Journal of The Korea Society of Computer and Information
Vol. 19, No. 1, January 2014
www.ksci.re.kr
http://dx.doi.org/10.9708/jksci.2014.19.1.085

파일 DNA 기반의 변종 악성코드 탐지를 위한 유사도 비교에 관한 연구

장 은 겸*, 이 상 준**, 이 중 인***

A Study on Similarity Comparison for File DNA-Based Metamorphic Malware Detection

Eun-Gyeom Jang*, Sang Jun Lee**, Joong In Lee***

요 약

본 논문에서는 사용자 시스템이 악성 프로그램에 의해 피해를 입은 후 시그니처나 보안 패치가 나오기 전에 피해를 최소화하기 위한 방법으로 파일 DNA 기반의 행위 패턴 분석을 통한 탐지 기법을 연구하였다. 기존의 네트워크 기반의 패킷 탐지기법과 프로세스 기반 탐지 기법의 단점을 보완하여 제로데이 공격을 방어하고 오탐지를 최소화하기 위해 파일 DNA 기반 탐지기법을 적용하였다. 파일 DNA 기반 탐지기법은 악성코드의 비정상 행위를 네트워크 관련 행위와 프로세스 관련 행위로 나누어 정의하였다. 사용자 시스템에서 작동되는 프로세스의 중요한 행위와 네트워크 행위를 정해진 조건에 의해 검사 및 차단하며, 프로세스 행위, 네트워크 행위들이 조합된 파일 DNA를 기반하여 악성코드의 행위 패턴의 유사도를 분석하여 위험경고 및 차단을 통한 대응 기법을 연구하였다.

▸ Keywords : 행위기반, 유사도, DNA, 악성코드

Abstract

This paper studied the detection technique using file DNA-based behavior pattern analysis in order to minimize damage to user system by malicious programs before signature or security patch is released. The file DNA-based detection technique was applied to defend against zero day attack and to minimize false detection, by remedying weaknesses of the conventional network-based packet detection technique and process-based detection technique. For the file DNA-based detection technique, abnormal behaviors of malware were splitted into network-related behaviors and process-related behaviors. This technique was employed to check and block crucial behaviors of process and network behaviors operating in user system, according to the fixed conditions, to

•제1저자 : 장은겸 •교신저자 : 이상준
•투고일 : 2013. 11. 09, 심사일 : 2013. 12. 08, 게재확정일 : 2013. 12. 17.
* 대전대학교 컴퓨터공학과(Dept. of Computer Science, Daejeon University)
**평택대학교 환경융합시스템학과(Dept. of Integrated Environment Systems, Pyeongtaek University)
***리얼타임테크(Real Time Tech. Co., Ltd.)

www.kci.go.kr

analyze the similarity of behavior patterns of malware, based on the file DNA which process behaviors and network behaviors are mixed, and to deal with it rapidly through hazard warning and cut-off.

▸ Keywords : Behavior-Based, Similarity, DNA, Malware

I. 서 론

현재까지 알려진 대부분의 백신 소프트웨어들은 시그니처 기반 탐지 방식이다. 시그니처 기반의 탐지 방식은 악성 코드로 분류된 파일의 특정 부분, 고유한 부분이 검사의 대상이기 때문에 오탐지 및 미탐지율을 최소화 할 수 있고 파일의 특징적인 부분만 비교하기 때문에 빠르게 검사할 수 있다. 하지만 시그니처 기반 진단법은 악성 코드 파일 자체가 수백 바이트만 바뀌어도 진단이 되지 않는 경우가 발생하기 때문에 유사 악성코드에 대해서는 탐지 할 수 없는 문제점을 가지고 있다[1,2].

한편, 안티 바이러스 업계뿐만이 아니라 보안 업체에서도 주목을 받고 있는 탐지 기법 중 하나는 행동기반 탐지기법이다. 행동기반 탐지 기법은 일반적으로 컴퓨터 시스템 내에서 발생하는 행동들을 바탕을 두고 있는 시스템 기반 혹은 프로세스기반 탐지기법과 전체 네트워크 내에서 발생하는 행동들에 바탕을 두고 있는 네트워크 기반 탐지기법으로 나뉜다. 행동기반 탐지는 치명적인 비정상 행동에 탐지하는 확률은 높지만, 하나의 행동만으로 정상행동 성격이 있는 행동을 이용하는 유사 악성코드 프로그램을 탐지하지 못하는 단점이 있다[1~4].

따라서 본 논문에서는 알려지지 않은 악성코드와 변종 악성코드를 탐지하기 위해 2장에서는 기존의 공격 차단기법과 패턴 유사도 추출 방법을 기술하고 프로세스 행위패턴을 바탕으로 악성코드를 탐지하고 순서패턴의 유사도 추출 기술을 3장에서 기술하였다. 또한 4장과 5장에서는 제안 기술을 실험하여 실험 결과를 분석하였다.

II. 관련 연구

1. 악성코드 공격 형태

악성코드는 웜, 백도어, 트로이목마, 그리고 루트킷 등의 악성코드가 단독으로 발생하기 보다는 이들의 복합적인 행태를 가지고 있다. 일반적으로 웜이나 트로이목마는 고급언어로 만들어져 있고 인터넷상에 오픈소스가 존재하기 때문에 제작이 쉽다. 이는 Win32기반의 트로이목마, 악성 IRC 관련 트로이목바, 웜 등이 많이 발견되고 VBScript, 매크로바이러스 등이 점차 급감하는 이유이기도 하다. 트로이목마의 경우 그 형태가 파일 삭제에서 정보 유출, 서비스 거부(DOS), 분산 서비스거부(DDOS)등과 같이 다른 시스템을 공격하는 행태로 변화되어 가고 있다. 그리고 웜의 경우, 자가 복제되어 네트워크상에서 고속으로 전파되어 피해 규모가 매우 커지고 큰 위험성을 가지고 있다[3,4]. 최근 발생하는 웜의 특징은 다음과 같다.

- 보안 관련 소프트웨어 프로세스 강제 종료
- 공유폴더를 이용한 웜 증가
- 스팸(Spam) 메일러 등장
- 인스턴트 메시지, IRC 및 P2P 웜 출현 증가
- 주소록 이외의 파일에서 메일 주소 추출 기능

SANS Institute에 따르면, IE 취약점은 수천 대의 컴퓨터를 스파이웨어 및 애드웨어 감염에 노출시키는 주요 원인으로 되고 있다. WMF(Windows Meta File) 취약점과 최신 Windows DOC 취약점 즉, 사용자 실적에 대해 악의적인 공격을 성공적으로 실행하는데 이용되는 문제는 사전 조치를 취해야 할 필요성을 부각시키는 사례들이다.

이와 같이 취약점의 발표와 탐지 프로그램 개발과의 시간

www.kci.go.kr

간격은 피해 정도를 가늠하는 중요한 사안이다. 2000년 10월 MS00-0078이 발표되고 11개월 후 닌다(Numda) 웜이 취약점을 공격했고 2005년 8월 Zotob 웜이 MS05-039를 공격한 시기를 기점으로 이 격차는 기하급수적으로 좁혀졌다[4,5]. 그러나 기존의 보안 기술로는 대응에 어려움이 존재한다.

- 제로데이 공격: 시그니처에 의해 보호되기 이전의 취약점을 이용한 공격이다. 일반적으로 악성코드의 시그니처 또는 패치를 만드는 데 걸리는 시간은 약 7시간정도이다. 그러므로 시그니처나 패치의 대응이전의 7~30시간동안 제로데이 공격에 노출된다.
- 알려진 공격 및 변종: 이미 알려진 공격이지만 지속적으로 특정 시스템을 접근하거나 네트워크의 트래픽을 증가시키는 공격이 있다. 즉 알려져 있으므로 탐지와 방어가 가능하지만 탐지와 방어를 하기 위하여 정상적인 트래픽 처리나 응용처리를 신속하게 처리하지 못하는 문제가 야기된다. 물론 그보다 어려운 점은 알려진 공격의 변종이 발생하는 것이고 그에 대한 대응은 시간을 필요로 한다.
- 목표 공격(Targeted Attack): 최근 공격은 특정 산업, 기관 또는 기업을 대상으로 이루어진다. 목표를 정하고 취약정보를 취득하여 공격하기 때문에 공격에 대한 대응에 어려움 한계를 가지고 있다.
- Storm 웜: 제로데이 공격을 이용한 Storm 웜은 고속으로 자가 전파 특성을 가지고 있어 매우 파괴력이 높다. Storm 웜에 대한 시그티처가 개발되더라도 그 후에 수많은 변종이 발생한다. 그리고 Warhol 웜과 Flaxh 웜은 전 시계 인터넷을 15분 이내에 감염시킬 수 있다.

2. 행위 차단 기법

행위기반 탐지기법은 대부분이 행위 차단 기법(Behavior Block)이 발전한 형태이다. 행위 차단 기법은 단순하게 컴퓨터 시스템에서 발생하는 모든 행위들을 모니터링만 한다. 그러나 이 모니터링 과정에서 악성 코드와 유사한 행위가 발생하게 되면 이 행위를 유발한 실행 파일이 어떠한 것인지를 추적하여 해당 실행 파일이 더 이상 진행하지 못하도록 차단시킨 이후 컴퓨터 시스템 사용자에게 조치를 할 수 있는 경고창을 발생시킨다. 만약 사용자가 해당 실행 파일이 정상 적인 파일이라면 그 이후의 행위를 지속적으로 수행할 수 있도록 진행시키게 된다. 그러나 사용자가 악성코드로 의심되는 파일일 경우에는 해당 실행 파일이 더 이상 시스템에서 실행되지 못하도록 강제 종료를 시킬 수 있다.

그림 1은 게이트키퍼 프로그램의 기본 구조이다. 게이트키퍼는 모니터링 엔진에 의해서 현재 시스템에서 실행중인 모든 파일들에 대해서 감시하게 된다.

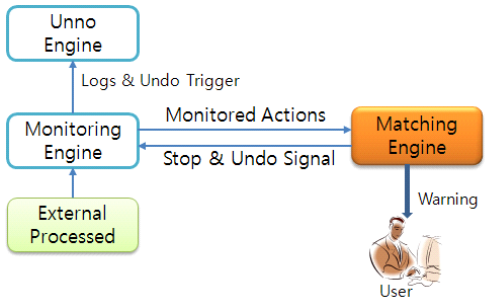

그림 1. GateKeeper의 기본 구조
Fig. 1. GateKeeper's basic structure

이 과정에서 새로운 실행 파일이 실행되면 이 모니터링 엔진은 해당 파일의 처음 시작부터 감시를 진행하며 해당 실행 파일이 어떠한 순서로 실행되는 지를 모두 추적한 후 복귀 엔진을 통하여 로그를 모두 남기게 된다. 그리고 해당 실행 파일의 추적을 통하여 생성한 로그를 비교 엔진을 통해 해당 실행 파일이 악성 코드의 실행 형태와 얼마나 유사한 행위를 유발하고 있는지를 측정하게 된다[6,7].

여기에서 만약 해당 실행 파일이 악성 코드로 판단되면 시스템 사용자에게 게이트키퍼는 현재 실행 중인 파일이 악성코드로 판단된다는 경고를 유발하게 된다. 시스템 사용자는 해당 판단을 근거로 실행 차단을 선택하게 될 경우, 복귀 엔진은 자신이 기록한 해당 실행 파일의 로그를 분석하여 실행된 순서를 역으로 시스템에 대한 복구 작업을 수행하게 된다. 이 복구 작업을 통하여 게이트키퍼는 해당 실행 파일에 의해서 발생한 시스템 변화에 대해 실행되기 전의 상태로 복구할 수 있게 된다.

이러한 형태의 행위 차단 기법은 알려지지 않은 악성 코드가 실행될 경우에는 이를 사용자게 알려주고 차단할 수 있다는 장점이 있다. 그러나 오탐지로 인하여 너무 많은 경고창이 발생되면 이는 오히려 시스템 사용자에게 어떠한 것이 알려지지 않은 악성 코드인지 정확한 판단을 하기가 어려워지게 된다. 그리고 이러한 경고 창으로 인하여 오히려 시스템 사용자는 정상적인 시스템에 불편함을 유발한다. 그리고 행위 차단 기법 역시 알려지지 않은 악성코드가 실행될 경우 이를 탐지해야 되는 윈도우 응용 프로그램이 행위 차단 기법이 적용된 시스템이 악성 코드의 실행을 정상적으로 컨트롤하지 못하게 될 경우에는 시스템이 악성 코드에 감염되는 치명적인 문제를

www.kci.go.kr

유발할 수 있다.

3. 변수간 유사도 측정 방법

① 피어슨 상관 계수

상관계수분석이란 변수간의 관련성을 분석하기 위해 사용하는 방법으로서 하나의 변수가 다른 변수와 관련성이 있는지를 알 수 있고, 또 어느 정도 관련이 있는지 알아보기 위한 방법이다. 피어슨 상관계수는 산관계수분석에서 자주 이용되는 계수이며 상관계수 r은 (-1,1)의 값을 갖는다. r의 값이 1에 가까울수록 두 변수는 양의 상관관계를 나타내게 되고 서로 유사하다는 것을 의미하는 반면 r이 -1에 가깝다면 두 변수 사이에 관계가 적음을 의미한다. N개의 원소를 갖는 두 벡터 X와 Y사이의 피어슨 상관계수는 식 1과 같이 정의된다 [6~8].

$$A^{(p)} = \frac{\sum XY - \frac{\sum X \sum Y}{N}}{\sqrt{(\sum X^2 - \frac{(\sum X)^2}{N})(\sum Y^2 - \frac{(\sum Y)^2}{N})}} \quad (1)$$

② 스피어맨 상관계수

비모수분석은 변수들이 양적 변수가 아니어도 될 때 이용할 수 있는 상관계수 분석 방법으로, 스피어맨 상관계수와 같은 방법들이 있다. 스피어맨 상관계수는 변수의 순위배열을 사용하여 변수간의 상관계수를 분석하는 방법인데 피어슨 상관계수와 마찬가지로 상관계수 [-1,1]의 값을 갖는다. 한편 스피어맨 상관계수는 X와 Y의 순위배열 Dx와 Dy를 사용하여 식2와 같이 나타낼 수 있다.

$$S^{(s)} = 1 - \frac{6\sum (Dx - Dy)^2}{N(N^2-1)} \quad (2)$$

③ 유클리드 거리

유클리드 거리는 두 변수간의 유사성을 측정하기 위한 방법이다. 두 변수간의 유사성은 거리로 나타낼 수 있는데, 거리가 가까울수록 유사성이 높다. 유클리드 거리는 두 변수간의 기하학적 공간에서의 거리를 나타내며, 거리 값이 크게 나올수록 유사한 정도가 낮은 것이기 때문에 사실 비유사성 정도를 나타낸다고 볼 수 있다. 두 벡터 X와 Y의 유클리드 거리는 식3과 같이 나타낼 수 있다.

$$S^{(e)} = \sqrt{\sum (X - Y)^2} \quad (3)$$

④ 코사인 계수

두 변수간의 유사성 측정을 위한 다른 방법으로 코사인 계수가 있다. 코사인 계수 방법에서 계수값은 [-1,1]의 범위를 갖게 되는데, 두 변수간의 유사성은 계수값이 클수록 높게 된다. 코사인계수는 두 변수 사이의 각을 측정해서 코사인 값으로 나타내어 주는데 유사성이 높을수록 각이 작고, 코사인값은 1에서 가까워지기 때문이다. X와 Y의 코사인 계수는 식4에 의하여 구할 수 있다.

$$S^{(e)} = \frac{\sum XY}{\sqrt{\sum X^2 \sum Y^2}} \quad (4)$$

데이터로부터 의미 있는 패턴을 발견하거나 클러스터링을 수행하기 위해서는 두 데이터들이 서로 얼마나 떨어져 있는가를 측정하는 것이 필요하다. 즉, 데이터들 사이의 유사도를 측정하는 것이 데이터 마이닝이나 지식발견에 있어서 중요한 단계가 되는 것이다. 지난 수십년간 서로 다른 응용분야에서 데이터들간의 유사도를 측정하는 방법에 대한 연구가 진행되어 왔다. 파일 DNA 생성을 위해서는 감염된 API의 연속적인 행위를 하나의 인자로 보고 각 행위간의 유사도 측정을 통해 그 값을 계산해야 한다[6~8].

III. 제안한 악성코드 탐지 기법

1. 악성코드 탐지 시스템 동작 원리

악성 코드 탐지 모듈의 동작구조는 그림 2와 같이 어플리케이션과 커널단으로 나누어진다. 어플리케이션단은 일반적인 응용 프로그램의 WIN32 API를 사용할 때 발생하는 운영체제의 동작으로 파일 읽기 함수를 호출하면 WIN32 API의 ReadFile이 호출되고 최종적으로 실제 드라이브에 액세스하는 커널 레벨에 있는 ZwReadFile이 호출된다.

www.kci.go.kr

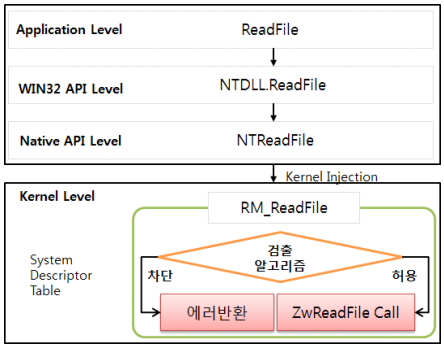

그림 2. 탐지 모듈 동작 개요
Fig. 2. Outline of detection module action

시스템의 중요한 행위를 감시하고 그것을 차단하기 위해서는 SDT(System Descriptor Table)를 본 연구에서 제안하는 탐지 모듈의 함수로 대치하여 사용하게 된다. 이것을 후킹이라 한다. 후킹은 원래 original 함수의 주소를 제안한 모듈의 함수 주소로 대치하여 original 함수는 탐지 모듈 함수 내부에서 호출한다.

후킹 시에 원래의 ZwReadFile을 탐지 모듈의 RM_ReadFile로 대치한 후 파일 DNA 검출 알고리즘에서 현재 프로세스 차단이 결정되면 에러를 발생하고 해당 행동을 못하도록 차단한다. 그리고 허용되면 ZwReadFile 운영체제 함수를 호출함으로서 정상 처리하도록 한다.

2. 탐지 모듈 구조

악성 코드 탐지 모듈은 그림 3과 같다.

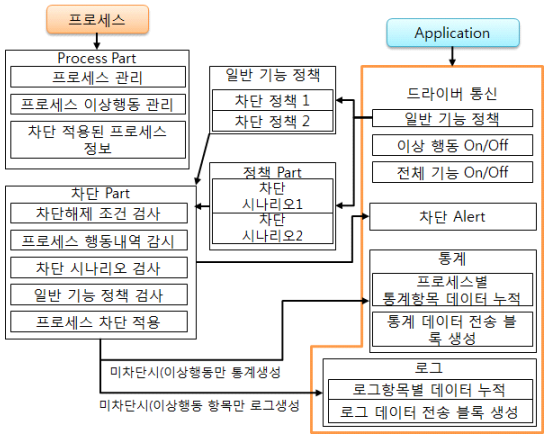

그림 3. 악성 코드 탐지 모듈
Fig. 3. Malware detection module

2.1 프로세스 영역

프로세스에서 행위가 발생되면 프로세스 id를 검증한다. 검출된 프로세스가 탐지 모듈의 프로세스 리스트에 없으면 추가하고 그림 4와 같이 해당 발생 내역을 누적한다.

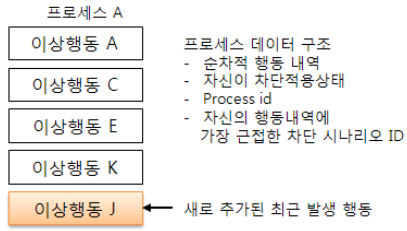

그림 4. 프로세스 이상행동 처리
Fig. 4. Handling of process abnormal behaviors

프로세스 데이터 구조에는 모든 행위에 대한 발생여부 및 실제 수치를 기록하는 곳을 가지고 있다. 디폴트로 모두 0이지만 행위가 발생할 경우에는 타입별로 악성 행위 종류 및 상세 스펙을 참조하여 기록한다.

프로세스 데이터 구조는 링크드 리스트와 같은 구조로 발생 내역을 가지며 "실행파일 생성" 행위가 한번 발생하면 노드가 추가되고 "실행파일 생성"이 발생하며 새로 노드를 추가하지 않고 수치만 업데이트 한다.

한 개 이상의 행위가 추가로 발생되면 해당 프로세스의 모든 행위 발생 내역과 일치되는 차단 시나리오를 실시간 검사하게 되지만 동일한 이상행동이 연속 2회 이상 발생할 경우에는 일치되는 차단 시나리오 검사를 수행하지 않고 행위 발생 내역을 누적하고 통계 데이터를 누적한다.

2.2 차단 영역

차단 적용이 이루어지면 시간 차단의 경우, 특정 시간동안 해당 프로세스의 네트워크가 차단되게 된다. 이때 프로세스 정보데이터가 차단되었다는 플래그가 설정되고, 이 플래그는 다음에 어떤 행위가 발생할 때마다 차단 해제에 대한 판단을 한다. 차단 시작 시간과 차단된 기간은 프로세스 정보데이터에 기록되어 관리된다. 또한 유효한 차단 시간을 지나면 차단을 해제하고 현재 새로 발생한 이상 행위에 대한 업데이트 이후 진행을 하게 된다. 차단 해제 시간이 아니라면 현재 행위를 업데이트 할 필요가 없다. 행위 업데이트들도 결국 차단을 위한 목적으로 존재하는데 이미 차단이 진행되고 있기 때문이다. 매번 이상 행위가 발생하였는지 순서와 구체적 수치가 존재한다. 이상 행위가 발행할 때마다 차단 파트가 차단 시나리

www.kci.go.kr

오에서 현재까지 발생한 행위 내역과 같은 차단 시나리오가 존재하지 않는다면 가장 근접한 차단 시나리오의 프로세스 id를 검색하여 프로세스 정보데이터에 보관한다.

3. 악성코드 유사도 측정

테스트 프로그램의 악성코드와 유사한 부분이 있는지 판단하기 위해서는 각 악성코드의 API 호출이 유사한지 판단해야 한다. API간 유사도 계산은 수식5와 같다.

$$Sim(S_i,S_j) = \frac{|S_i \cap S_j|}{|S_i \cup S_j|} \quad (5)$$

행위패턴은 하나의 집합으로 표현할 수 있다고 가정하고 이때의 집합에서 각 원소들은 ‘,’ 구문기호로 구분되는 각 API를 의미한다. 행위패턴의 교집합은 단순히 서로 중복되는 원소들의 개수로만 얻어져서는 안되며 순서에 대한 처리가 포함되어야 한다. 이러한 문제들을 해결하기 위해서 본 논문에서 사용되는 행위패턴 유사도 비교 기법은 동적 프로그래밍 기법을 응용하였다. 그림 5와 같이 교집합 계산을 위하여 먼저 가중치 행렬을 임의로 구성하였다.

행위 패턴 S_i = {A, B, C, C, A, B, B}과 행위패턴 S_j = {D, A, B, D, C, A, A, C, B}으로 임의로 설정하여 교집합을 구하기 위한 가중치 행렬을 생성하고 각 교차점의 가중치를 계산한다. 최종 교집합 개수는 이들 가중치들로 구할 수 있다.

수식6은 좌표 (x,y)에서 가중치(w)를 구하기 위한 함수이다.

$$W_{(x,y)} = 1 - \frac{((x-x_{prev})(y-y_{prev}))-1}{W_c} \quad (6)$$

Wc = 패턴의 길이, 윈도우 크기

y \ x	0	1	2	3	4	5	6	7
0	S_j \ S_i	A	B	C	C	A	B	B
1	D							
2	A	6/7						
3	B		1					
4	D							
5	C			6/7				
6	A					6/7		
7	A							
8	C				5/7			
9	B						6/7	

그림 5. 가중치 행렬
Fig. 5. Weight matrix

x와 y좌표 상에 놓인 두 행위패턴들의 공통된 요소를 교집합의 원소로 정하고, 서로 교차된 지점에서의 가중치는 이전의 두 행위패턴의 요소가 교차된 지점으로부터 거리 값에 전체 위도우의 크기를 고려하여 결정된다.

윈도우의 크기는 패턴의 길이와 동일하며 두 좌표의 거리는 윈도우의 크기를 초과할 수 없으므로 윈도우 크기를 고려하여 윈도우 안의 거리를 가중치로 설정한다.

교차점 (x,y)가 이전의 교차점(x_{prev}, y_{prev})로부터 떨어진 거리는 두 교차점 사이의 빈 블록의 개수에 비례하며, 만일 바로 이전의 교차점이 존재하지 않는다면 초기 교차점은 좌표 (0,0)이며 이 좌표의 가중치는 0값을 갖는다. 이전 좌표를 구하는 조건과 규칙은 다음과 같다.

조건 1. 현재 매치된 좌표를 (x,y)로 놓고 어떤 임의의 이전 매치된 좌표를 (x_{any}, y_{any})로 놓도록 한다.

조건 2. (x_{prev}, y_{prev})가 (x,y)의 이전 매치된 좌표로 할 때, x_{prev}는 x와 같지 않아야 하고 y_{prev}는 y와 같이 않아야 한다.

조건 3. 만일, 좌표상의 x의 순환이 끝나고 y의 순환이 끝나지 않았을 경우 x좌표를 0으로 하여 나머지 y 좌표에 대한 매치 작업을 시작한다.

규칙 1. 일치된 좌표는 $|x_{any} * y_{any}|$ 값이 $|x * y|$값보다는 작은 모든 값들 중에서 최대값을 가져야 한다.

규칙 2. 만일, 두 개 이상의 이전 매치된 좌표에서 $|x_{any} * y_{any}|$값을 가진다면, 그들 중 $|x_{any} + y_{any}|$값이 더 큰 좌표를 최종적으로 인전 매치된 좌표를 선택한다.

규칙 3. 만일 $|x_{any} * y_{any}|$값과 $|x_{any} + y_{any}|$ 값이 모두 같은 두 개 이상의 이전 매치된 좌표가 발견된다면 다음 두 단계의 계산 과정을 다시 따른다.

규칙 3-1. x좌표와 y좌표가 같을 때에는 고려할 필요 없이 이전 매치된 좌표를 선택하더라도 가중치는 같게 된다.

규칙 3-2. x좌표가 y좌표와 같지 않을 때에는 각 이전 대치된 좌표로부터 $|x-x_{any}|$값과 $|y-y_{any}|$값을 계산하여 둘 중 더 큰 값을 추출하여 모든 추출된 값중 최소값을 가지는 좌표를 최종적으로 이전 매치된 좌표로 선택한다.

최종 교집합의 수 계산은 식7과 같다.

www.kci.go.kr

$$|S_i \cap S_j| = \sum S \in M \frac{W_s}{C_s} \quad (7)$$

식7에서 M은 모든 일치된 교차점의 좌표들이고, W_s는 교집합의 원소 s가 가지는 모든 가중치들의 합이며, C_s는 두 행위패턴에서 원소 s가 중복되는 개수들의 최대값을 의미한다. 각 원소별로 두 행위패턴 Si와 Sj마다 포함된 개수들의 최대값은 A는 3개, B는 3개, 그리고 C는 2개가 됨을 알 수 있다. 이 결과와 최종적으로 얻은 모든 교차점들에서의 가중치들을 이용하여 교집합은 식8과 같다.

$$\begin{aligned} |S_i \cap S_j| &= \left(\frac{A\text{가중치합}}{A\text{개수의 최대값}}\right) + \left(\frac{B\text{가중치합}}{B\text{개수의 최대값}}\right) \quad (8) \\ &\quad + \left(\frac{C\text{가중치합}}{C\text{개수의 최대값}}\right) \\ &= \frac{\left(\frac{6}{7} + \frac{6}{7}\right)}{3} + \frac{\left(1 + \frac{6}{7}\right)}{3} + \frac{\left(\frac{6}{7} + \frac{5}{7}\right)}{2} \\ &= 1.98 \end{aligned}$$

합집합의 개수는{A, B, C, D}의 4개로 얻어질 수 있다. 따라서 유사도의 수치는 1.98 / 4 = 0.494 이다.

기존 방식의 유사성 척도와의 비교를 위해 다음과 같은 상황을 제시한다. 단순비교를 통하여 V1은 V2와 유사하지만, 순서가 정렬되지 않은 모습을 보인다. V1과 V3는 유사하지 않지만, 일정한 편차를 보인다는 점에서 여러 유사성 척도 비교시 오류 발생 확률이 높다.

V1 = (1,9,1,9,1,9,1,9,1,9,1,9)
V2 = (0,1,9,1,9,1,9,1,9,1,9,1)
V3 = (5,5,5,5,5,5,5,5,5,5,5,5)

유사도 비교를 위해 표1과 같이 V1, V2, V3간의 예를 여러 가지 알고리즘별로 계산한 결과이다. 유클리드 거리법은 서열이 정렬이 되어 있지 않은 관계로 V1, V3가 더 유사하다고 표시되는 문제가 있으며, 코사인 계수법도 유클리드 거리법과 유사한 결과를 보인다. 피어슨 상관관계법은 V1, V2가 완전히 일치하는 1의 값을 표현하여 신뢰도에 문제가 발생한다. 제안된 방법에서는 V1, V2의 유사도가 0.7778으로 완벽하게 일치하지 않음을 보이지만 상당히 유사한 패턴이다. V1, V3 관계에서는 숫자가 단 한 개도 없으므로 유사도를 0으로 산출된다. 탐지 시에 API 시퀀스의 유사도 비교는 이 함수를 사용한다.

표 1. 각종 유사도 함수의 비교 결과
Table 1. Comparison results of various similarity functions

(V_s, V_u)	$S^{(M)}$	$S^{(E)}$	$S^{(C)}$	$S^{(P)}$
V1=(1,9,1,9,1,9,1,9,1,9,1,9) V2=(9,1,9,1,9,1,9,1,9,1,9,1)	0.7778	27.713	0.2195	-1.000
V1=(1,9,1,9,1,9,1,9,1,9,1,9) V3=(5,5,5,5,5,5,5,5,5,5,5,5)	0	13.856	0.7878	0

$S^{(M)}$: 제안된 방법, $S^{(E)}$: 유클리드 거리법
$S^{(C)}$: 코사인 계수법, $S^{(P)}$: 피어슨 상관관계법

IV. 제안 시스템 실험 및 분석

1. 실험 환경

제안한 악성 코드 탐지 모듈과 기존 발표된 악성 코드와 유사 변종 코드들의 행위 패턴을 비교·분석하였다. 이를 위해 8가지의 악성 코드와 9개의 악성코드에 대한 10개 이상 100개 이하의 유사 변종 악성 코드를 적용하여 실험하였다. 실험의 주요사항은 "실행 파일 생성", "레지스트리 등록", "서비스 등록", "외부 네트워크 접속 다운로드/실행", "바로가기 생성", "외부로 전송", "포트 오픈", "팝업창 생성", "특정프로세스 종료", "뮤텍스 생성", "타 프로세스 오픈", "레지스트리 삭제", "방화벽 해제"이다.

2. 악성코드탐지 분석

2.1 유사도 측정 방법 비교

그림 6은 변종 시퀀스가 가장 많이 발생하는 Agobot을 대상으로 유사도 측정 방식을 비교한 결과이다.

피어슨 상관관계 측정법과 스피어맨 상관관계 측정법에서는 패턴과 API 순서간의 길이에 따라 큰 폭으로 차이가 발생하였으며, 제안된 유사도 측정 방법에서는 0.8 이상의 탐지 가능한 범위의 유사도 값을 추출하였다.

www.kci.go.kr

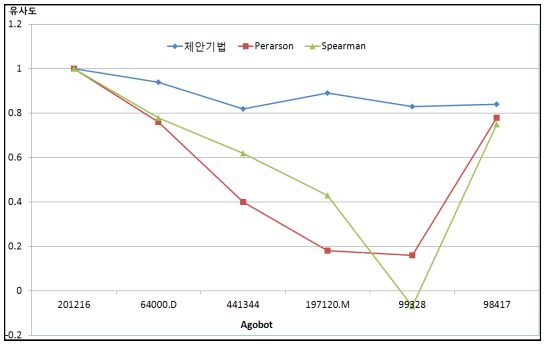

그림 6. 유사도 비교 알고리즘
Fig. 6. Similarity comparison algorithm

2.2 임계치 수준에 따른 탐지율 변화

각 유형별 유사도 임계치 설정을 위해 1부터 0까지의 순서로 수치를 낮추어 탐지율을 분석하였다. 그림 7과 같이 유사도의 임계치는 취약점형은 0.9이상, 메일형과 메신저형은 0.6이상, 봇형은 0.7 이상일 때 100% 탐지율을 보인다.

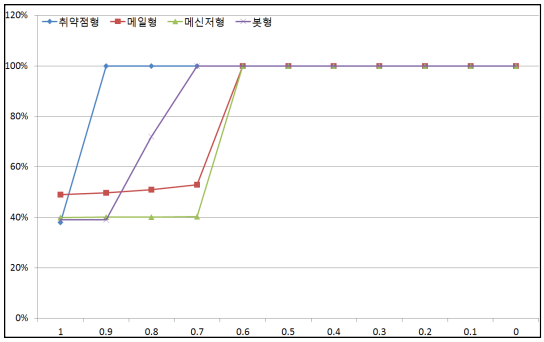

그림 7. 유사도 수치에 따른 탐지율
Fig. 7. The rate of detection depending on the similarity value

2.3 탐지시간별 유사도 비교 결과

API 순서 패턴과 호출되는 API유사도 비교는 호출되는 API의 과다발생에 따라 일정 기간의 정보를 토대로 분석이 가능하여야 한다. 그림 8은 탐지시간이 흐름에 따라 API의 유사도 변경 사항을 보인다.

분석결과 정상 프로그램은 API의 유사내용이 발생하지 않아 탐지가 되지 않음을 알 수 있고, 악성코드에 해당하는 경우, 모두 1초의 시간이 지난 뒤에는 유사도 수치가 낮아지는 결과를 얻었다. 이것은 주요자원에 정보를 등록하려는 악성코드의 반복된 API 호출에 따른 감소한 결과로 일정 시간 이상의 정보를 유지할 필요가 없다.

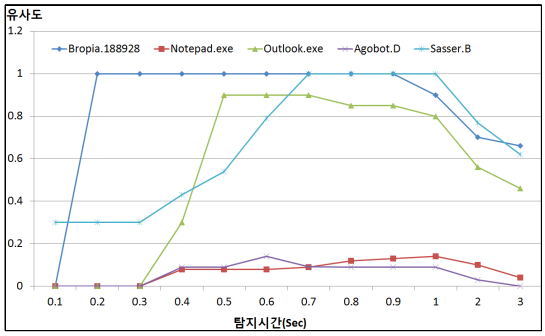

그림 8. 시간별 유사도 비교
Fig. 8. Time-based similarity comparison

2.4 동종의 유형별 웜 유사도 비교 결과

메일형 웜은 Bagle, Mydoom, Netsky 등이 있다. 동일한 종류의 악성코드일 경우 탐지가 가능함을 보인다. Bagle은 세 종류 중 가장 최근에 발견된 것으로 메일형 웜 중에서 가장 많은 변종을 발생시킨 웜이다. Bagle 웜을 기준으로 비교하였을 때 표 2와 같이 임계치로 설정한 0.6을 모두 넘는 유사도를 추출하여 비슷한 유형의 다른 웜을 탐지 가능한 결과를 얻었다.

표 2. 유형별 웜 유사도 비교 결과
Table 2. Comparison results of worm similarity of each type

구분	bagle	mydoom	netsky
bagle	1	0.767	0.700
mydoom	0.767	1	0.929
netsky	0.700	0.929	1

2.5 서비스별 패턴 유사도 비교 결과

서로 다른 서비스 간의 상호 패턴 비교는 표 3과 같다. 상호간 패턴 비교는 모두 0.4 이하의 수치를 보이며, 이는 임계치로 결정된 0.6에 해당되지 않으므로 한 종류의 패턴으로 사로 다른 유형에 대한 탐지 불가능하였다.

www.kci.go.kr

표 3. 서비스별 웜 유사도 비교 결과
Table 3. Comparison results of worm similarity of each service

구분	메일형	취약점형	봇형	메신저형
메일형	1	0.076	0.182	0.200
취약점형	0.076	1	0.231	0.245
봇형	0.182	0.231	1	0.346
메신져형	0.200	0.245	0.345	1

V. 결론

본 논문에서는 유사 악성 코드를 탐지하기 위해 행위 패턴 분석을 이용한 탐지 기법을 연구하였다. 악성코드의 변종 추출을 위해 유사도를 분석하여 메일형, 취약점형, 봇형, 메신저형의 유형별로 한 개의 행위 패턴으로 변종의 웜을 탐지 할 수 있도록 하였다. 유사도 비교시 API와 인자 필터링을 통해 탐지의 정확성을 높일 수 있었으며, Bagle을 기준으로 작성된 순서 패턴이 Netsky, Mydoom도 탐지할 수 있음을 보였다. 또한 악성코드가 아님에도 악성코드의 행위와 같은 행위로 조작하여 설정한 메일 프로그램도 탐지되었지만, 유사도 비교에는 해당하는 순서가 발견되지 않아 악성 코드가 아닌 것으로 분석되었다. 따라서 패턴이 업데이트 되지 않은 시스템에서도 탐지가 어려운 여러 가지 변종에 대해 탐지가 가능하였다.

향후, 정상 프로그램에 대한 정상 행위 분석을 통하여 안전한 프로세스 DNA를 구축하고, 다양한 종류의 악성코드를 분석하여 추가의 행위 패턴에 대한 파일 DNA를 추가한다면 더 안전한 시스템을 유지할 수 있을 것이다.

참고문헌

〔1〕 A. Sung, J. Xu, P. Chavez and S. Mukkamala, "Static Analyzer for Vicious Executable(SAVE)", 20th Annual Computer Security Application Conference, pp. 326-334, 2004.

〔2〕 Alex Ship, "Heuristic Detection of Viruses within E-mail"m Virus Bulletin Paper, 2001.

〔3〕 Baudorin Le Charlier, Morton Swimmer, Abdelaziz Mounji, "Dynamic detection and classification of computer viruses using general behavior patterns", th International Virus Bulletin Conference. Boston, September pp. 20-22, 1995.

〔4〕 John Aycock, "Computer Viruses and Mallware", 2006.

〔5〕 Matthew Evan Wagner, "Behavior Oriented Detection of Malicious Code at Run-Time", F.I.T Pater, 2004.

〔6〕 Matthew M. Williamson, "Using Behavior to Detect and Classification Information-Stealing Malware", 2005.

〔7〕 Nam-Youl Park, Yong-Min Kim, Bong-Nam Noh, "A Behavior based Detection for Malicious Code Using Obfuscation Technique", KIISC, Vol. 16, No. 3, June. 2006.

〔8〕 Eun-Gyeom Jang, "A Study on Comparison of Road Surface Images to Provide Information on Specific Road Conditions", KSCI, Vol. 17. No. 4, April. 2012.

www.kci.go.kr

저 자 소 개

장 은 겸

2007: 대전대학교
컴퓨터공학과 공학박사
2009 ~ 현재: 대전대학교
컴퓨터공학과 겸임교수
관심분야: 시스템 접근제어, 모바일앱,
컴퓨터 포렌식스, DRM
Email : janegu@dju.ac.kr

이 상 준

1989 : 한양대학교
전자계산학과 공학사.
1995 : Univ. of Utah Mechanical
Eng. 공학석사.
2003 : Arizona State Univ.
Mechanical & Aerospace
Eng. 공학박사
현 재: 평택대학교
환경융합시스템학과 조교수
관심분야: 물류정보시스템, 유비쿼터스,
RFID/USN,
Email: sjlee2026@ptu.ac.kr

이 중 인

1987 충남대학교
계산통계학과(이학사)
1992 충남대학교 대학원
전산학과(이학석사)
2002 충남대학교 대학원
전산학과(이학박사)
2009~현재 충남대학교
컴퓨터공학과 겸임교수
㈜리얼타임테크 이사
관심분야: 멀티미디어, 이러닝,
컴퓨터 통신
E-Mail: jilee@realtimetech.co.kr

www.kci.go.kr