기술문서	`07. 10. 21 작성

악성코드 분석 방법론

2007. 10. 21

동명대학교 정보보호동아리 THINK

태인규 graylynx at gmail.com

1. 개요

누구든지 한번쯤은 뉴스나 미디어를 통해 악성코드에 대해 들어봤을 것이다. 컴퓨터를 잘 다루지 못하는 컴맹들도 악성코드가 무엇인지 알고 있을 만큼, 더 이상 악성코드는 남의 이야기가 아니다. 이는 악성코드가 우리 실생활에 아주 밀접해졌고, 그만큼 우리의 IT환경이 위협받고 있다는 이야기이다.

그럼 무엇이 악성코드인가? 인터넷 백과사전에서 검색해보면, 악성코드란 악의적인 목적을 위해 작성된 실행 가능한 코드의 통칭으로 자기 복제 능력과 감염 대상 유무에 따라 바이러스, 웜, 트로이목마 등으로 분류된다[1]라고 정의하고 있다.

예전 도스시절 사용자에게 해를 끼치는 프로그램은 바이러스가 전부였을 때, 그때는 악성코드라는 개념이 없었다. 하지만 세월이 지나고 IT 환경이 발달하면서 신기술들이 매일 홍수 터지듯 넘쳐났고, 다들 신기술을 개발하느라 여념이 없을 때, 그 취약한 부분을 파고들며 악성코드는 발전하기 시작했다.

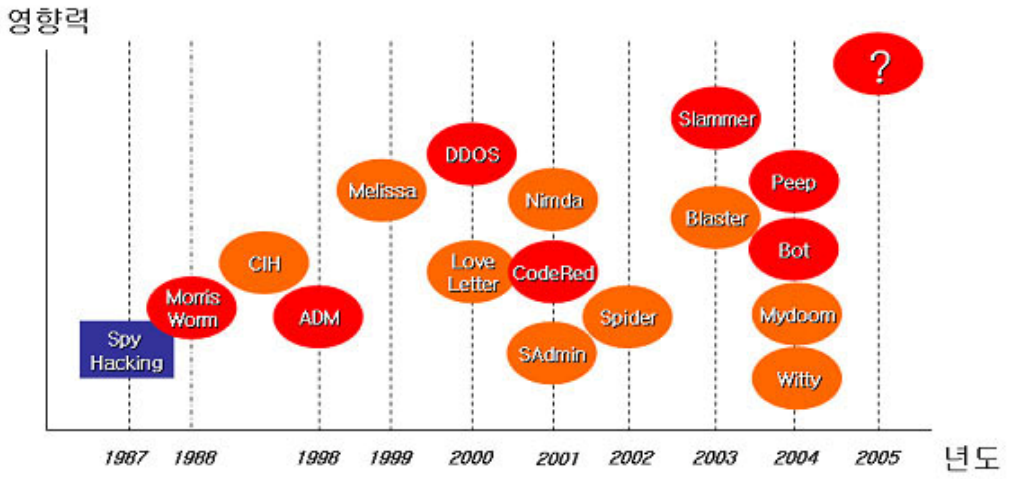

그림1. 년도별 웜(악성코드) 발전 그래프[2]

특히 우리나라는 발 빠르게 IT 인프라를 보급하며 세계 최고의 IT 국가로 올라섰지만, 한편으로는 그로 인해 악성코드에 훨씬 더 많이 노출되는 결과를 초래했고, 이는 악성코드가 번식하기에 최적의 환경을 제공해주었다. 결국 2003년 1월 25일 그 유명한 1·25 인터넷 대란[3]이 터졌고, 이제 보안은 국가차원의 문제가 되었다.

---

[1] 네이버 백과사전, 『악성코드』

[2] 아이뉴스24, 『제2의 인터넷 대란 대비하자』, 2005.

[3] 위키백과, 1·25 인터넷 대란은 2003년 1월 25일 대한민국 대부분의 인터넷 망이 마비된 사건이다. 마이크로소프트사 SQL 서버 헛점을 이용하는 슬래머 웜 이 일으켰다. 이 사건으로 피해를 입은 인터넷 사용자들은 한국통신을 상대로 피해보상소송을 제기하기도 하였다.

허나 다행인건 인터넷 대란의 충격 때문인지, 이후 많은 정보보호 회사들과 국가기관이 생겨났으며, 지금 그들은 우리나라 IT 인프라를 외부로부터 보호하기 위해 혼신의 노력을 다하고 있다. 덕분에 우리가 그들의 보호를 받으며 별탈 없이 IT 생활을 영위할 수 있는 것이다.

하지만 이런 노력에도 불구하고, 우리가 악성코드로부터 완전히 자유로울 수 없는 이유는, 하루에도 수천 개씩 쏟아지는 악성코드를 감당하기엔 그 대응 인력이 턱없이 모자라기 때문이다. 이는 곧 제2의 인터넷 대란을 유발할 수 도 있는 잠재적 위협으로 연결된다.

그림2. 최근 1년간 월별 악성코드 감염사고 발생 추이[4]

악성코드로부터 자유롭기 위해서는 각 사용자들의 주의와 관심이 최고의 방법이겠지만, 일반적으로 대부분이 컴퓨터에 대해 자세한 지식을 가지고 있지 못한 사람이므로, 꾸준하게 사용자들에게 그 내용을 홍보하는 것 이외에 별다른 도리가 없다.

그럼 또 어떤 해결책이 있을까? 악성코드 분석 인력을 늘려서 그만큼 악성코드를 빨리 분석하고 대응하는 것이다. 하지만 실제 우리나라에서 활동하고 있는 악성코드 분석 인력은 그 수가 아주 적은데, 그것은 악성코드 분석을 하기 위해 알아야 하는 고급 기술의 내용이 워낙 깊고 방대하다는 것에 기인한다.

특히 악성코드 분석에 필요한 핵심 기술인 리버스 엔지니어링(Reverse Engineering)[5] 기술이 일반인들에게는 매우 생소한 분야라 그만큼 처음 접근하기가 어려우므로, 많은 사람들이 힘들어하는 것 같다. 하지만 이것도 첫 단추 꿰는 것이 힘들 뿐, 한번 맛을 보고 나면 더 이상 어렵지 않은 기술임을 알게 될 것이다.

[4] 국가사이버안전센터, 『월간 사이버위협 동향 및 대응활동』, 2007

[5] 네이버 백과사전, 소프트웨어 공학의 한 분야로, 이미 만들어진 시스템을 역으로 추적하여 처음의 문서나 설계기법 등의 자료를 얻어 내는 일을 말한다.

이 글에서 필자는 악성코드를 분석하기 위한 리버스 엔지니어링의 시작에 대해 설명하려 한다. 많은 사람들이 궁금해 하는 것들, 이를테면 어떤 도구로 어떤 책을 보고, 어떠한 방법으로 분석을 하며 결과는 어떻게 마무리 지어야 하는가? 등에 대한 답변을 적어보려 한다.

2. 지식

악성코드를 분석하기 위해 필요한 지식은 다음과 같다.

가. 해당 운영체제에 대한 지식

먼저 해당 운영체제에 대한 지식이라 함은, 악성코드가 번식하는 해당 운영체제가 작동하는 원리와 구조에 대한 지식이라 할 수 있다. 운영체제가 어떤 하드웨어를 기반으로 동작하며, 어떤 API[6]들을 제공하며, 시스템 주요정보가 저장되는 위치는 어디인지 등에 대한 세부 지식을 말한다.

악성코드도 하나의 프로그램이다. 다른 프로그램들과 똑같이 대상 운영체제가 지원하는 자원을 빌려 쓰며, 그 운영체제 내에서만 존재가 가능하다. 그렇게 때문에 우리는 악성코드를 분석하기에 앞서 해당 운영체제에 대해 알아야 할 필요가 있는 것이다.

그럼 어떤 운영체제를 공부해야 할까? 최근 거의 대부부분의 악성코드는 MS Windows 시리즈 운영체제를 대상으로 만들어지고 있는데, 그 이유는 일반 PC용으로 Linux등의 타 운영체제보다 Windows의 사용자가 압도적으로 많기 때문이다.

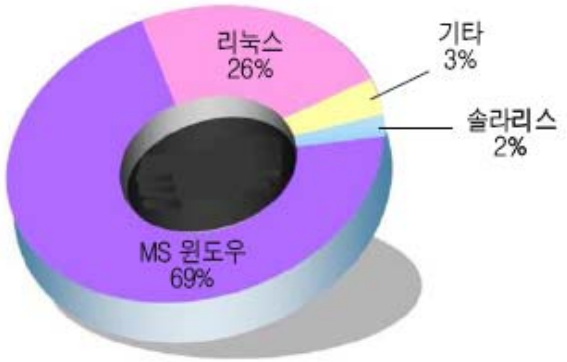

그림3. 운영체제별 침해사고 현황[7]

그러므로 악성코드를 분석하기 위해서는 Windows 운영체제에 대해 잘 알아야 한다. 하지만 여기서 잘 안다는 건 일반적인 사용용도로서의 지식이 아닌 악성코드가 운영체제를 악용하는데 쓰이는 기술을 파악하기 위한 지식이다.

---

[6] 네이버 백과사전, 운영체제와 응용프로그램 사이의 통신에 사용되는 언어나 메시지 형식을 말한다.

[7] 국가사이버안전센터, 『월간 사이버위협 동향 및 대응활동』, 2007.

아래는 필자의 추천 도서이다.

1) 운영체제론 (ANDREW S.TANENBAUM 지음 / 사이텍미디어)
2) Programming Windows (Charles Petzold / 한빛미디어)
3) Windows 시스템 실행파일의 구조와 원리 (이호동 지음 / 한빛미디어)
4) Windows 구조와 원리 (정덕영 지음 / 한빛미디어)
5) Microsoft Windows Internals (Mark E. Russinovich 지음 / 정보문화사)

물론 위의 책을 무조건 다 읽고 알아야 악성코드를 분석할 수 있다는 건 아니다. 위의 책들 중 한 권도 읽지 않은 사람이라도 어느 정도 분석은 가능하다. 하지만 악성코드를 완벽히 분석하기 위해서는 반드시 숙지해야 할 책들이다. 또한 악성코드를 분석하다 보면, 종종 필요한 정보들을 찾아야 할 때가 많은데, 그때에 아주 도움을 주는 좋은 내용을 많이 담고 있다. 가능하면 한번씩은 꼭 읽어보도록 하자.

나. 어셈블리 언어에 대한 지식

다음으로, 어셈블리 언어[8]에 대한 지식이다. 우리가 접하는 악성코드가 프로그램 소스일 가능성은 0% 이다. 모든 악성코드는 그 실행파일만으로 전파되며 우리는 그것을 분석해야 한다. 즉 이미 컴파일 된 바이너리만 가지고 그 소스를 역으로 분석할 수 있어야 한다는 것이다. 바로 이 기술을 리버스 엔지니어링이라 한다. 어셈블리 언어는 컴퓨터가 사용하는 기계어와 1대1 대응이 되므로, 컴퓨터로 실행되는 파일을 역으로 어셈블리 언어로 바꾸는 것이 가능하다. 이를 디스어셈블(Disassemble)이라 한다. 어셈블리 언어를 알면 디스어셈블된 코드를 토대로 악성코드의 소스를 복원할 수 있으며, 이것이 분석에 있어서 가장 어렵고 중요하다고 할 수 있다.

아래는 필자가 추천하는 도서이다.

1) 80x86 마이크로프로세서 (John Uffenbeck 지음 / 홍릉과학출판사)
2) IA-32 Intel Architecture Software Developer's Manuals (Intel 홈페이지)
3) 어셈블리언어 (KIP R. IRVINE 지음 / 교보문고)

하지만 어셈블리 언어를 자유자제로 구사할 만큼 잘 할 필요는 없다고 생각한다. 어셈블리 언어는 꼭 알아야 할 지식임에 틀림없지만, 아직도 어셈블리 언어를 사용하여 프로그래밍 하는 사람은 거의 없을 것이다. 분석의 측면에서도 그렇다. 단지 필요한 건 분석하는데 필요한 지식일 뿐이다. 단지 어셈블리 코드를 보고 이것이 어떤 행동을 하는 것인가 분석할 수만 있으면 그만이다.

[8] 기계어와 1대1로 대응하는 언어로 사람이 이해하기 쉽게 약간 변형시킨 언어이다. 기계어와 가장 가깝기 때문에 그 기계의 특성을 가장 잘 살릴 수 있다는 것이 장점이다. 하지만 기계어와 매우 비슷하기 때문에 이해하기 어렵다는 것이 단점이다.

다. 네트워크에 대한 지식

마지막으로 네트워크에 대한 지식이다. 예전 도스시절 바이러스를 분석할 때는 전혀 필요 없었던 지식이지만, 인터넷이 발달하여 전세계 모든 컴퓨터들이 거미줄처럼 엮여 서로 통신하는 지금은, 웜[9]의 출현으로 리버스 엔지니어링만큼 중요한 지식이 되었다.

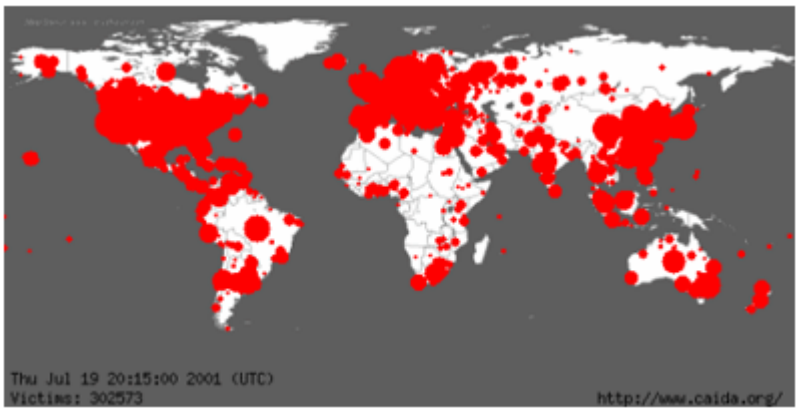

그림4. 코드레드 웜 감염 위치[10]

모든 국가기관, 단체, 은행, 기업, 가정의 컴퓨터들이 서로 네트워크로 연결되어 있다. 자칫 잘못해서 네트워크가 마비되기라도 한다면 사회적 대공황으로 이어질 수도 있을 만큼 네트워크는 IT 그 자체라고도 볼 수 있다. 2001년 코드레드 웜[11]을 보라. 단지 수백 바이트의 작은 패킷이 우리나라 네트워크 전체를 과부하로 정지시켰다. 이는 분명 네트워크의 관점에서 분석되어야 한다. 리버스 엔지니어링이 미시적 분석 방법이라 한다면, 네트워크 분석은 거시적 분석 방법이라 할 수 있겠다.

역시 다음의 서적을 추천한다.

1) Unix Network Programming (W. Richard Stevens 지음 / Prentice Hall)
2) Network Programming for Windows (Anthony Jones 지음 / 정보문화사)
3) 성공과 실패를 결정하는 1%의 네트워크 원리 (Tsutomu Tone / 성안당)

위에서 필자가 소개한 책만 봐야 할 필요는 절대 없다. 단지 일반적으로 가장 많이 추천하는 책 위주로 직접 읽어본 책만 서술하였다. 자신이 읽어보지도 않은 책을 남에게 추천할 수는

[9] 위키백과, 스스로를 복제하는 컴퓨터 프로그램이다. 컴퓨터 바이러스와 비슷하다. 하지만 컴퓨터 바이러스와 웜의 중요한 차이점은 바이러스는 스스로 전달할 수 없지만 웜은 가능하다는 점이다. 웜은 네트워크를 사용하여 자신의 복사본을 전송할 수 있으며, 어떠한 중재 작업 없이 그렇게 할 수 있다. 일반적으로 웜은 네트워크를 손상시키고 대역폭을 잠식하지만, 바이러스는 컴퓨터의 파일을 감염시키거나 손상시킨다. 바이러스는 보통 네트워크에 영향을 주지 않으며 대상 컴퓨터에 대해서만 활동한다.

[10] http://www.caida.org

[11] 컴퓨터 프로그래머들이 좋아하는 카페인 음료의 이름을 딴 것으로 웹 서버 운영체제인 윈도 NT와 윈도 2000의 취약점을 찾아 공격하는 웜 바이러스.

없는 노릇 아닌가? 자신이 부족하다 싶은 부분은 더 많은 책을 구입해서 닥치는 데로 읽는 것이 실력향상의 최단 지름길이라 생각한다.

3. 도구

그럼 악성코드를 분석하기 위해 필요한 도구에 대해 알아보도록 하자.
분석 도구는 다음과 같이 크게 세 가지로 나눌 수 있다.

가. 분석에 도움을 주는 도구

분석에 도움을 주는 도구란 분석을 하는데 직접적인 역할을 하지 않지만, 분석을 더욱 편하게 도와주거나, 분석환경을 만들어 주는 도구라 할 수 있다.

악성코드가 어떤 동작을 하는지 파악하려면 좋든 싫든 악성코드를 직접 실행시켜봐야 한다. 과연 실행이 되면 운영체제에서는 어떤 일이 일어나고 네트워크에는 어떤 변화가 일어나는가에 대한 정보가 필요한데, 그러면 실행시킬 대상이 있어야 한다. 제약회사에서 새로운 바이러스가 출현했을 때, 쥐들에게 바이러스를 투입하여 어떤 증상이 나타나는지 조사하는 것에 비유할 수 있겠다. 하지만 이렇게 하루에 수천 개씩 새로 생겨나는 악성코드들을 분석했다간, 윈도우즈를 다시 설치하는데 더 많은 시간이 필요할 것이다. 아주 비효율적이다.

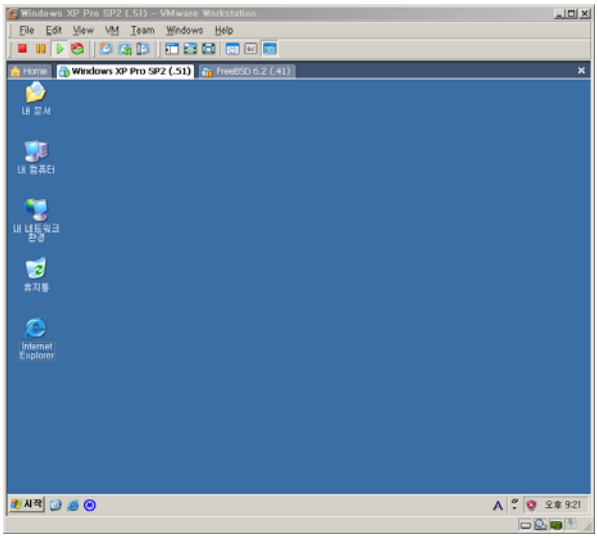

그림5. VMware Workstation[12]

그렇기 때문에 실제 악성코드를 분석할 때는 실제 PC가 아닌, 그것을 대신할 가상의 PC에서 악성코드를 실행시킨다. 여러분들도 다들 알고 계시는 가상 PC 프로그램[13]들이다.

[12] http://www.vmware.com

[13] 컴퓨터 내에서 각각 해당 프로그램을 실행하는 여러 운영 체제를 동시에 실행할 수 있다. 가상 PC 프로그램에는

최근에는 그 기능이 아주 좋아져서 진짜 PC와 거의 동일한 기능을 지원함은 물론이고, 악성코드에 감염이 되었다 하더라도, 마우스 클릭 한번이면 악성코드가 실행되기 전 시점으로 감쪽같이 돌아갈 수도 있다. 이런 기능들을 이용해서 각 운영체제 별로 악성코드에 대해 몇 번씩이나 재실행이 가능한 것이다.

나. 행동기반 분석 도구

행동기반 분석 도구란, 위에서 설명했듯이 실제 악성코드를 실행시켜보고 실행되기 전과 실행된 후를 비교하여, 그 행동 패턴을 분석하는 도구라 할 수 있다.

주로 악성코드가 이용하는 자원들에 대한 변화를 탐지하게 되는데, 일반적으로 프로세스 목록, 보조 기억장치, 레지스트리, 네트워크 아답터에 대해 로깅을 실시한다.

다른 말로 이들을 모니터링 툴이라고도 하는데, 지금은 MS에 인수된 SYSINTERNALS[14]라는 회사에서 나온 툴이 많이 쓰인다. 그 목록은 다음과 같다.

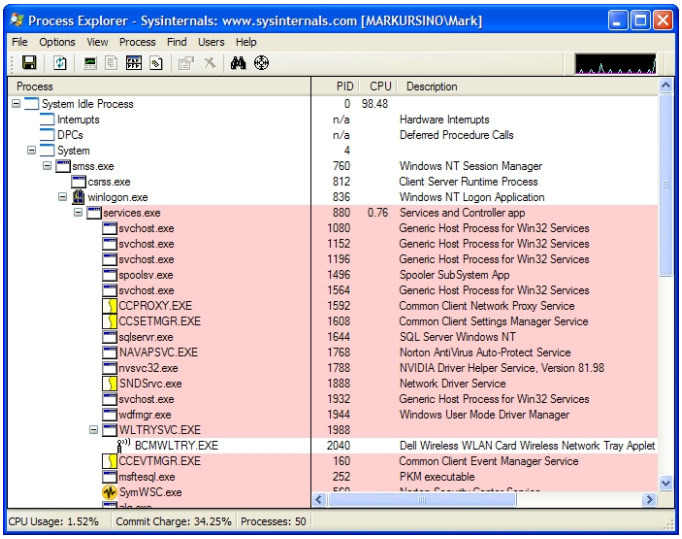

그림6. ProcessExplorer

1) Autoruns	윈도우즈 시작 시 자동으로 실행되는 프로그램 감시
2) Filemon	대상 프로그램이 읽거나 쓰는 파일 감시
3) Regmon	대상 프로그램이 읽거나 쓰는 레지스트리 감시
4) RootkitRevealer	루트킷 탐지 프로그램
5) Tcpviews	네트워크를 통해 주고 받는 TCP 패킷 감시
6) Tdimon	현재 시스템의 모든 TCP/UDP 입출력 상황 감시

VMware, Virtual PC, Boch 등이 있다.

[14] http://www.microsoft.com/technet/sysinternals/utilitiesindex.mspx

7) ProcessMonitor 프로세스 목록 감시
8) ProcessExplorer 프로세스 목록 감시 (기타 추가 기능 지원)

또한, 다음의 툴들[15]도 많이 쓰인다.

9) IceSword 숨겨진 프로세스 목록을 탐지
10) MultiMon 위의 모든 대상에 대한 통합 실시간 감시
11) Regshot 레지스트리 분석도구 (실행전과 후의 스냅샷 비교)
12) Smartsniff 간단한 네트워크 패킷 스니퍼
13) Winalysis 위의 모든 대상에 대해 통합 분석 (스냅샷 비교)

그림7. IceSword

악성코드들은 기술이 날로 발전하여, 최근의 거의 모든 악성코드들은 스스로를 외부로부터 보호하는 실행압축[16]이나 안티-디버깅[17] 등의 코드를 포함하고 있다. 또한 자신이 실행되는 곳이 실제 물리적 머신인지, 가상의 머신인지 파악하는 코드가 들어있어서 가상 PC 프로그램 안에서 실행되었을 때 아무런 동작도 취하지 않는 악성코드들도 출현하기 시작했다. 간혹 이런 악성코드들은 위의 도구들을 탐지할 때가 있는데, 이 때엔 다음에 설명

[15] 여기에 있는 모든 도구들은 google에서 검색해보면 쉽게 찾을 수 있다.
[16] 국가사이버안전센터, 『악성코드 분석을 위한 "실행압축" 해제 기법』, 흔히 말하는 ZIP, RAR처럼 데이터들을 하나로 묶어 놓는 압축과는 달리 그 대상이 notepad.exe처럼 실행할 수 있는 파일을 압축한 것으로, 실행압축 된 notepad.exe는 압축을 푸는 과정 없이 바로 프로그램을 실행할 수 있다. 악성코드도 이런 점을 이용해 짧은 시간 안에 많은 곳으로 전파되도록 실행압축을 사용하고 있으며, 백신 제작자들로 하여금 악성코드를 분석하기 어렵도록 하는데도 사용된다. 분석하는데 시간이 걸리는 동안 악성코드 전파 시간을 늘릴 수 있기 때문이다. 이와 반대로 만약 실행압축을 빨리 해제하여 분석할 수 있다면, 그 피해규모 또한 현저히 줄일 수 있을 것이다.
[17] 프로그램 단독으로 실행됐을 때와, 디버거 위에 실행됐을 때의 시스템적인 차이를 이용하여 자신이 디버깅되고 있는지 파악하는 기술.

할 코드기반 분석 툴과 리버스 엔지니어링 기술로 그 알고리즘을 분석한 뒤 보호기능을 무장해제시키고 행동 기반 분석 툴로 다시 분석해야 할 필요가 있다.

다. 코드기반 분석 도구

마지막으로 코드기반 분석 도구들을 살펴보도록 하겠다. 코드기반 분석 도구는 크게 적극적 또는 비적극적의 성격으로 나눌 수 있다.[18]

1) 비적극적 코드기반 분석 도구

비적극적 분석 도구란 일반적으로 코드를 기반으로 분석하되, 분석자의 노력 또는 경험이 필요하지 않는 도구로서, 대표적인 도구로는 ClamAV[19]와 Strings[20]가 있다. 이들 도구는 악성코드를 세밀하게 분석할 수는 없지만, 간단하게 악성코드에 대한 요약정보를 얻을 수 있다는 장점이 있다. 그래서 코드기반 분석 초기에 필수적으로 실행하는 것이 보통이다.

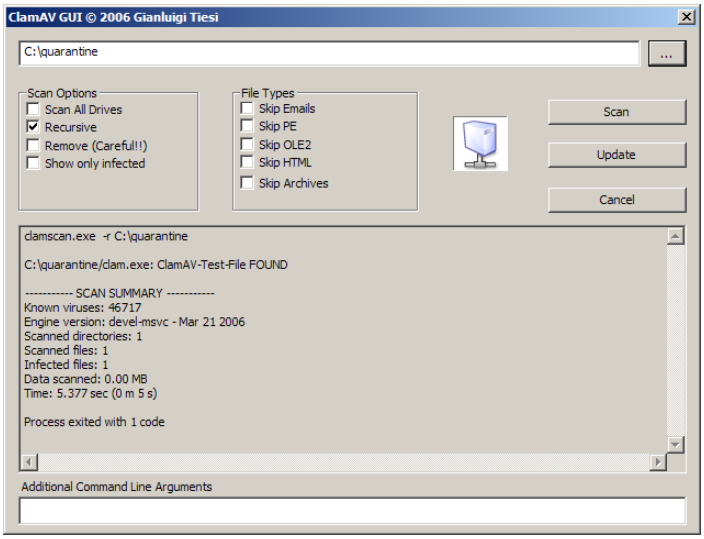

그림8. ClamAV GUI

2) 적극적 코드기반 분석 도구

적극적 분석 도구란, 분석자의 노력 또는 경험을 기반으로 악성코드의 세세한 부분까지 분석하는데 필요한 도구로서, 디버거[21]와 디스어셈블러[22]가 있다.

---

[18] 필자 나름대로의 기준으로 봤을 때 마음대로 지어낸 기준이니 무시해도 좋다.

[19] 주로 이메일로 전파되는 악성코드를 관문에서 차단하기 위해 개발된 소프트웨어이다. 시그니쳐 기반의 악성코드 판단 도구로, 이미 특정 악성코드를 분석한 사람이 특정 악성코드라고 판단할 수 있는 최소한의 바이너리(악성코드의 DNA쯤 되겠다)를 인터넷에 올리면, 다른 사람들은 단지 그 시그니쳐만 받음으로서 악성코드를 파악할 수 있게 되는 원리이다.

[20] 바이너리에서 아스키코드만 출력해주는 도구 (3글자 이상의 출력 가능한 문자로 시작하고 NULL 값으로 끝나는 문자열을 출력한다)

[21] 실시간으로 대상 프로그램의 코드를 추적하는 프로그램.

[22] 기계어로 이루어진 실행파일을 사람이 알아볼 수 있는 어셈블리어로 바꾸는 프로그램.

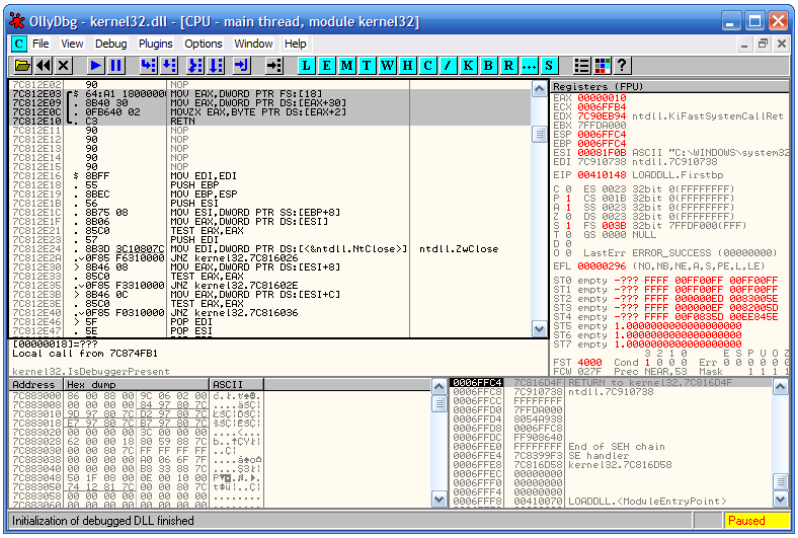

그림10. OllyDbg

주로 쓰이는 디버거로는 SoftIce[23], OllyDbg[24], WinDbg[25] 등이 있으며, 디스어셈블러에 비해 실시간으로 실행코드를 추적하면서 악성코드의 세세한 부분까지 분석할 수 있다는 장점이 있다. 또한 디스어셈블러로는 IDA[26]와 W32Dasm[27]이 있으며, 특히 IDA는 현존 최고의 디스어셈블러로 손꼽힐 만큼 최고의 분석 기능을 제공한다.

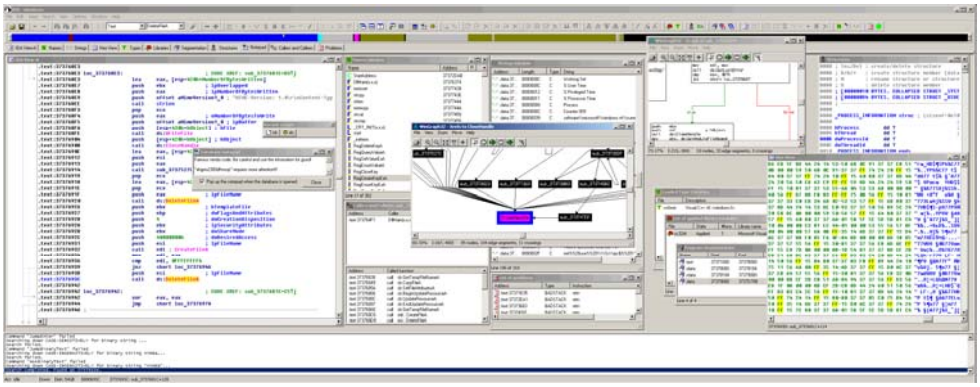

그림11. IDA Pro

하지만, 명심해야 할 것은 적극적 분석 도구는 그 도구 자체로는 아무런 역할도 할 수 없다는 것이다. 아무리 도구가 뛰어나다 한들, 분석가의 능력과 경험에 의해 결과물의 수준은 천차만별로 차이가 난다. 피아니스트에 비유하자면, 세계 최고의 피아노

[23] 몇 년 전까지만 해도 Ring0 기반 최고의 디버거였지만, 지금은 개발이 중단되었다.

[24] http://www.ollydbg.de

[25] MS에서 제공하는 커널 디버거 (http://www.microsoft.com/whdc/devtools/debugging/default.mspx)

[26] http://www.datarescue.com/idabase/

[27] 요즘엔 거의 쓰지 않는다.

가 있다 한들, 그 피아노를 아름답게 칠 능력이 없으면 그 피아노는 그저 그런 나무 토막에 불과한 것과 같은 이치이다.

4. 절차

이제 무엇을 공부해야 하는지, 어떤 도구를 이용해야 하는지 알게 되었다. 그럼 다음으로 필요한 것은 무엇일까? 바로, 어떻게 분석하나 정도가 아닐까? 그럼 먼저 어떻게 분석해야 하는지 큰 절차를 알아보고, 실제로 악성코드 샘플을 예로 분석하면서 세세한 절차를 알아보도록 하자.

악성코드를 분석하기 위한 큰 절차는 다음과 같다.

가. 허니팟 프로젝트[28]나 악성코드 연구 홈페이지 등을 통해 악성코드 샘플을 수집한다.

나. ClamAV로 분석해보고, 이미 분석된 악성코드라면 어떤 행동을 하는지 미리 파악한다. (하지만 변종 악성코드 또한 같은 시그니쳐를 가질 수 있으므로, 대충 이러할 것이다 라고만 생각하고 결과를 완전히 신뢰하지 않는 것이 좋다)

다. 앞서 소개한 행동기반 분석 도구들을 통해 가상Box안에서 악성코드를 실제 실행시켜보고, 어떤 행동을 하는지에 대해 기록한다. 이 때 최대한 많은 데이터를 수집하는 것이 중요한데, 이는 다음에 있을 세부적인 분석을 하는 데에 매우 도움이 된다.

라. PEiD[29] 등의 PE헤더[30] 분석 툴을 통하여, 악성코드가 자동압축 되어있는지 검사한다.

마. 만약 자동압축 되어있다면, 언팩킹[31] 한 뒤, 다음으로 넘어간다.

바. Strings로 어떤 문자열이 쓰이는지, 어떤 API들이 사용되는지 조사한다.

사. 디스어셈블러로 구한 어셈블리 코드를 토대로 악성코드의 루틴을 머릿속으로 그려본다.

아. 앞서 분석한 악성코드의 루틴을 따라, 디버거로 실제 실행코드를 실시간으로 추적한다. (이 부분에서 디스어셈블러로는 알아내지 못한 코드들이 나타날 가능성이 높다. 많은

[28] 야후 용어사전, 간단히 '해커 잡는 덫'이란 뜻의 용어. 즉 해커를 잡는 유혹의 꿀단지라는 의미이다. 해커 공격에 대응할 수 있는 시간을 벌고 해커의 움직임을 면밀히 파악함으로써 사이버 테러를 방지하는 신기술로 관련업계에서는 기대를 모으고 있다.

[29] 시그니쳐기반 자동압축 탐지 도구. 대상 실행파일이 어떤 방식으로 자동 압축되어 있는지 알려준다.

[30] Portable Executable Header. MS Windows에서 사용되는 COFF 기반의 실행파일 포맷.

[31] 자동압축을 강제 해제하는 것을 말한다.

악성코드들이 자신의 중요한 정보는 실시간으로 숨겨놓고 꺼내보기 때문이다)

자. 분석을 통해 구한 모든 데이터를 체계적으로 문서화 한다. (중요)

차. 악성코드에 대응하고 치료하는 방법을 다른 사람과 공유한다. (중요)[32]

카. 가. 절차로 돌아간다.

위의 절차는 순전히 필자의 경험에 기반한 제멋대로 절차 이므로, 무조건 신뢰할 필요는 없다. 그저 공부 목적으로 깨작깨작 분석해본 부족한 이의 경험이 어찌 수년 동안 실무에서 구르고 구른 분들의 노하우를 따라갈 수 있으리..

그럼 이제 실제로 악성코드 하나를 분석해보도록 하자!

먼저 위에서 소개한 모든 도구들을 준비하기 바란다. 더불어 자신만의 악성코드 분석에 필요한 가상 머신을 운영체제 별로 하나씩 구비해놓으면 금상첨화이다. 실제로 필자는 모든 도구들이 입맛에 맞게 설치된 가상 머신을 수십 개 준비해놓고 쓰고 있다. 설치하는 데는 시간이 오래 걸리지만 한번 설치해놓으면 나중에 분석할 때 매우 편하므로 하루 날 잡고 한번 준비해 보는 것도 좋겠다.

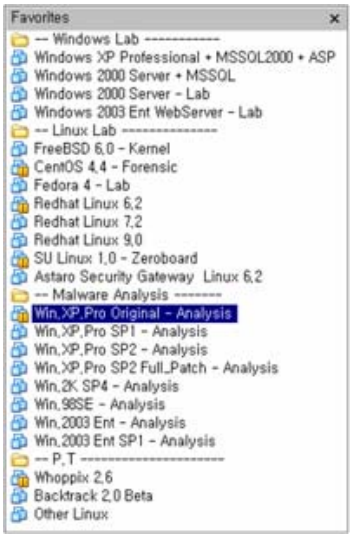

그림12. 실제 한 실무자의 가상 머신 이미지들[33]

그럼 본격적으로 분석을 시작하자. 먼저, 목표물이 있어야 한다. 필자는 wsass1.exe 라는 악성

[32] 개인적인 생각으로는 역시 가장 중요한 것은 악성코드의 분석결과를 많은 사람들과 공유하고, 탐지하고 치료하는 방법을 개발하는 것이라 생각한다. 그게 바로 우리가 악성코드를 분석해야 하는 근본적인 이유이기 때문이다.

[33] CertLab, 『Malware Analysis』, 2007

코드 샘플을 목표물로 설정했다. 또한 대상 운영체제는 취약점 패치가 안된 MS Windows XP ServicePack2를 선택했다.

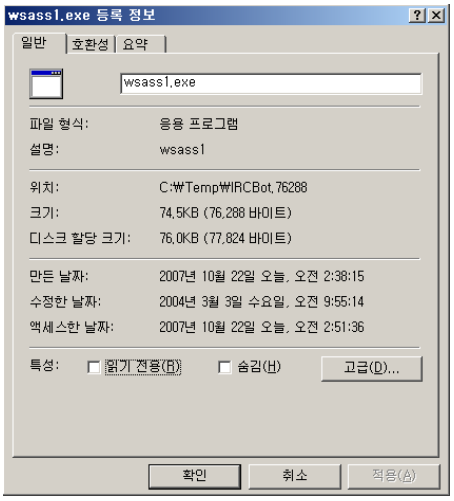

그림13. 목표로 설정한 악성코드 wsass1.exe

다음으로 ClamAV나 Virus Total[34]등의 악성코드 탐지 도구로 어떤 악성코드인지 사전 정보를 구해보자.

File **wsass1.exe** received on **10.21.2007 19:59:55 (CET)**
Current status: **scanning**

Your file is being scanned by VirusTotal in this moment,
results will be shown as they're generated.

Antivirus	Version	Last Update	Result
AhnLab-V3	2007.10.20.0	2007.10.19	Win32/IRCBot.worm.76288.B
AntiVir	7.6.0.27	2007.10.21	Worm/IrcBot.76288.B
Authentium	4.93.8	2007.10.20	W32/Sdbot.UX
Avast	4.7.1051.0	2007.10.21	Win32:IRCbot-P
AVG	7.5.0.488	2007.10.21	IRC/BackDoor.SdBot.FYO
BitDefender	7.2	2007.10.21	Generic.Sdbot.84D1C341
CAT-QuickHeal	9.00	2007.10.20	Backdoor.IRCBot.Gen
ClamAV	0.91.2	2007.10.21	Worm.IRC.Randbot
DrWeb	4.44.0.09170	2007.10.21	Win32.IRC.Bot.based

Additional information

File size: 76288 bytes
MD5: a9d62ac95dfc1b9aeead8f4ec181a1d4
SHA1: 731f199998b8f0bf8812387d02efeb33d5be119c

그림14. Virus Total로 얻어낸 악성코드의 정보

그림14.를 보면 많은 안티바이러스 프로그램들이 대상 파일 wsass1.exe에 대해 IRCBot기능이있는 웜이라 판단하고 있다. 우리에게 친숙한 안철수 연구소의 판단을 믿고, 안랩[35] 홈페이지의 바이러스DB에서 해당 웜을 검색해보자.

[34] http://www.virustotal.com

[35] http://www.ahnlab.co.kr

Win32/IRCBot.worm.76288.H

상세정보	처료법	예방법	참고자료

• 최초 입력시간 : 2006. 10. 18 09:52 (GMT+9) • 최종 수정시간 : 2006. 10. 18 09:52 (GMT+9)

위험도	시스템 위험도	네트워크 위험도	확산 위험도	현재 확산도 그래프
	높음	높음	높음	보통

다른이름	Backdoor.Win32.SdBot.qv, Generic.Sdbot.3169B48D, Win32.HLLW.MyBot, TR/Packed.CryptExe, W32/Sdbot.worm.gen.ai virus		
생성 파일명	lsscs.exe		
대표적 증상	시스템 관련, 네트워크 관련, 보안상 위험, 사용자 정보 유출		
활동 플랫폼	윈도우	감염/설치 경로	네트워크, 보안취약점
종류	웜, 백도어	형태	실행파일
들어오는 포트		나가는 포트	
제작국	불분명	특정활동일	특정일 활동 없음
최초 발견일	2006-10-17 (현지시각 기준)	국내 발견일	2006-10-17

그림15. 안철수 연구소에서 제공하는 해당 웜의 정보

밑으로 계속 읽어보면, 해당 웜의 세부 동작에 관해서도 설명하고 있는데, 이미 이 웜에 대한 분석이 이미 끝났음을 말해준다.

증상 및 요약 ▲ TOP

Win32/IRCBot.worm.76288.H 는 Win32/IRCBot.worm의 변형 중 하나이다. 해당 웜은 잘 알려진 윈도우 취약점과 윈도우 사용자 계정의 취약한 암호를 통해 전파된다. 해당 웜이 실행되면 윈도우 시스템 폴더에 lsscs.exe (76,288 바이트) 를 생성하고 자신을 서비스에 등록하여 윈도우 시작 시 자동으로 실행되게 한다. 그리고 특정 IRC 서버의 채널에 접속하여 오퍼(방장)가 내리는 명령에 따라 다양한 악의적인 기능을 수행하게 된다.

그림16. 안철수 연구소에서 제공하는 해당 웜의 정보

하지만, 이 정보가 반드시 내가 가진 웜이라고 확신할 수는 없다. 다만 대충 이렇다고만 알아두고, 다음 분석절차에 참고 정도만 하도록 하자.

다음 절차로 웜을 직접 실행시켜보도록 하겠다. 그전에 반드시 각종 모니터링 도구가 웜의 행동을 감지 할 수 있게 준비해놓자. 또한 가상 머신의 이미지에 대한 스냅샷은 반드시 찍어두도록 하자.

svchost.ex...	TCP	winxp-sp0:ep...	winxp-sp0:0	LISTENING
System:4	TCP	winxp-sp0:micr...	winxp-sp0:0	LISTENING
System:4	TCP	winxp-sp0:net...	winxp-sp0:0	LISTENING
System:4	UDP	winxp-sp0:micr...	*:*	
System:4	UDP	winxp-sp0:net...	*:*	
System:4	UDP	winxp-sp0:net...	*:*	
wsass.exe:...	TCP	winxp-sp0:auth	winxp-sp0:0	LISTENING
wsass.exe:...	TCP	winxp-sp0:1104	webrouter0.telia.nic.nu:9070	SYN_SENT

그림17. 네트워크 상태 변화

그림17. 네트워크 상태 변화에 대한 감지 결과를 보면 wsass.exe라는 파일이 특정 주소의 9070 TCP 포트로 접속을 시도하고 있음을 알 수 있다. 결과를 통해 이 서버가 바로 해커가 만들어놓은 IRC 서버라는 것을 추측해 볼 수 있다.

Name	Critical	Warning	Info	Ignored
Files	0	0	3	0
Groups	0	0	0	0
Registry	0	3	9	7
Rights	0	0	0	0
Scheduler	0	0	0	0
Services	0	0	0	0
Shares	2	0	0	0
System	0	0	0	0
Users	0	0	0	0
Volumes	0	0	0	0

그림18. 악성코드 실행 전/후의 시스템 상태의 변화 요약

그림18.은 전체적인 시스템 상태 변화에 대한 감지 결과로, 이 악성코드가 파일과 레지스트리, 그리고 공유폴더에 대한 상태를 조작하고 있음을 말해준다.

Description	Name	New Value	Old Value	Severity
New File	C:₩a.bat			3

Description	Name	New Value	Old Value	Severity
New File	C:₩WINDOWS₩System32₩msnsrv.exe			3
New File	C:₩WINDOWS₩System32₩wsass.exe			3

그림19. 파일 읽기/쓰기 감지 결과

그림19.는 파일 읽기/쓰기 상태 변화에 대한 감지 결과이다.

C:₩a.bat

C:₩WINDOWS₩System32₩msnsrv.exe

C:₩WINDOWS₩System32₩wsass.exe

위 3개의 파일이 새로 생성되었음을 알 수 있다.

D..	Name	New Value
	HKLM₩SYSTEM₩CurrentControlSet₩Services₩SharedAccess₩Par...	2
	HKLM₩SYSTEM₩CurrentControlSet₩Services₩SharedAccess₩Par...	2007-10-22 오전 3:56:23
	HKLM₩SYSTEM₩CurrentControlSet₩Services₩SharedAccess₩Par...	C:₩WINDOWS₩system32₩wsass.exe:...
	HKLM₩SYSTEM₩CurrentControlSet₩Services₩SharedAccess₩Epo...	2007-10-22 오전 3:56:23
	HKLM₩SYSTEM₩CurrentControlSet₩Services₩SharedAccess₩Epo...	6
	HKLM₩SYSTEM₩ControlSet001₩Services₩SharedAccess₩Parame...	2
	HKLM₩SYSTEM₩ControlSet001₩Services₩SharedAccess₩Parame...	2007-10-22 오전 3:56:23
	HKLM₩SYSTEM₩ControlSet001₩Services₩SharedAccess₩Parame...	C:₩WINDOWS₩system32₩wsass.exe:...
	HKLM₩SYSTEM₩ControlSet001₩Services₩SharedAccess₩Epoch	2007-10-22 오전 3:56:23
	HKLM₩SYSTEM₩ControlSet001₩Services₩SharedAccess₩Epoch...	6
	HKLM₩SOFTWARE₩Microsoft₩Windows₩CurrentVersion	48
	HKLM₩SOFTWARE₩Microsoft₩Windows₩CurrentVersion	2007-10-22 오전 3:56:21
	HKLM₩SOFTWARE₩Microsoft₩Windows₩CurrentVersion₩RunServi...	
	HKLM₩SOFTWARE₩Microsoft₩Windows₩CurrentVersion₩RunServi...	wsass.exe
	HKLM₩SOFTWARE₩Microsoft₩Windows₩CurrentVersion₩Run	8
	HKLM₩SOFTWARE₩Microsoft₩Windows₩CurrentVersion₩Run	2007-10-22 오전 3:56:21
	HKLM₩SOFTWARE₩Microsoft₩Windows₩CurrentVersion₩Run₩Wi...	wsass.exe
	HKLM₩SOFTWARE₩Microsoft₩Cryptography₩RNG	2007-10-22 오전 3:56:22
	HKLM₩SOFTWARE₩Microsoft₩Cryptography₩RNG₩Seed	46 4f 0c fb 38 64 e4 6c 24 a2 80 08 c6 a9 ...

그림20. 레지스트리 읽기/쓰기 감지 결과

그림20.은 레지스트리 읽기/쓰기에 대한 변화로, 특정 레지스트리를 편집한다. 그 중 중요한 값을 정리해보면 다음과 같다.

HKLM₩SOFTWARE₩Microsoft₩Windows₩CurrentVersion₩RunServices에
이름 Windows WKS, 값 wsass.exe의 레지스트리 생성
HKLM₩SOFTWARE₩Microsoft₩Windows₩CurrentVersion₩Run에
이름 Windows WKS, 값 wsass.exe의 레지스트리 생성

위의 레지스트리가 등록되면 시스템이 시작될 때마다 새로 생성된 파일이(악성코드 사본으로 추측) 실행된다. 실제로 변화된 레지스트리 값을 확인해 보자.

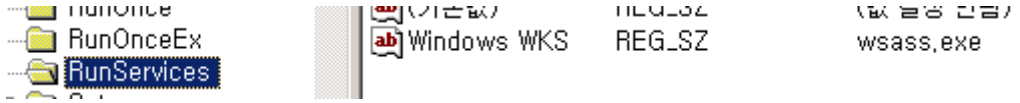

그림21. 변화된 레지스트리 값1

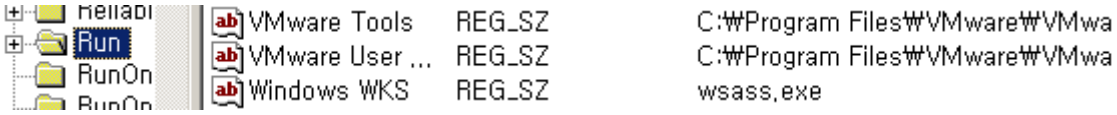

그림22. 변화된 레지스트리 값2

이 정도면 행동기반 분석은 OK다. 이제 이 악성코드가 대충 어떤 방식으로 동작하는지 알 수 있을 것이다. 그럼 이 정보를 토대로 코드기반 분석 절차를 밟도록 하자.

먼저 PEiD로 실행압축 여부를 확인한다.

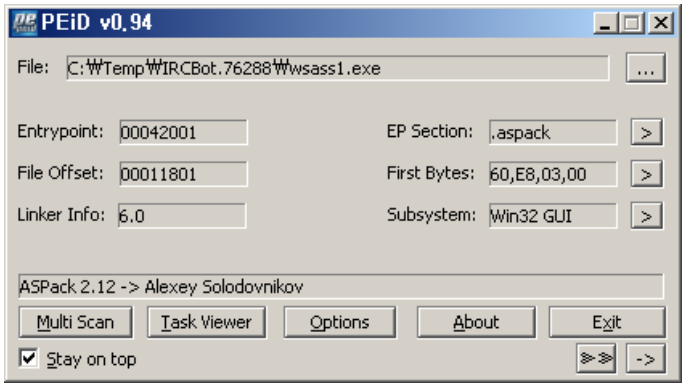

그림23. PEiD로 탐지한 실행압축 정보

ASPack 2.12 -> Alexey Solodovnikov라는 팩커로 실행 압축 되어있다. 실행 압축이 되어있으므로, 제대로 분석하기 위해선 먼저 언팩킹을 할 필요가 있다. 언팩킹에 관해서 이 문서에 일일이 설명하는 것은 양도 많거니와 문서의 주제에서도 벗어나므로, 간단하게 요약하고자 한다.

더 자세한 언팩킹 정보는 각종 리버스 엔지니어링 사이트를 참조하기 바란다.

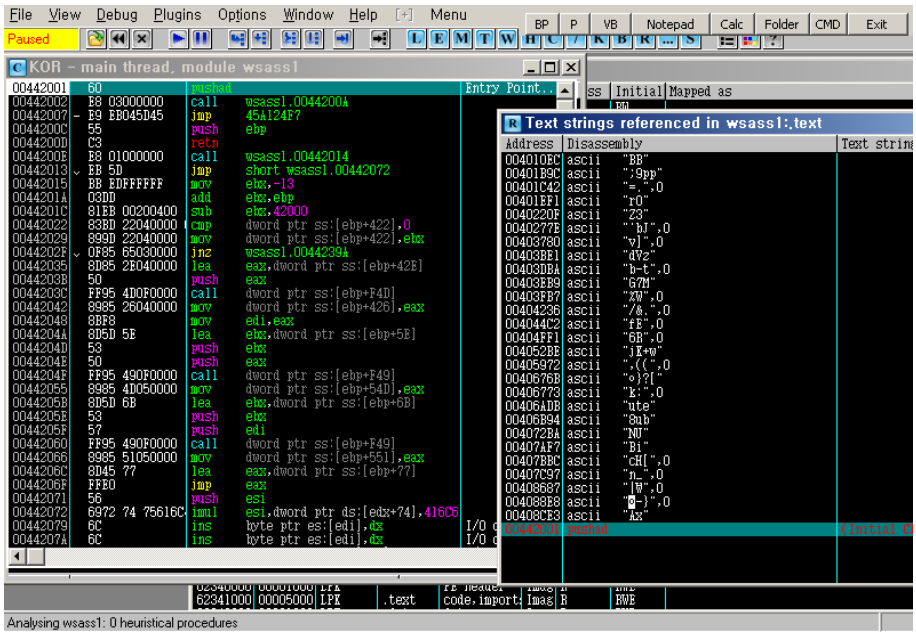

그림24. OllyDbg 디버거로 분석한 화면

그림24.를 보면 디버거가 인식할 수 있는 아스키 문자열이 하나도 없다. 바로 이 때문에 언팩킹이 필요한 것이다.

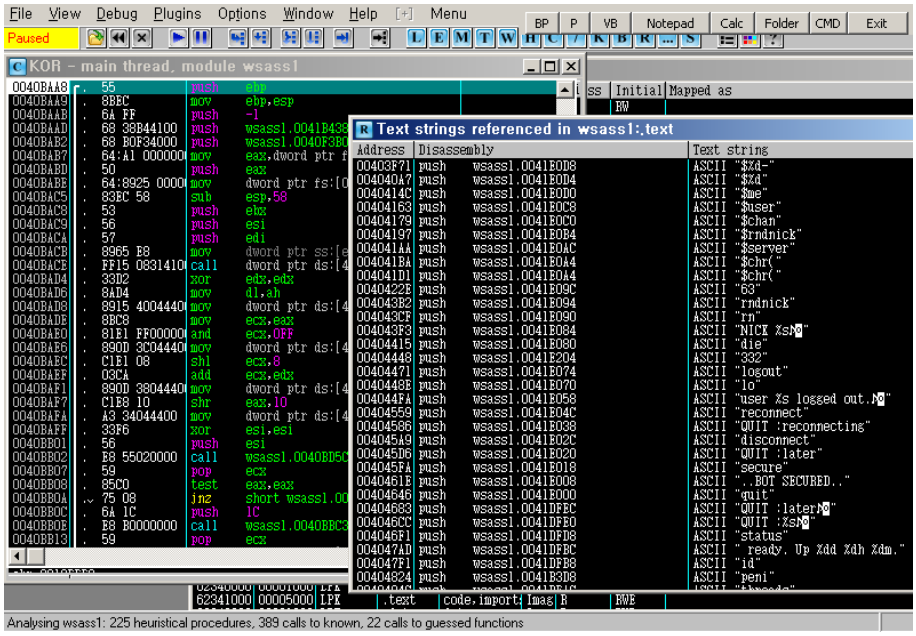

그림25. OEP[36]를 찾은 화면

[36] Original Entry Point, 실행 압축되기 전의 원래 프로그램의 시작 포인트

그림25.는 OEP(0x40BAA8)를 찾은 화면이다. 오른쪽 창을 보면 그림24.에는 나오지 않았던 아스키 문자열들이 보인다. 바로 이 웜이 사용하는 문자열들이다. 자세히 들여다보면 상당수의 IRC 명령어가 포함되어 있음을 알 수 있다. 언팩킹된 파일을 Strings로 문자열을 뽑아보면 더 자세하게 알 수 있다.

OEP를 찾은 뒤에는 덤프[37]를 받고 IAT[38]를 복구 시켜줘야 한다.

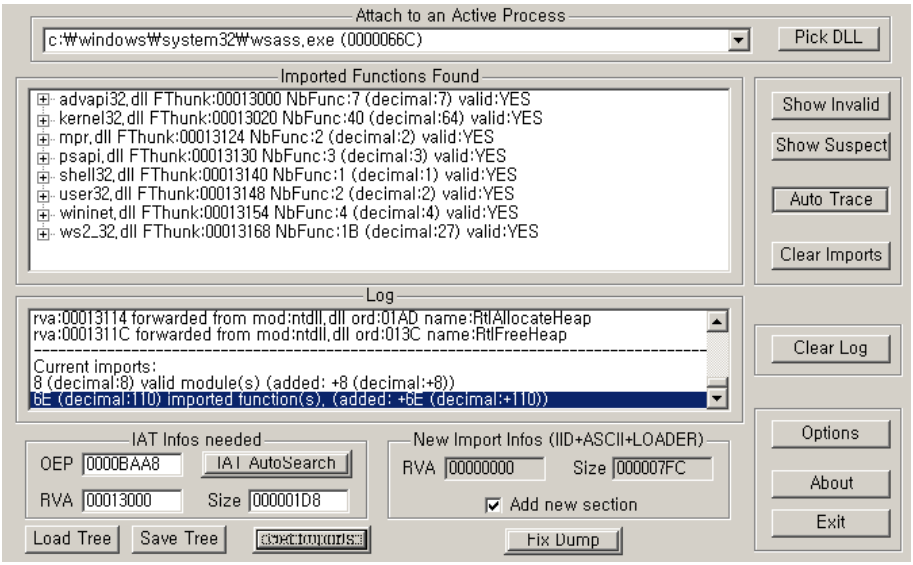

그림26. IAT 복구 화면

IAT까지 복구 해주면 이제 웜은 완전히 실행 압축이 해제된 것이다. 진짜 실행 압축이 해제되었는지 PEiD로 확인해 보자.

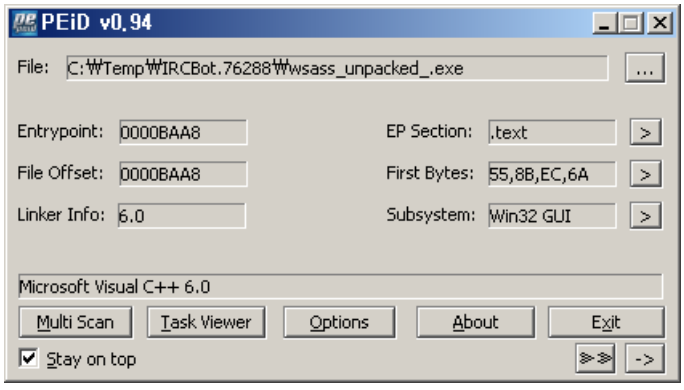

그림27. 언팩킹 후의 웜 실행 파일

결과를 보면, 이 웜은 Visual C++ 6.0으로 제작되었다고 나온다. 그럼 다음으로 넘어가서 언팩

[37] 메모리의 내용을 하드디스크의 파일로 저장하는 것.

[38] Import Address Table, 자세한 내용은 PE헤더에 대한 서적을 참조하기 바란다.

킹된 실행파일에 대해 Strings로 분석해보자.

그림28. 웜이 사용하는 IRC 명령어 목록

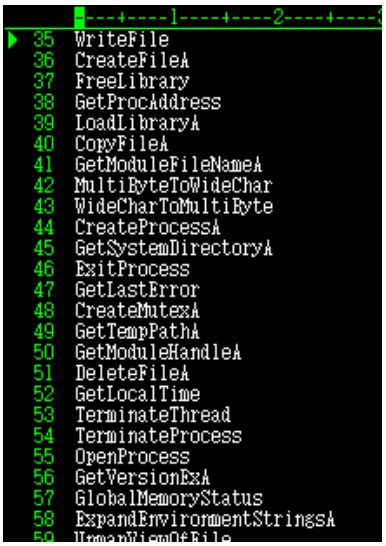

그림29. 웜이 사용하는 API 함수 목록

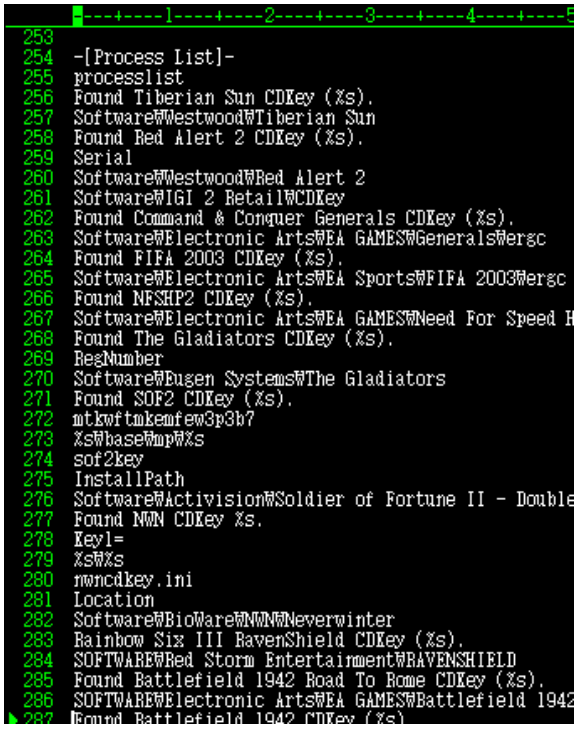

그림30. 각종 게임의 시디키를 빼내는 코드

그림30.을 보면 재미있는 사실을 알 수 있다. 바로 특정 게임의 여부를 검사해서 그 게임의 시

디키를 빼내는 기능이다. (아마도 이 웜을 만든 해커는 이 게임들을 좋아하는 것 같다)

IRC 명령어들로 추측되는 문자열들을 분석해보면, 해커는 피해자에게 원격으로 명령을 내려 시스템 정보를 얻거나 파일을 지우거나 다른 PC를 공격하도록 할 수 있음을 짐작할 수 있다. 그럼 실제로 그 코드를 눈으로 확인해 보자.

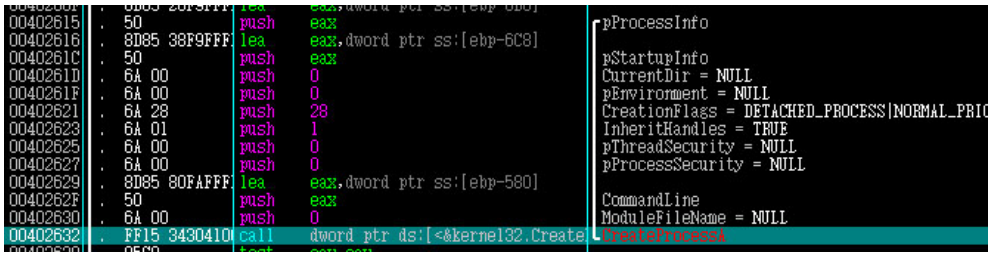

그림31. 프로세스 생성 코드

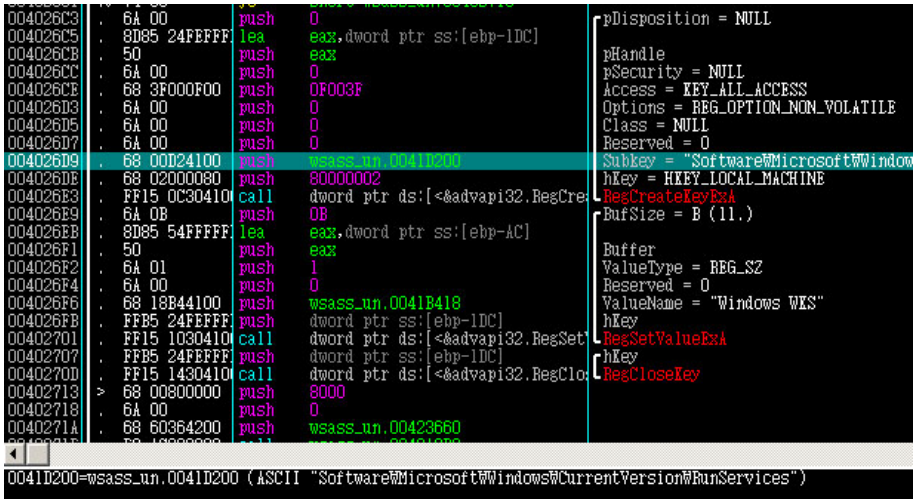

그림32. 시작 프로그램 레지스트리값 생성 코드1

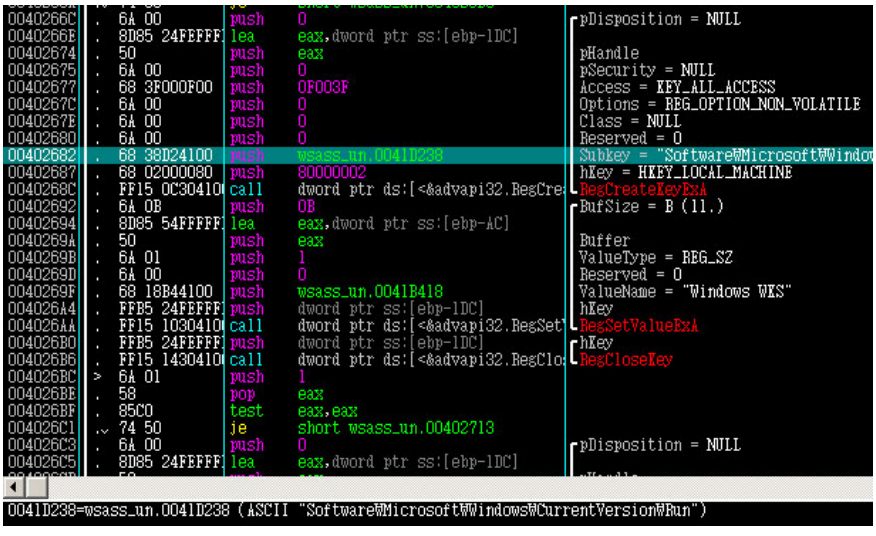

그림33. 시작 프로그램 레지스트리값 생성 코드2

사실 여기까지만 분석해도 대부분의 정보를 다 알았기에 중단해도 되지만 실제로 웜이 어떤 일을 하는지 테스트 해보자(!) 방법은 간단하다, 디버거로 실시간 추적을 하다가 봇이 해커가 만들어놓은 IRC 서버로 접속을 시도하기 직전 우리가 만들어놓은 IRC 서버 주소로 값을 조작하면 된다. 그리고 실제로 웜이 잘 작동하는지, 어떤 기능들이 있는지 해커의 입장에서 웜에게 명령을 내려보자.

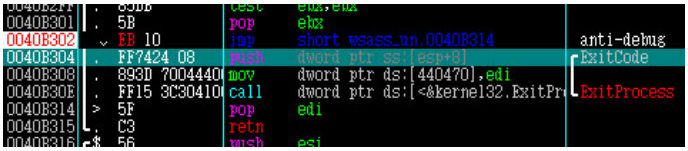

그림34. 안티-디버깅 루틴

하지만 가장 처음에 특정 조건에 따라 프로세스를 종료하는 코드가 들어있는데, 항상 프로세스가 종료되었다. 그림34.를 보면 알 수 있듯이 이부분을 jmp 명령으로 패치해서 우회하였다.

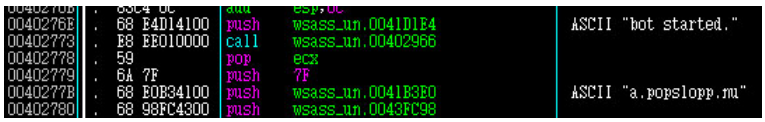

그림36. 해커가 만들어놓은 IRC서버 주소

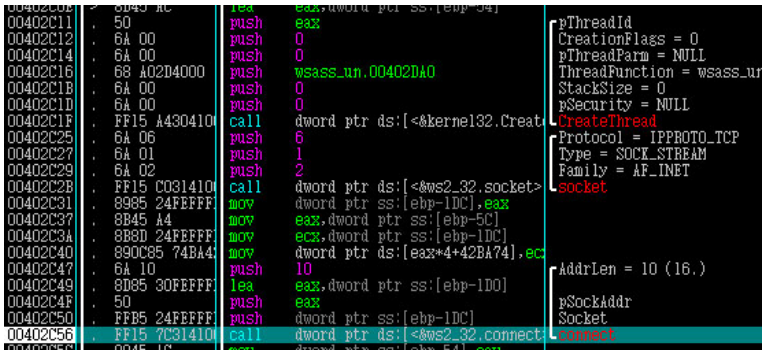

그림35. IRC서버에 접속 하는 코드

그림36.의 코드를 통해 봇은 미리 해커가 만들어놓은 특정 IRC서버로 접속을 시도하고 있다. 이 부분을 우리가 원하는 주소로 수정하면, 봇을 우리가 원하는 서버로 접속시킬 수 있다. 또한 봇을 일반적인 IRC서버로 접속을 유도하려면 스택에 포인터로 올라와있는 구조체의 포트번호도 수정해야 한다.

그림36. 특정 채널에 접속하는 코드

그림36.의 코드를 통해 봇은 해커가 미리 만들어놓은 채널에 자동으로 들어가게 된다.

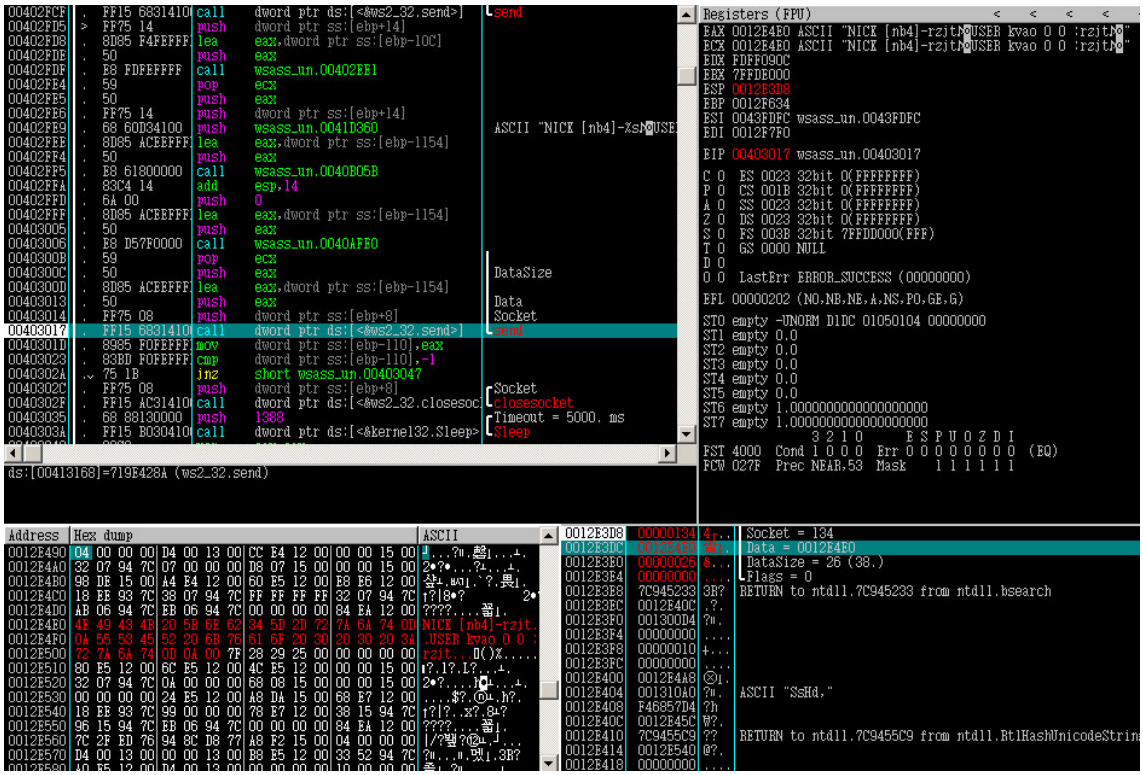

그림37. IRC봇이 서버에 접속한 후 닉네임을 등록하는 코드

여기까지의 모든 값들을 내가 원하는 값들로 수정하게 되면 아래 그림과 같이 봇은 내가 만들어놓은 IRC채널로 접속하게 된다.

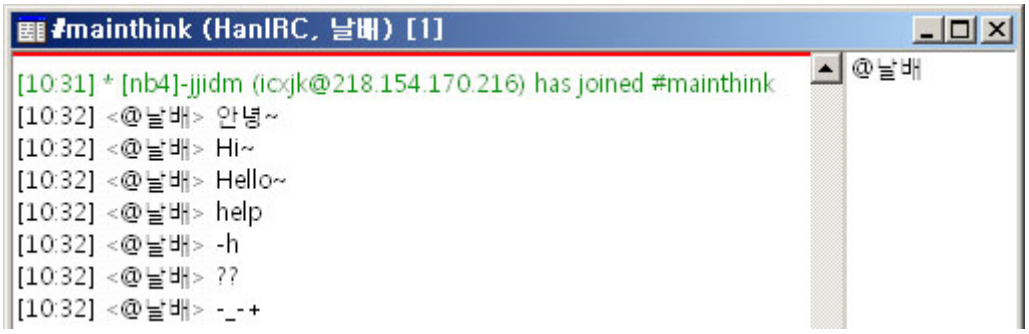

그림38. 봇을 내가 원하는 IRC채널로 접속시킨 화면

채널에 들어온 봇에게 말을 걸어봤지만 아무 대꾸도 하지 않았다. 이는 봇을 만든 해커가 자신의 말만 듣게 하기 위해 만들어놓은 인증 코드 때문이다.

이 문서에서 분석하고 있는 봇은 암호인증과 호스트인증의 2단계 인증 절차를 가지고 있었지만, 약간의 분석기술만 있으면 얼마든지 크랙 가능한 쉬운 코드로 되어있었다. (문서에 인증 받

는 부분의 코드도 넣으려 했지만, 그렇게 되면 악용될 소지가 있으므로 생략했다. 또한 인증 코드 역시 모자이크 처리했음을 양해 바란다)

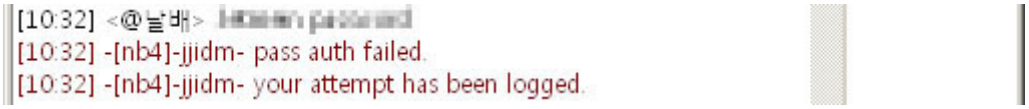

그림39. 암호 인증 기능

그림40. 암호 인증 기능 크랙

그림41. 호스트 인증 기능 크랙

호스트인증까지 크랙하면 봇으로부터 password accepted. 라는 응답을 받게 되며 이후부터는 봇에게 다음과 같이 내가 원하는 명령을 내릴 수 있게 된다.

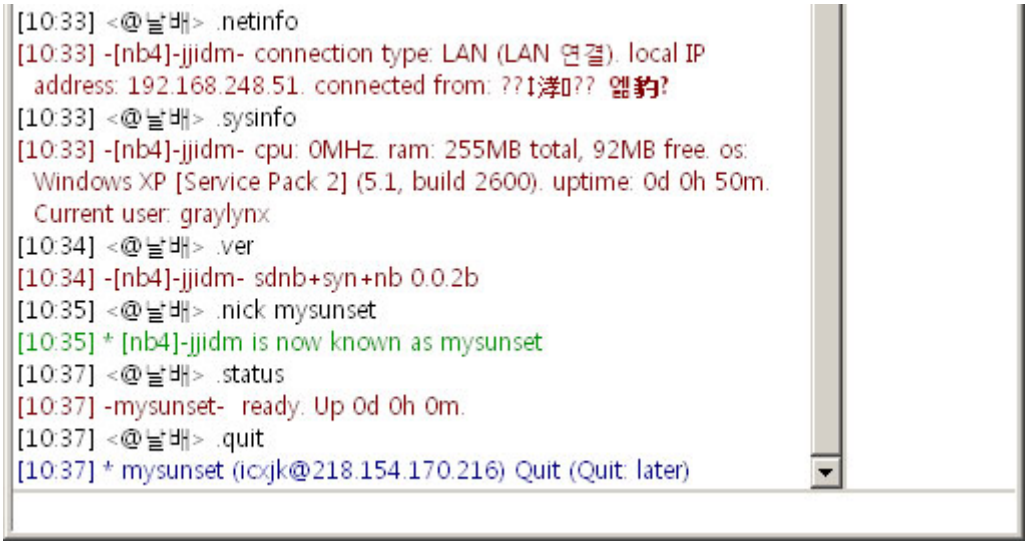

그림42. 두가지 인증을 모두 크랙하고, 봇에게 명령을 내린 화면

5. 대응

마지막으로, 혹시나 이미 자신의 시스템이 이 웜에 감염 되었다면, 다음과 같은 방법으로 치료가 가능하다.

가. C:₩WINDOWS₩System32₩wsass.exe 삭제

나. HKLM₩SOFTWARE₩Microsoft₩Windows₩CurrentVersion₩RunServices₩Windows WKS

HKLM₩SOFTWARE₩Microsoft₩Windows₩CurrentVersion₩Run₩Windows WKS 레지스트리값 삭제

다. 최신 윈도우즈 취약점 패치 수행

라. 최신 엔진의 안티 바이러스 프로그램 실행

6. 마치며..

이제까지 악성코드를 분석하기 위한 여러 가지 방법들과 필요한 도구들을 알아보았다. 하지만 필자가 이런 방법들과 도구들을 소개했다고 해서 그 내용이 이 글을 읽는 독자에게 완전히 흡수 될 수는 없다. 악성코드를 잘 분석하기 위해서, 가장 좋은 방법은 역시 무조건 분석해보는 것이다. 악성코드를 수집하는 허니팟 프로젝트 또는 악성코드 연구 홈페이지들을 찾아 다니며, 끊임없이 수집하고 분석해보는 노력이 필요하다.

하지만 그 중 가장 중요하고 명심해야 할 점은 악성코드 수집이나 분석은 100% 공부나 연구를 목적으로 수행되어야 한다는 것이다. 실제로 악성코드를 많이 분석 하다 보면, 악성코드의 원리와 코드가 속속들이 다 보이게 된다. 더 나아가 이런 부분은 이렇게 만들었다면, 더 치명적이었을 건데.. 라는 까지 생각들 때가 있다. 인터넷 윤리의식이 똑바로 자라지 못한 사람이 악성코드 분석 기술을 습득하게 된다면, 그건 국가적 손실일 뿐만 아니라 치명적인 잠재적 위협요소로써 매우 위험한 일이다.

이는 악성코드 분석가들이 가장 조심하고 신중해야 할 부분이기도 하다. 어떤 사이트를 해킹하다 경찰에 잡혀서 옥살이를 하고 나왔더니, 수많은 기업에서 서로 데려가려고 하더라 라는 말은 이제는 근거도 없는 헛소리다. 오히려 해킹에 대한 단속이 심하고 처벌이 강한 우리나라에서는 한번 불법자로서 낙인이 찍히면 두 번 다시 이 계열로는 직장을 구하지 못할 만큼 해커에 대한 인식이 안 좋은 편이다.

경제적 여건이나 기타 다른 이유로 이런 뛰어난 능력을 까딱 잘못 사용했다가는 평생 낙오자로 살아야 할 수도 있다. 그만큼 위험한 기술이기 때문이다. 이 분야에 종사하는 사람들은 자부심과 책임감을 갖고 항상 올바른 마음가짐을 가지려고 스스로 노력해야 한다. 필자는 적어도 이 문서를 읽는 모든 사람들이 그런 마음가짐과 의식을 가지고 있다고 믿고 싶다.

부족한 글 끝까지 읽어주셔서 감사합니다. ☺ Happy Hacking~!