알려지지 않은 스파이웨어 탐지기술

최근 인터넷에 접속한 10대의 PC중 9대가 스파이웨어에 감염된다는 보고가 있다. 그러나 대부분의 사용자는 자신의 PC가 스파이웨어에 감염되었는지 잘 모르는 경우가 많다. 인터넷 사용자에게 큰 위협이 되는 기존의 스파이웨어 탐지 기술을 소개하고 새로운 탐지 기법인 HoneyID를 소개한다.

글 · 한제헌 고려대 컴퓨터보안연구실 연구원, 이희조 고려대 교수

연재순서

1. 행위기반 봇넷 탐지기술
2. **알려지지 않은 스파이웨어 탐지기술**
3. VoIP공격 탐지기술
4. 알려지지 않은 신종 인터넷 웜 탐지기술
5. 악성코드 사전 탐지기술
6. IP 스푸핑 탐지기술

스파이웨어는 PC에 사용자 모르게 설치되어 악성 행위를 하는 소프트웨어를 말한다. 스파이웨어는 사용자에게 광고를 노출하고 사용자의 행위나 정보를 관찰하거나 사용자의 PC를 조작하는 등 여러 가지 목적으로 제작된다.

스파이웨어는 인터넷 사용자에게 큰 위협 중 한 가지가 됐다. Webroot Software, Inc.의 2006년 'Spyware Info and Facts that All Internet Users Must Know' 보고서에 따르면 인터넷에 접속한 10대의 PC중 9대가 스파이웨어에 감염된다고 한다. 그러나 대부분의 사용자는 자신의 PC가 스파이웨어에 감염되었는지 잘 모르는 경우가 많기 때문에 스파이웨어에 의한 피해는 점점 더 커지고 있다.

특히 사용자의 행위나 정보를 관찰하는 유형의 스파이웨어는 사용자의 금융정보와 같은 중요한 정보를 유출시킬 수 있기 때문에 이런 유형의 스파이웨어는 금전적 피해를 발생시킬 수도 있다. 스파이웨어에 의해 크고 작은 사고가 발생하고 있는 가운데, 2005년에 엔씨소프트의 리니지 게임 계정이 스파이웨어에 의해 대량 유출되었던 사례가 있으며 2007년에는 사용자의 공인인증서를 빼돌리는 악성 코드로 인한 피해가 발생된 사례가 있다.

이러한 사용자의 요구에 따라 스파이웨어 탐지, 치료 솔루션이 많이 배포되고 있으나 대부분의 상용 프로그램은 시그니처 탐지 기법을 사용하기에 신종 스파이웨어에 대한 시그니처가 업데이트되기 전까지는 해당 스파이웨어를 탐지할 수 없다는 단점이 있다. 이 글에서 기존의 행위기반 탐지기법을 소개하고 HoneyID를 이용한 탐지기법을 설명한다.

기존 탐지기술의 문제점

스파이웨어 탐지기술은 크게 시그니처 기반 탐지, 프로그래밍 언어기반 탐지, 행위기반 탐지로 분류될 수 있다. 시그니처 기반 탐지는 현재 대중적으로 사용되고 있는 Lavasoft의 AdAware나 AhnLab의 SpyZero와 같은 상용 안티 스파이웨어 솔루션에서 주로 채택하고 있는 기법이다.

이 기법은 스파이웨어의 파일, 레지스트리, 프로세스에 대한 정보를 DB에 저장하고 PC에 해당 내용과 일치하는 프로그램이 있다면 삭제하는 방식을 사용한다. 오탐률이 적다는 장점으로 상용 프로그램에서 많이 사용하지만 새로 발견된 스파이웨어에 대한 시그니처가 사용자에게 업데이트되기 전에 탐지될 수 없다는 단점이 있다.

프로그래밍 언어기반 탐지기법은 스파이웨어의 실행 파일을 탐지하기 위해 바이러스 탐지기법과 유사하게 실행 가능한 코드에 대해 정적인 분석을 하는 기법이다. Berkeley 대학의 M. Christodorescu는 2003년 Usenix Security Semposium에서 실행 파일 내부의 특정한 패턴을 이용해 악성 실행 파일을 탐지하는 방법을 제안했다. 하지만 이 방법 또한 파일 내부의 특정한 패턴을 이용하는 것이기 때문에 다형성 기법을 통해 우회될 수 있다는 단점을 가지고 있다. 행위기반 탐지기법은 알려지지 않은 스파이웨어를 탐지할 수 있기 때문에 다른 기법에 비해 활발한 연구를 보이고 있다.

Y. M. Wang이 2004년에 Usenix Large Installation System Administration Conference에서 ASEP(Auto-Start Extensibility Points)를 감시하여 스파이웨어를 탐지하는 방법을 제안했다. 이 방식은 MS 윈도우의 시작 프로그램과 같은 운영체제가 시작될 때 자동으로 프로그램을 실행시키는 위치에 생성되는 프로그램을 감시하고 사용자에게 알려준다. 이는 대부분의 스파이웨어가 스스로를 항상 동작시키기 위해 프로그램 시작 포인트에 자신을 설치하는 행위에 근거하여 탐지하는 방식이다. 이 방식은 시작 프로그램에 등록하는 스파이웨어를 잡는 것에는 탁월하지만 그렇지 않은 스파이웨어는 탐지할 수 없다. E. Kirda는 2006년 Usenix Security Symposium에서 행위기반의 스파이웨어 탐지 시스템을 제안했다. 이는 MS Explorer에 애드온으로 설치되는 BHO(Browser Helper Object)에서 악성의 위험이 있는 API를 호출하는 것을 탐지하는 방식이다. 예를 들어 BHO에서 파일 생성 API를 호출하거나 다른 서버에 접속하는 API를 호출하는 것을 잡는 것이다. 이 방식은 일반적인 스파이웨어를 잡지 못하고 BHO 방식의 스파이웨어만 탐지할 수 있다.

침해형 스파이웨어와 대화형 스파이웨어

스파이웨어는 크게 침해형 스파이웨어와 대화형 스파이웨어로 분류될 수 있다. 침해형 스파이웨어는 사용자의 PC의 성능을 저하시키거나 마비시키며 이차적 공격을 위해 다른 스파이웨어를 다운로드 하는 등의 행위를 하는 스파이웨어를 지칭한다. 대화형 스파이웨어는 사용자의 정보를 탈취하거나 조작하는 등의 행위를 하는 스파이웨어를 말한다. 침해형 스파이웨어로는 Adware, Dropper, Bot 등이 있으며, 대화형 스파이웨어로는 Keylogger, Hijacker 등이 있다.

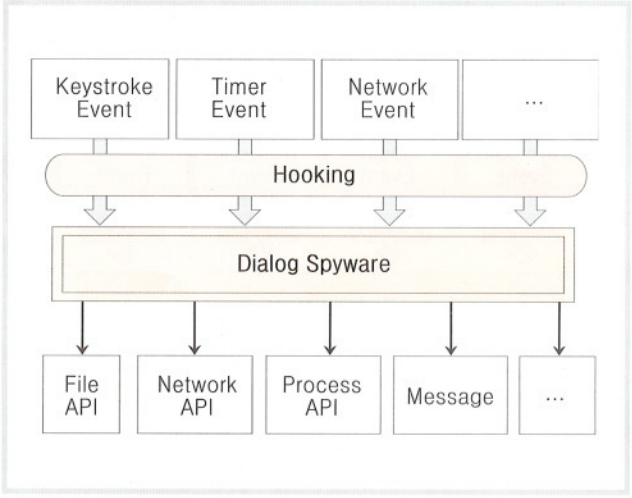

그림 1. 대화형 스파이웨어 동작방식

대화형 스파이웨어가 침해형 스파이웨어와 분류될 수 있는 가장 큰 차이점은 사용자의 정보를 이용한다는 것이다. 대화형 스파이웨어는 사용자의 정보를 이용하기 때문에 금융관련 정보나, 계정정보와 같은 민감한 정보를 유출시킬 수 있다. 최근에는 침해형 스파이웨어와 대화형 스파이웨어가 결합된 형태의 스파이웨어가 등장하고 있다.

예를 들면 Bot이 Keylogger의 기능을 탐재하여 배포되거나 Dropper가 Hijacker를 다운로드 한다. 대화형 스파이웨어는 사용자의 행동을 관찰하다가 원하는 정보가 생성될 때 이를 이용하여 악의적인 목적을 하는 특징을 가지고 있다.

HoneyID 메커니즘 개발

고려대학교 컴퓨터보안연구실은 대화형 스파이웨어를 탐지할 수 있는 새로운 방식의 행위기반 탐지 메카니즘인 HoneyID를 개발했다. HoneyID는 가짜 이벤트와 프로세스의 행동을 분석하는 Trap으로 구성된다. 기본 개념은 Honeypot과 비슷하게 공격을 유인하고 탐지하는 방식을 사용한다.

하지만 Honeypot이 네트워크상에서 일부러 취약하게 만든 시스템을 이용해 접근하는 공격행위를 탐지하는 반면에, HoneyID는 가짜 이벤트(Bogus Event)를 이용하여 로컬 PC에서 동작하고 있는 스파이웨어 프로세스가 악성 행위를 하도

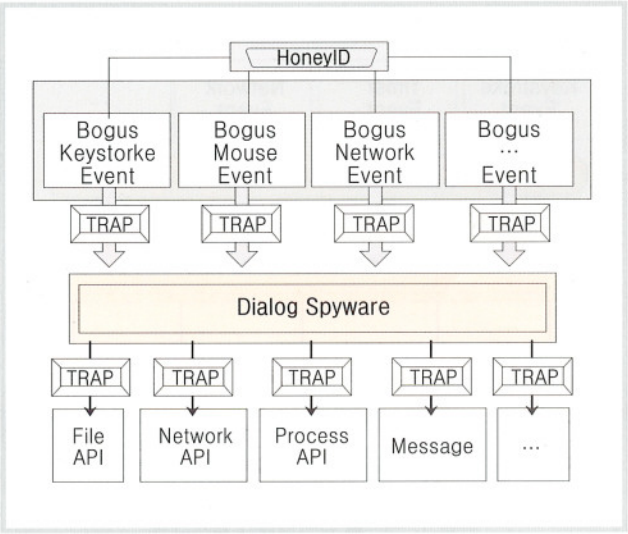

그림 2. HoneyID 동작 구조

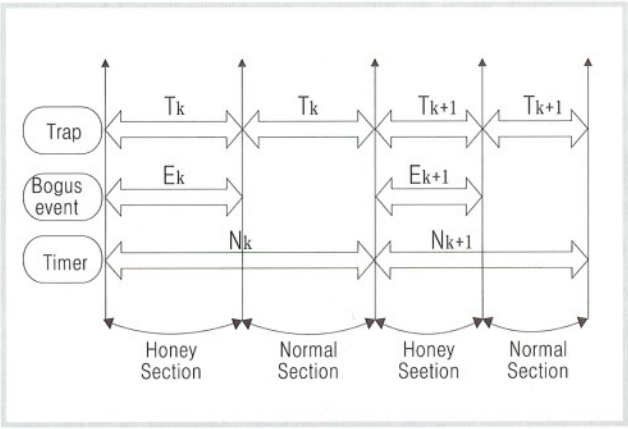

그림 3. 탐지 알고리즘

록 유인한다. 즉, Honeypot이 취약한 시스템을 이용해 공격 행위를 기다리는 수동적인 방식인 것과는 달리 HoneyID는 스파이웨어 프로세스가 동작하도록 만드는 능동적인 방식이다.

또한 탐지하는데 있어서도 Honeypot이 단순히 공격에 대한 정보를 수집하는 반면에 HoneyID는 Trap을 이용하여 일반 프로세스와 스파이웨어 프로세스를 구분하는 역할까지 한다.

가짜 이벤트를 통해 스파이웨어를 유인하는 것은, 앞서 설명한 것과 같이 대화형 스파이웨어는 사용자의 행위를 관찰하다가 원하는 정보가 생성될 때 이 정보를 이용하는데, 정보가 생성되는 것은 운영체제의 관점에서 이벤트가 발생되는 것과 같기 때문에 이 이벤트를 운영체제에서 가상으로 발생시키면 스파이웨어가 악성 행위를 수행 하도록 할 수 있다.

가짜 이벤트는 탐지하려는 스파이웨어의 종류에 따라 그 종류가 달라진다. 예를 들어 사용자의 키로깅 정보를 수집하는 키로거를 탐지하기 위해서 가상의 키 입력 이벤트를 발생시키고, 사용자의 로그인 정보를 수집하는 패스워드 스틸러를 탐지하기 위해서 가상의 로그인 이벤트를 발생시킨다.

Trap은 프로세스의 함수 호출, 이벤트 발생, 메시지 발생 등과 같은 정보를 관찰한다. 이를 통해 해당 프로세스가 발생시킨 가짜 이벤트를 이용하는지 이용하지 않는지 판별하는 것을 통해 스파이웨어 프로세스를 를 일반 프로세스와 구분할 수 있다.

HoneyID를 이용한 탐지 알고리즘

Trap은 일반 프로세스의 정상적인 동작도 수집하기 때문에 가짜 이벤트의 발생과 연계하여 정상 프로세스와 스파이웨어 프로세스를 구분해야 한다. 이것은 프로세스가 가짜 이벤트에 반응하여 동작한 것 인지 그렇지 않은지를 판별하여 대화형 스파이웨어를 판별해낼 수 있다. 만약 가짜 이벤트가 발생될 때마다 동작하고 발생되지 않을 때는 동작하지 않는 프로세스가 있다면 해당 프로세스는 가짜 이벤트에 반응하여 동작한다고 판단할 수 있으며 이는 대화형 스파이웨어로 분류될 수 있는 것이다.

이러한 방식을 컴퓨터로 판단할 수 있도록 하기 위해 탐지구간을 Honey Section으로 불리는 가짜 이벤트를 발생시키는 구간과 Normal Section으로 불리는 가짜 이벤트를 발생시키지 않는 구간으로 나눈다. 그 후 두 섹션에서 Trap에 수집된 정보를 비교하여 Honey Section에서 얼마나 많은 동작이 있었는가를 체크한다.

만약 Honey Section에서의 동작이 Normal Section에 비해 가짜 이벤트를 발생시킨 수와 비례하여 높다면 이는 대화형 스파이웨어로 분류될 수 있다. 일반 프로세스의 경우는 발생된 가짜 이벤트를 이용하지 않기 때문에 Section의 종류와 상관없이 동작하므로 이 과정을 여러 번 반복했을 때 평균적으로 두 섹션에서의 동작 개수가 같아진다는 것을 의미한다.

따라서 그림 3의 변수를 이용하여 수식으로 스파이웨어 프로세스와 일반 프로세스를 구분할 수 있다.

E는 초당 발생된 가짜 이벤트의 개수를 의미하고 N은 이벤트가 발생되고 Trap으로부터 정보를 수집한 초를 의미한다. T는 Honey Section에서 Trap으로부터 수집된 프로세스의 동작 개수, T는 Normal Section에서 Trap으로부터 수집된 프로세스의 동작 개수를 의미한다.

스파이웨어 프로세스의 경우 발생시킨 가짜 이벤트의 개수 $(Ek \cdot Nk)/2$ 와 Honey Section에서의 동작 개수 Tk가 같으며, Normal Section에서의 동작 Tk은 0이 되기 때문에 스파이웨어 프로세스이 C 값은 0이 된다. 만약 중간에 가짜 이벤트와는 상관없는 동작이 수행된다고 해도 이는 Section의 종류와는 상관없이 수행되기 때문에 위 과정을 z만큼 반복하면

결국 Tk-Tk수식에 의해서 0이 된다.

일반 프로세스의 경우는 Section의 종류와 상관없이 동작하기 때문에 Honey Section에서의 동작 개수 Tk와 Normal Section에서의 동작개수 Tk가 같다. 결국 가짜 이벤트 발생 개수만이 남게 되어 일반 프로세스의 C 값은 (Ek · · Nk)/2가 된다.

이러한 탐지를 z번 반복하는 사이에 중간에 Honey Section이나 Normal Section에서만 갑자기 가짜 이벤트와는 상관없는 동작이 일어나는 에러가 발생할 수 있다. 이러한 에러가 반복되면 결국 C에 영향을 미치기 때문에 이를 보정하기 위해 전체 탐지 시간 Nk로 C를 나눈다. 일반 프로세스의 C값은 발생된 이벤트 개수가 되기 때문에 최대값 MaxC는 그림 6의 수식과 같게 된다.

MaxC를 이용하여 스파이웨어와 일반 프로세스를 구분할 수 있는 임계치 값을 MaxC/2로 쉽게 계산할 수 있다. 즉, C값이 MaxC/2 보다 높다면 일반 프로세스로 구분하고 MaxC/2 보다 낮다면 스파이웨어로 구분할 수 있다.

스파이웨어 탐지 실험

일반적인 환경에서 실험하기 위해 사용하던 PC를 그대로 사용했으며 Google에서 Keylogger와 FTP Password Stealer 키워드를 이용해 각각 3개씩의 스파이웨어를 다운로드하여 설치했다.

가짜 이벤트로는 키로거와 FTP 패스워드 스틸러를 탐지하기 위해 각각 키보드 입력 이벤트와 FTP에 접속하여 로그인하는 패킷을 가상으로 만들어 발생 시켰다. Trap은 프로세스의 모든 정보를 수집하기에는 많은 자원을 소모하여 탐지 결과에 영향을 끼칠 수 있기 때문에, File 관련 함수, 사용자 메시지 발생 그리고 네트워크, 레지스트리, 파일 관련된 이벤트를 이용해 프로세스의 동작을 확인했다.

PMMODE는 일반 프로세스이며 ActualSpy는 키로거 그리고 AceSniffer는 FTP 패스워드 스틸러이다.

초기 100초간 FTP 로그인 이벤트를 발생 시켰을 경우에는 AceSniffer가 동작한 것을 확인할 수 있는데 이 때 Honey Section에서만 동작하는 것을 확인할 수 있다. 100초부터 200초까지는 키 입력 이벤트를 발생시켰는데 이때는 키로거가 동작한 것을 확인할 수 있다.

이에 반해 일반 프로세스는 가짜 이벤트의 발생과 Section과 상관없이 지속적으로 동작하는 것을 확인할 수 있다.

같은 방법으로 모든 프로세스에 대하여 탐지를 시도하였다. 앞에 3개의 키로거가 탐지되고 일반 프로세스는 탐지되지 않은 것을 확인할 수 있다. Nateon 프로세스는 사용자의 키보드 이벤트를 이용하여 현재 사용자의 상태를 변경하기 때문에 값이 임계치와 가깝게 나타났으나 모든 키 입력 이벤트를 이용하여 특수한 동작을 하는 것이 아니기 때문에 키로거로는 탐지되지 않았다.

FTP 패스워드 스틸러 또한 모두 탐지를 했지만 Alg 프로세스를 FTP 패스워드 스틸러로 오탐지했다. Alg 프로세스는 윈도우 방화벽과 관련된 프로세스로 모든 네트워크 이벤트를 이용하기 때문에 오탐한 것이다. 하지만 이러한 프로세스는 잘 알려져 있기 때문에 화이트 리스트를 이용하여 해결할 수 있다.

HoneyID로 대화형 스파이웨어 탐지

대화형 스파이웨어는 사용자의 중요한 정보를 유출시킬 수 있기 때문에 침해형 스파이웨어보다 더욱 위험하다. 침해형 스파이웨어가 PC나 기타 장비의 1차적 피해에서 머무는 반면 대화형 스파이웨어는 사용자가 직접적으로 금전적 피해를 입거나 사생활 정보가 유출되는 등 2차적 피해로 번질 수 있기 때문이다.

HoneyID는 능동적으로 대화형 스파이웨어를 탐지할 수 있는 새로운 행위 기반의 메커니즘이다. 가짜 이벤트를 발생시킨 후 Trap에서 수집하는 프로세스의 동작 정보를 이용하여 스파이웨어를 탐지하는 방식으로 실험 결과 높은 탐지율을 보이는 것을 확인할 수 있었다.

하지만 방화벽이나 정상적인 로깅 프로세스를 스파이웨어로 오탐지할 수 있어서 이를 화이트 리스트나 또 다른 방법으로 해결할 수 있는 방안이 필요하다. HoneyID는 대부분의 대화형 스파이웨어에 확장할 수 있다. 예를 들어 사용자의 계좌 정보를 탈취하는 대화형 스파이웨어를 탐지하기 위해 가상의 계좌 이체 이벤트를 발생시켜 탐지하거나, 웹 계정 정보를 탈취하는 대화형 스파이웨어를 탐지하기 위해 가상의 웹 로그인 이벤트를 만들어 탐지할 수 있을 것이다.

HoneyID는 다양한 환경에 응용될 수 있다. 독립적인 시스템으로 개발되어 사용자에게 배포할 수 있다. 또한 인터넷 뱅킹이나 온라인 상거래 같이 중요한 정보가 입력되는 시스템에서 미리 가상의 이벤트를 발생시켜 스파이웨어의 유무를 판단하여 사용자가 안정적으로 시스템을 이용하도록 할 수도 있을 것이다. 그 외에도 다양한 분야에 본 시스템을 응용하여 사용할 수 있다.

지금까지 새로운 방식의 스파이웨어 탐지 메카니즘 HoneyID를 이용하여 대화형 스파이웨어를 탐지할 수 있다는 가능성을 보았다. HoneyID를 이용하면 알려지지 않은 스파이웨어를 탐지함으로서 사용자의 정보가 외부로 유출되는 사고를 사전에 차단할 수 있을 것이다.