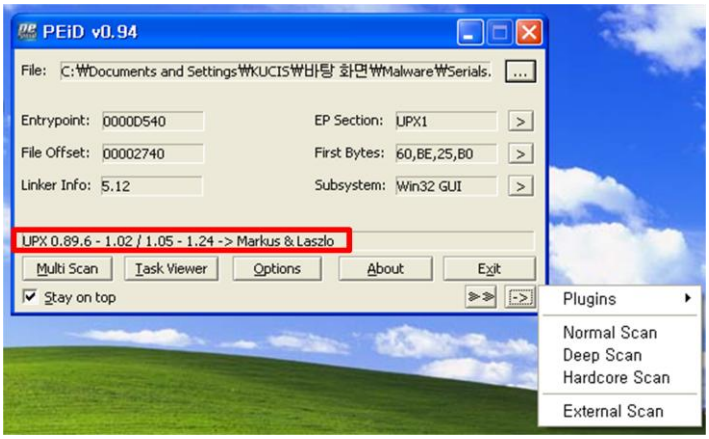
4. 악성코드 분석

본격적인 악성코드 분석에 들어가보자. 필자는 E-mail 악성코드를 분석해보았다.

가. 초기 분석

우선 분석할 악성코드를 준비해보았다. 악성코드의 이름은 'Serials.txt'인 exe파일이다. 파일의 크기는 12KB이다.

아래 그림처럼 PEiD를 사용하여 Packing여부를 확인해보았다. UPX로 패킹된 것이 확인되었다. 더 확실히 패킹여부를 확인하고 싶다면 'Hardcore Scan'을 하면 된다.



언패킹을 하는 방법은 두가지가 있다. 하나는 프로그램을 사용하여 언패킹을 하는 방법이고 다른 하나는 직접 매뉴얼 언패킹을 하는 것이다. 당연히 프로그램을 사용하는 것이 쉬운 방법이지만 매뉴얼 언패킹을 하는 것은 내공을 쌓는데 도움이 많이 될 것이다.

필자가 분석할 악성코드는 UPX 로 패킹이 되어 있었기 때문에 그에 맞는 UPX 프로그램을 이용하여 언패킹을 하였다. 그 후, 언패킹을 하기 전과 언패킹을 하고 난 후의 악성코드 파일을 비교해보았다. 언패킹한 파일은 패킹된 파일보다 용량이 12KB 에서 24KB 로 늘어나있었다. 이는 PE 파일구조와 패킹에 대해 공부하면 쉽게 이해 할 수 있을 것이다.

그 다음 BinText 를 사용해 언패킹한 악성코드 파일의 문자열을 추출해보았다.

```
544E        0  FindFirstFileA
545E        0  FindNextFileA
546E        0  GetCommandLineA
5480        0  GetCurrentProcessId
5496        0  GetDateFormatA
54A6        0  GetDriveTypeA
54B6        0  GetFileSize
54C4        0  GetLocalTime
54D2        0  GetLogicalDriveStringsA
54EC        0  GetModuleFileNameA
5500        0  GetSystemDirectoryA
5516        0  GetTickCount
5524        0  GetTimeFormatA
5534        0  GetTimeZoneInformation
554C        0  GetWindowsDirectoryA
5562        0  GlobalAlloc
5570        0  GlobalFree
557C        0  LocalAlloc
5588        0  LocalFree
5594        0  MapViewOfFile
55A4        0  FindClose
55B0        0  Process32First
55C0        0  Process32Next
55D0        0  ReadFile
55DA        0  ReleaseMutex
55E8        0  SetEndOfFile
55F6        0  SetFileAttributesA
560A        0  SetFilePointer
561A        0  Sleep
5622        0  SystemTimeToFileTime
5638        0  TerminateProcess
```

왼쪽의 그림은 추출한 문자열 중 중요한 부분을 캡쳐해본 것이다. 여러 가지 함수 이름들을 볼 수 있다. 악성코드가 이 함수를 이용하여 실행되지 않을까? 이 함수들을 이용하여 악성코드의 동작방식을 어느 정도 추측할 수 있다.

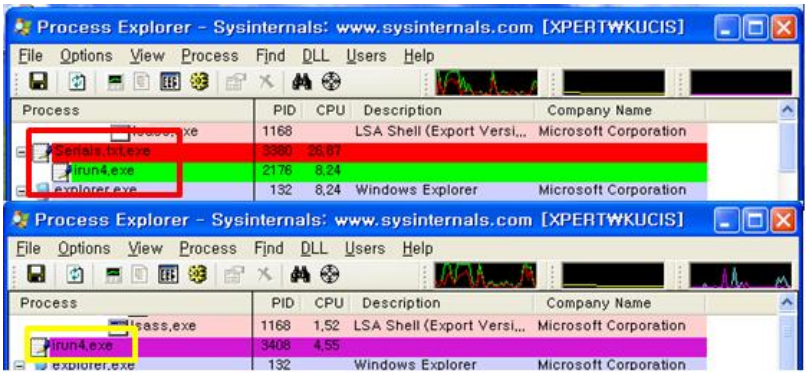
이 외에도 추출한 문자열들 중에는 여러 가지 DLL, EXE 파일의 이름이 나열되어 있다. 악성코드 프로그램이 실행되었을 때 이 DLL, EXE 파일을 사용하거나 이름을 참조한다고 추측할 수 있다. 또한 시작 프로그램에 대한 레지스트리를 조작하는 것으로 추측되는 문자열도 있었으며 P2P 에 공유하기 위해 복사할 파일 명, 이 메일을 보낼 때 사용되는 것으로 추측되는 문자열도 찾아 볼 수 있었다.

나. 동적 분석

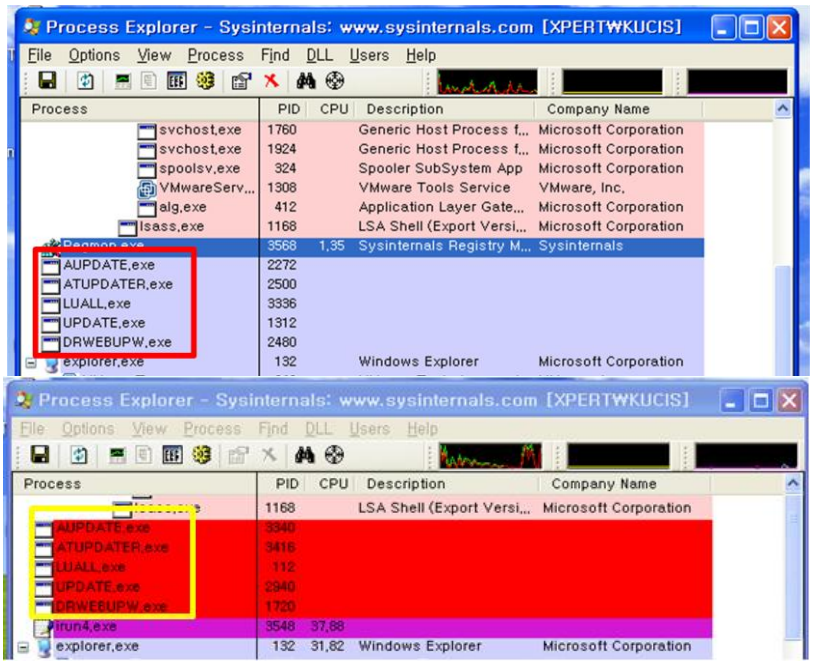
악성코드를 실행시키기 전에 악성코드를 실행할 GUEST머신의 시스템 상황을 Winalysis를 사용하여 스냅샷을 찍어둔다. 이는 악성코드 실행 후 악성코드의 동작 방식을 보다 쉽게 이해하기 위함이다. 이 외에도 앞에 소개했던 모니터링 프로그램들을 실행시킨다.

HOST머신에는 와이어샤크를 실행시켜놓아 GUEST에서 날아오는 패킷을 캡쳐할 수 있도록 한다.

이제 본격적으로 악성코드를 실행시켜보자. 악성코드 실행 후 Process Explorer를 확인해면 악성코드 프로세스를 볼 수 있다.



얼마 지나지않아 'Serials.txt.exe' 프로세스가 'irun4.exe' 프로세스를 생성하였다. 곧 'Serials.txt.exe'가 종료되고 'irun4.exe'프로세스만 남았다.

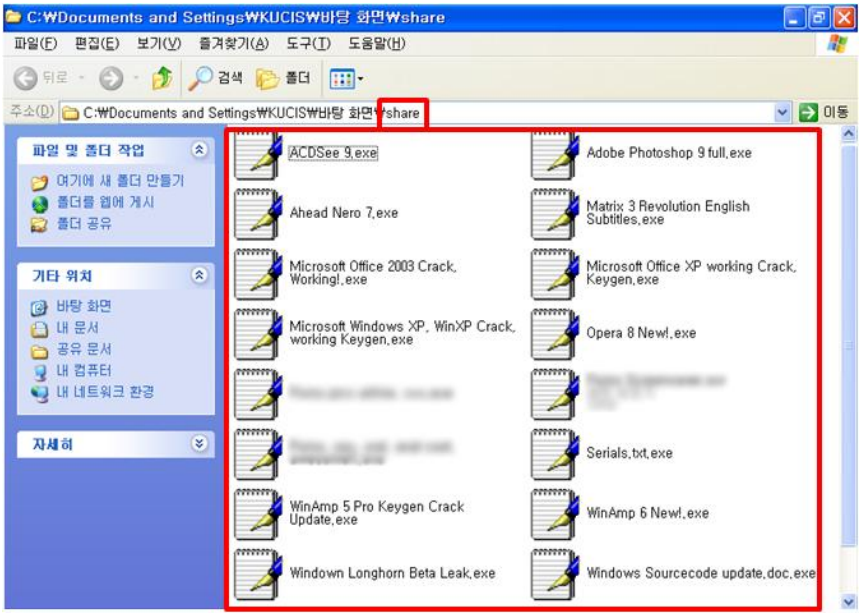


또한 BinText로 추출한 문자열을 바탕으로 특정 이름의 프로세스들을 만들어 실행시킨 후 악성코드를 실행시켜 보았다. 이 임의의 프로세스들은 별다른 행동을 하지 않는 것들이다. 악성코드를 실행시키자 특정 이름의 프로세스들도 얼마 지나지 않아 종료되는 것을 알 수 있었다. (왼쪽의 가장 아래의 그림. 빨간색으로 표시된 부분이 곧 종료된다는 뜻이다.) 이를 토대로 이 악성코드는 업데이트를 하는 것으로 추정되는 프로세스들을 자동으로 종료시키는 것을 알 수 있다. 이 외에도 악성코드가 실행되는 도중에 이 프로세스를 실행시켜보았지만 바로 종료되는 것으로 보아 이 악성코드는 프로세스 목록을 실시간으로 감시한다는 것을 알 수 있다.

악성코드를 실행시키고 어느 정도 시간이 지난 후, 악성코드를 실행시키기 전에 찍었던 스냅샷과 현재 시스템 상황을 Winalysis를 사용하여 비교해본다. Winalysis의 비교상황을 토대로 system32폴더에 들어가보니 'irun4.exe'파일과 'irun4.exeopen'파일이 생성되어 있었다.

생성된 파일을 간단히 분석해보니 처음의 악성코드 파일을 언패킹한 것과 용량, 패킹여부가 같았다. 이 외에도 필자가 간단한 코드 비교를 해본 결과 처음의 악성코드 파일과 'irun4.exe'파일은 같은 파일인 것을 알았다. 나중에 코드분석을 하며 알게 된 사실이지만 'irun4.exe'파일이 처음의 악성코드의 언패킹된 파일과 같은 이유는 실행한 'Serials.txt.exe'파일이 언패킹된 것이기 때문이였

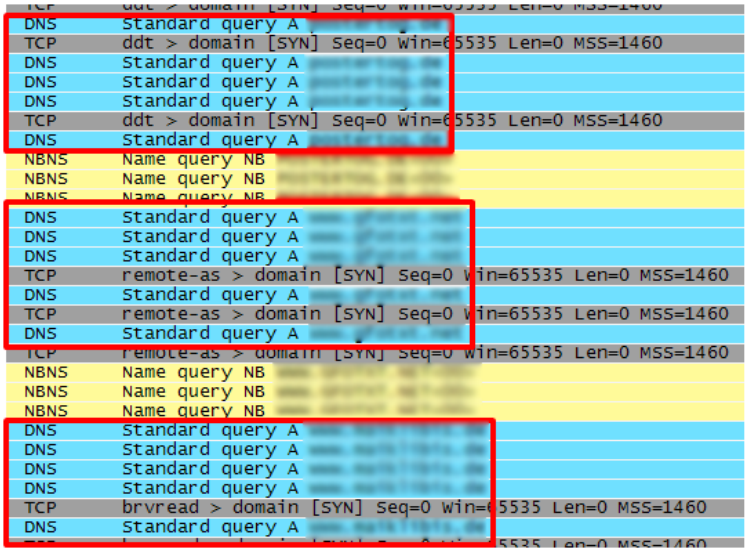
다. 즉, 이 악성코드는 실행된 악성코드 자신을 system32폴더에 복사해 놓는 것이다.



악성코드의 문자열을 추출한 것을 보았을 때 이 악성코드는 P2P와 E-mail을 사용하여 번식하는 것으로 추측되었다. 그래서 악성코드를 실행하기 전 'Share'라는 이름으로 폴더를 하나 생성해 놓았다. 아니나 다를까 오른쪽 그림처럼 특정 이름들의 같은 악성코드 파일들이 생성되었다.

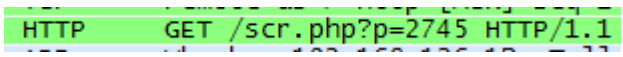
이 외에도 Winalysis를 토대로 Autoruns를 확인해보니 'ssate.exe'라는 이름으로 레지스트리가 시작프로그램에 등록되어있었다. 윈도우를 시작할 때마다 악성코드도 자동으로 실행되게 하기 위함이다.

다시 악성코드의 동작방식으로 돌아가보자. TCPView를 확인해보니 'irun4.exe'가 2745번 포트를 LISTENING하고 있었다. 이는 백도어의 가능성이 있다. 이 외에도 같은 이름으로 1028포트에서 포트를 하나씩 증가시켜가며 SYN을 보내는 것을 발견했다. 동적 분석만으로는 이 행동을 이해할 수 없었으나 후에 코드 분석을 하며 정확한 동작방식을 이해할 수 있었다. 이 이유는 정적 분석 때 알아보도록 하자.



이 악성코드가 보내는 패킷은 이뿐만이 아니였다. 특정한 3개의 사이트로 접속을 시도하였다. 왼쪽의 그림이 HOST머신의 와이어샤크에서 캡쳐한 모습이다.

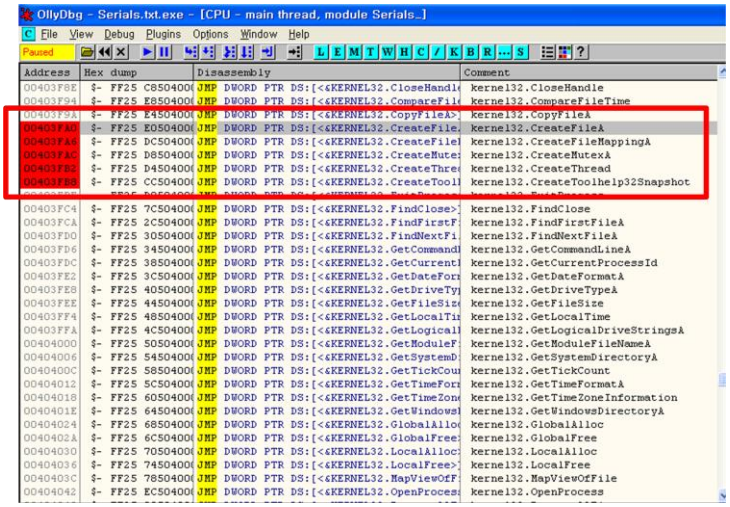
이 사이트에 접속하여 어떤 동작을 하는지 자세히 알아보기 위하여 HOST머신의 호스트파일을 변조하고 다시 악성코드를 실행해보았다.



와이어샤크를 관찰한 결과, 악성코드는 각 홈페이지에 GET방식으로 오른쪽 그림과 같이 scr.php를 요청하고 있었다. 하지만 시간을 두고 기다려보아도 별다른 결과가 나오지 않았다. 이

후 직접 이 세 개의 사이트에 접속해보았지만 현재 전부 접속이 되지 않아 더 이상 이 사이트에 관한 분석은 진행할 수 없었다.

다. 정적 분석



앞의 분석들을 토대로 OllyDbg를 사용하여 언패킹한 악성코드 파일의 코드분석을 해보았다.

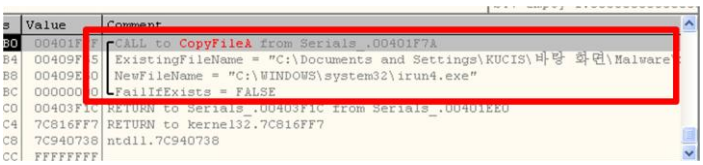
OllyDbg를 실행시켜보니 여러 함수들이 보인다. 왼쪽의 그림과 같이 'CreateFileA'등의 중요하다고 생각되는 함수이름들에 브레이크포인트를 설정하고 실행시켜보았다. 아래는 코드를 분석하면서 알게 된 동작 방식들이다.

1) 레지스트리 등록

처음 악성코드가 하는 일은 'RegCreateKey'와 'RegSetValueEx', 'RegCloseKey'라는 함수를 사용하여 레지스트리를 등록하는 것 이였다.

2) 파일 복사

두 번째로 악성코드는 'CopyFile' 함수를 사용하여 system32 폴더에 자신을 복사하였다. 오른쪽 그림은 'CopyFile'함수를 실행할 때의 스택 모습이다.



그리고 자신을 복사한 후 이 프로그램은 끝이 났다. 계속 분석을 하기 위해 system32폴더에 생성된 'irun4.exe'파일을 OllyDbg로 실행시킨 후 분석을 계속하였다. 실행시킨 후 처음은 'Serials.txt.exe'와 동작방식이 같았다.

3) 'shar' 폴더에 파일 생성

'FindFirstFile', 'FindNextFile'함수를 사용하여 전체 디렉토리와 파일을 하나하나 찾아나가다가 디렉토리를 발견하면 'StrStr'함수를 사용하여 'shar'이란 단어와 디렉토리 명을 비교, 이 단어가 포함되어 있으면 ②와 같이 'CopyFile'함수를 사용하여 'Serials.txt.exe'와 같은 이름들로 해당 디렉토리에 자신을 복사한다.

4) Mutex 생성

Mutex란 "MUTual Exclusion"의 약자로 '상호 배제'라고 한다. 이는 데드락을 방지하기 위해 스레드들이 단독으로 실행되게 하는 기술이다. 이 악성코드도 'CreateMutex'라는 함수를 사용하여 Mutex를 생성한다.

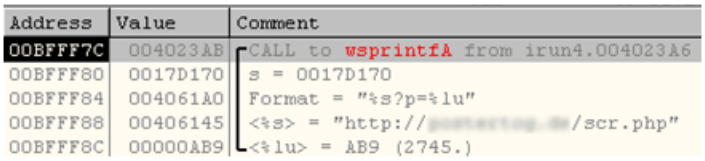
5) 2745포트 listen

동적 분석단계에서 보았듯이 이 악성코드는 2745포트를 listen한다. 'socket'과 'listen'등의 함수를

사용하여 2745번 포트를 open하고 listening모드로 변경한다.

6) URL접속

'wsprintf'와 'InternetGetConnectedState', 'InternetOpen', 'InternetOpenUrl'등의 함수를 사용하여 특정 URL로 접속을 시도한다. 오른쪽 그림은 'wsprintf'함수를 실행할 때의 스택의 모습이다. 인자 값을 보면 알 수 있듯이 p값을 이용하여 scr.php 웹페이지를 요청한다.



7) SYN_SENT

앞의 ③에서 'FindFirstFile', 'FindNextFile'을 사용하여 파일과 디렉토리의 이름을 차례로 검색, 비교한다고 하였다. SYN을 보내는 과정에서도 같은 방법을 사용한다. ③에서는 디렉토리를 검색했던 것과 달리 이번에는 파일의 확장자를 하나씩 비교한다. 'FindFirstFile', 'FindNextFile'로 파일을 발견하면 'StrStr'을 사용하여 파일의 확장자를 비교한다. 이때 비교하는 확장자는 [.wab .txt .htm .xml .cfg .asp .php .pl .cgi]등이다. 만약 같은 확장자를 찾으면 파일을 읽은 후 URL을 검색하여 접속을 시도한다.

8) Process 강제 종료

'Process32First', 'Process32Next'함수를 사용하여 실행되고 있는 Process의 목록을 실시간으로 검색한다. 만약 'UPDATE.EXE', 'AUPDATE.EXE'등과 같은 특정한 이름의 프로세스를 발견하면 'OpenProcess', 'TerminateProcess'함수를 사용하여 프로세스를 강제 종료시킨다.

5. 결과 및 복구 방법

가. 결과

지금까지 분석해 온 악성코드의 동작방식을 요약하면 다음과 같다.

1) "C:₩WINDOWS₩system32"에 'irun4.exe'파일을 생성하고 'shar'폴더에 같은 악성코드 파일을 복사한다.
2) 레지스트리의 시작파일에 자신을 등록한다.
3) 2745포트를 listen하며 백도어 등을 가능하게 한다.
4) (악성코드파일이 있는 곳으로 추측되는) URL로 접속 시도한다.
5) DNS로 1028포트부터 하나씩 증가시키며 3번씩 SYN을 보낸다.
6) 특정 이름의 프로세스를 검사하며 강제 종료시킨다.

나. 복구 방법

1) 레지스트리 복구
   "HKCU₩Software₩Microsoft₩Windows₩CurrentVersion₩Run"의 'ssate.exe' 값 삭제
2) 악성코드 파일 삭제
   "C:₩WINDOWS₩system32"의 'irun4.exe, 'irun4.exeopen' 삭제
   'shar' 폴더의 악성코드 파일삭제

6. 참고 문서

- ✓ 차민석 / '악성코드의 역사' / 안철수 연구소 / 2007-01 ~ 2008-09
- ✓ 'Monthly 사이버 시큐리티' 9월호, 10월호 / 국가 사이버 안전센터 / 2008-09, 2008-10
- ✓ 대학정보보호동아리 교육 / 'Reverse Engineering' / KISA, KUCIS / 2008-08
- ✓ 이강석 / 'Email-Worm Analysis' / 어셈러브 / 2006-12
- ✓ 네이버 백과사전, 위키피디아