악성프로그램 탐지를 위한 PE헤더 특성 분석 기술

최양서** · 김익균** · 오진태** · 류재철***

요 약

최근 발생하는 다양한 악성 프로그램을 분석해 보면, 해당 악성 프로그램을 쉽게 분석할 수 없도록 하기 위해 다양한 분석 방해 기법들이 적용되고 있다. 그러나, 분석 방해 기법들이 적용될수록, 악성 프로그램의 PE파일 헤더에는 정상적인 일반 PE파일의 헤더와는 다른 특징이 더 많이 나타난다. 본 논문에서는 이를 이용하여 악성 프로그램을 탐지할 수 있는 방법을 제안하고자 한다. 이를 위해, PE파일 헤더의 특징을 표현할 수 있는 특징 벡터(Characteristic Vector, CV)를 정의하고, 정상 실행 파일의 특징 벡터의 평균(ACVN)과 악성 실행 파일의 특징 벡터의 평균(ACVM)을 사전 학습을 통해 추출한다. 이후, 임의 파일의 특징 벡터와 ACVN, ACVM간의 Weighted Euclidean Distance(WED)를 계산하고, 이를 기반으로 해당 파일이 정상파일인지 혹은 악성 실행 파일인지를 판단하는 기술을 제안한다.

PE Header Characteristics Analysis Technique for Malware Detection

Yang Seo Choi** · Ik Kyun Kim** · Jin Tae Oh** · Jae Cheol Ryu***

ABSTRACT

In order not to make the malwares be easily analyzed, the hackers apply various anti-reversing and obfuscation techniques to the malwares. However, as the more anti-revering techniques are applied to the malwares the more abnormal characteristics in the PE file's header which are not shown in the normal PE file, could be observed. In this letter, a new malware detection technique is proposed based on this observation. For the malware detection, we define the Characteristics Vector(CV) which can represent the characteristics of a PE file's header. In the learning phase, we calculate the average CV(ACV) of malwares(ACVM) and normal files(ACVN). To detect the malwares we calculate the 2 Weighted Euclidean Distances(WEDs) from a file's CV to ACVs and they are used to decide whether the file is a malware or not. The proposed technique is very fast and detection rate is fairly high, so it could be applied to the network based attack detection and prevention devices. Moreover, this technique is could be used to detect the unknown malwares because it does not utilize a signature but the malware's characteristics.

Key words : Malware Detection PE Header

---

* 본 연구는 지식경제부 및 정보통신연구진흥원의 IT성장동력기술개발 사업의 일환으로 수행하였음[2008-S042-03, 네트워크 위협의 Zero-day Attack 대응을 위한 실시간 공격 Signature 생성 및 관리 기술개발].

** 한국전자통신연구원 보안게이트웨이연구팀

*** 충남대학교 컴퓨터공학과

1. 서 론

최근 공격 동향을 살펴보면, 다량의 네트워크 트래픽을 생성함으로써 특정 시스템 및 네트워크를 마비시키는 수퍼웜 형태의 공격의 발생 빈도는 매우 낮고, 특정 시스템에서 원하는 정보를 유출시키기 위한 악성프로그램을 이용한 공격이 주류를 이루고 있다[1]. 이러한 공격은 널리 사용되는 정당한 응용 프로그램을 통해 전파되고, 폴리몰픽[2] 기법 등이 적용되기 때문에, 기존의 네트워크 기반 침입탐지 기법을 통해서는 악성 프로그램 탐지가 매우 어려운 것이 사실이다. 이를 극복하기 위해 최근에는 악성 프로그램 자체를 탐지하기 위한 연구가 활발히 이루어지고 있다[3, 4].

악성 프로그램 탐지 및 분석을 위한 연구가 진행됨에 따라 해커들은 자신이 개발한 악성 프로그램이 쉽게 발견되지 않고, 분석되지 않게 하기 위해 다양한 분석 방해기법(obfuscation 및 anti-reversing기법)[5]들을 적용하고 있다. 그러나 이와 같은 분석 방해기법이 적용됨에 따라 악성 프로그램의 PE파일[8] 헤더에는 일반 정상 파일의 헤더와는 다른 특징이 나타나게 된다. 본 논문에서는 이 특징의 차이를 이용하여 악성 프로그램을 탐지하는 방법을 제안한다.

본 논문에서는 실행 PE파일 헤더의 특징을 표현하기 위해 특징 벡터(Characteristic Vector, CV)를 정의한다. 그리고, 정상 프로그램과 악성 프로그램 샘플을 이용하여 사전에 해당 프로그램의 특징벡터 평균(Average of CV, ACV)을 도출하고, 도출된 ACV와 임의 파일 F의 특징 벡터(CVF) 간의 Weighted Euclidean Distance(WED, d(ACV, CVF))[6]를 계산함으로써 이를 기반으로 악성 파일 여부를 판단한다.

본 논문에서 제안하는 악성 프로그램 탐지 방법은 다양한 분석 방해 기술이 적용된 윈도우 운영체제에서 실행 가능한 PE파일 형태의 악성 프로그램을 탐지하는 방법이다. 실제로 현재 대부분의 악성 프로그램은 윈도우 운영체제에서 동작되는 PE파일 형태이며, 또한 대부분 다양한 분석 방해 기법들이 적용되어 있다.

본 논문은 다음과 같이 구성되어 있다. 먼저 제2장에서 악성코드 탐지와 관련된 관련 연구에 대해 알아보고, 제 3장에서 공격자가 일반적으로 적용하는 분석 방해 기법에 대해 알아본다. 제 4장에서는 PE헤더 분석을 통한 악성 프로그램 탐지 기술에 대해 논의하고, 제 5장에서 실험 결과에 대해 기술하며 제 6장에서 향후 연구 방향 및 결론과 함께 본 논문을 마치도록 한다.

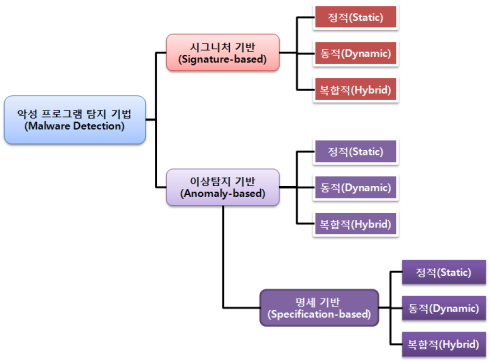
2. 관련 연구

2.1 악성 프로그램 탐지 기법 분류[9]

악성 프로그램(Malware)을 탐지하기 위한 기법은 크게 이상상태기반탐지(anomaly-based dtection)와 시그니처기반탐지(signature-based dtection)로 분류할 수 있다. 이상상태기반탐지 방법은 특정 프로그램의 악성 여부를 판단하기 위해 그 프로그램의 정상적인 동작들에 대한 지식을 이용하는 것이다. 이 유형 중 특별한 방법으로는 명세기반탐지(specification-based detection) 방법이 있는데, 이는 프로그램 수행에 있어 유효한 동작에 대한 명세나 조건을 이용하는 것이다. 시그니처기반탐지 방법은 특정 프로그램이 위험한 동작을 수행할 경우(혹은 하기 위해) 나타나는 특징을 이용하여 탐지하는 방법이다.

각 방법 별로 악성 프로그램을 탐지하기 위한 정보를 수집하기 위한 접근 방법에 따라 정적/동적/복합적 방법으로 세부 구분될 수 있다. 정적접근방법은 프로그램의 문맥(syntax)이나 구조적 특성을 이용하고 동적접근방법은 실제 프로그램이 수행 중인 프로세서의 문맥이나 구조적 특성을 이용한다. 즉 정적접근방법은 프로그램이 수행되기

전에 악성 프로그램을 탐지하기 위한 것이고 동적 접근방법은 실제 프로그램이 수행 중이거나 수행이 완료된 후에 악성 프로그램을 탐지하기 위한 방법이다. 복합적접근방법은 정적방법과 동적방법을 함께 사용하는 방식이다.



(그림 1) 악성프로그램 탐지기술 분류[9]

2.2 악성 프로그램 탐지 기법

(그림 1)의 각각의 분류에는 매우 많은 수의 악성프로그램 혹은 악성코드 탐지 기술이 존재하나, 본 논문에서는 각 분류에 속하는 기술 1가지씩을 설명한다.

2.2.1 시그니처기반-Analyzing and Detecting Malicious Mobile Code[10]

Mori는 자기 암호화가 가능한 모바일 다형 바이러스를 탐지하는 툴을 제안하였다. 전통적인 패턴 매칭기술은 이러한 종류의 바이러스를 검출할 수 없다. 다시 말하면, 바이러스 자체를 암호화함으로써 패턴 매칭 기법에서 사용하는 특정 패턴을 나타내지 않는다. 그러므로 저자는 이러한 문제를 다루기 위한 기술과 툴을 제안하였다.

Mori가 제안한 방법을 사용할 때, 코드는 운용체제 에뮬레이터에서 해독된다. 일단 바이러스가 해독되면, 시스템 호출을 식별하기 위해 정적 분석 방법을 사용한다. 검출 정책은 악성코드를 나타내는 상태 전이도 형태로 모델링되며, 사용자는 검출 정책(시그니처)를 정의한다. 시그니처와 일치하면 모바일 어플리케이션은 악성코드로 간주된다. 실험 결과에 따르면 제안된 검출 방법을 통해 600개의 바이러스/웜 샘플을 탐지할 수 있었다.

2.2.2 이상상태기반-PAYL[11]

PAYL은 시스템의 각 응용 포트별로 예상되는 페이로드 계산하여 이를 바탕으로 이상상태를 탐지한다. 학습단계에서 각 응용 포트별로 바이트별 발생빈도 확율분포함수를 생성하고 탐지단계에서는 각 응용포트별로 유입되는 페이로드를 이와 비교하여(Mahalanobis distance계산) 이상 상태를 판단한다. Mahalanobis distance는 특성 벡터의 평균뿐만 아니라 통계적 유사성을 판단할 수 있는 분산값과 Covariance값을 고려하여 계산하게 된다.

이 방법은 1999 MIT 링컨 랩 데이터를 바탕으로 진행된 평가 결과 60%의 탐지율과 1% 이하의 오탐율을 보였다. 즉 해당 데이터의 201개 공격 중에서 97개의 공격이 제안된 방법에 의해 탐지되어야만 하나 시험결과 57개의 공격에 대해 탐지가 가능하였다.

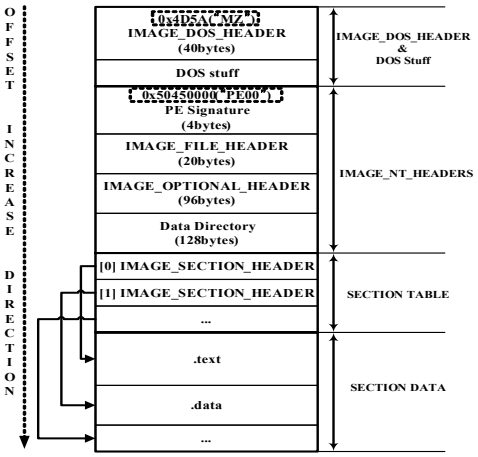
2.2.3 명세기반-Process Behavior Monitoring[12]

각 프로그램 수행 시 시스템에서 발생할 감사 데이터에 대한 명세를 기술할 수 있는 언어인 ASL (Auditing Specification Language)을 제안하고 시스템에서 수행할 각 프로그램에 대해 미리 기술된 감사 명세와 실제 시스템에서 발생하는 감사 데이터 간의 비교를 통한 이상 여부를 판단할 수 있는 방법을 제안하였다. 우선 시스템에서 실행될 프로그램에 대해서 ASL로 명세를 작성한 후 ASL 코드는 C++ 클래스로 컴파일된다. 컴파일 된 C++ 클래스들은 시스템 콜을 검사할 수 있는 기반구조

에 링크된다. 이 실행 프로그램은 시스템콜 탐지 엔진으로 불리며 호출되는 시스템 콜 마다 이를 포착하여 ASL에 의해 정의된 명세와 비교하여 실행 프로그램의 이상 여부를 판단하게 된다.

2.3 PE 파일 Header

PE파일은 마이크로소프트사의 윈도우 시스템의 기본 실행 파일 포맷이다. 윈도우 시스템에서 동작되는 대부분의 실행파일은 PE파일이며 PE파일은 (그림 2)와 같은 PE 파일 헤더를 갖는다[8].



(그림 2) PE파일 헤더[8]

(그림 2)와 같은 PE파일 헤더는 PE파일이 디스크로부터 읽혀져 메모리상에 적재될 때, 어느 위치의 어떤 정보를 메모리의 어디에 위치시키고 실행을 시작하는지 등과 같은 다양한 실행 환경 정보를 가지고 있다.

본 논문에서는 PE파일 헤더의 각 필드의 값들이 정상 PE파일인 경우 어떤 값을 갖고, 분석 방해기법이 적용된 악성 프로그램의 경우에 어떤 값들을 갖는지를 분석하여 이를 이용하여 해당 PE파일의 악성여부를 판단하게 된다.

3. 분석 방해(Anti-Reversing) 기법

최근 발견되는 대부분의 악성 프로그램에는 분석 전문가에게 쉽게 분석되지 않게 하기 위해 다양한 분석 방해 기법들이 적용되어 있다. 이는 악성 프로그램의 의도가 쉽게 노출되지 않게 하기 위해 악성 프로그램을 개발한 개발자에 의해 적용되는 방법으로 이러한 분석 방해 기법들에는 다음과 같은 기술들이 존재한다.

- 실행간 실행 코드 변경
- PE헤더 내 비정상적인 섹션 크기 할당
- PE 시그니처 위치 조정
- 정상적이지 않은 PE헤더 필드 값 사용

이와 같은 기술들이 적용되면, 해당 PE파일의 헤더 내에 특정 필드 값이 일반적인 PE파일과는 다른 값을 갖게 되어, 일반 정상 PE파일 헤더에는 자주 발생하지 않는 특징들이 나타나게 된다. 예를 들면, “실시간 실행코드 변경”을 가능하게 하기 위해서는 특정 섹션은 실행가능 속성과 쓰기 속성이 동시에 존재해야 한다. 따라서, 분석방해기법이 많이 적용되면 적용될수록, 해당 파일의 PE파일의 헤더는 일반적인 PE파일의 헤더와 다른 특징을 더 많이 갖게 된다.

4. 악성 프로그램 탐지 알고리즘

4.1 PE헤더 특징 벡터

본 논문에서는 정상적인 실행 파일과 바이러스 파일의 PE헤더에서 추출한 다양한 필드 값 및 특징들을 이용하여 해당 PE헤더의 특징을 표현하는 특징 벡터(Characteristic Vector, CV)를 정의한다.

본 논문에서 사용하는 특징 벡터 CV는 다음과 같이 도출되었다. 먼저, 1) PE헤더에서 얻을 수 있는

다양한 필드 값을 추출하고, 2) 해당 필드 값들의 평균값을 정상 PE파일과 악성 프로그램 PE파일별로 도출하고, 3) 추출된 두 평균값들을 비교하여 그 차이가 큰 경우, 이를 CV의 값으로 포함시켰다. 이때 차이가 크다는 것은 다음 두 경우를 의미한다.

- 정상 PE파일의 특징 값 == 0 and 악성 프로그램 PE 파일의 특징 값 != 0
- 10×정상 PE파일의 특징 값 ≤ 악성 프로그램 PE파일의 특징 값

상기 두 경우에 해당하는 특징벡터 원소를 총 10개를 선정하였고, 이를 특정 파일 F에 대해 표현하면, 식 (1)과 같이 표현된다.

$$CV_F = (V_{F,1}, V_{F,2}, \cdots, V_{F,10}) \quad (1)$$

4.2 평균 특징 벡터(ACV)

<표 1>과 같이 정의된 특징벡터를 기반으로 평균 특징 벡터(Average of CV, ACV)를 다음과 같이 정의한다. 주어진 N개의 파일의 평균특징벡터 ACV(N)은 식 (2)로 정의된다.

$$ACV(N) = \left(\frac{\sum_{i=1}^{N} V_{F_i,1}}{N}, \frac{\sum_{i=1}^{N} V_{F_i,2}}{N}, \cdots, \frac{\sum_{i=1}^{N} V_{F_i,10}}{N}\right) \quad (2)$$

본 ACV(N)은 차후, 악성 파일과 정상 파일들을 대상으로 그 값을 도출하여 특정 파일의 악성 여부 판단을 위한 기준값으로 활용된다.

4.3 악성파일 탐지 기법

주어진 임의 파일 F에 대하여 해당 파일의 악성 여부를 판단하기 위해서, 본 논문에서는 Weighted Euclidean Distance(WED)를 사용한다. 다시 말하면, 주어진 임의 파일 F에서 추출한 특징 벡터(CV_F)와 사전 학습을 통해 획득한 정상 파일 및 바이러스 파일의 평균 특징 벡터ACV_N과 ACV_M 간의 WED를 구하여 악성 파일 여부를 판단한다.

이때, 임의의 두 특징벡터 벡터 CV_F와 CV_L 간의 WED $d(CV_F, CV_L)$는 식 (3)과 같이 정의한다.

$$d(CV_F, CV_L) = \gamma \cdot \frac{\sqrt{\sum_{i=1}^{n} W_i \cdot (CV_{F,i} - CV_{L,i})^2}}{\sum_{i=1}^{n} W_i} \quad (3)$$

〈표 1〉 특징 벡터 목록

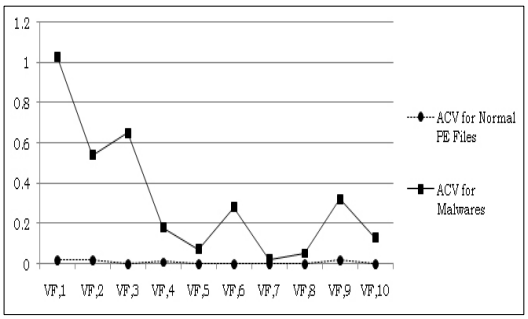
특징 값	설 명
$V_{F,1}$	실행가능 및 쓰기가능 속성을 동시에 갖고 있는 섹션의 개수
$V_{F,2}$	실행가능 속성을 갖고 있는 코드가 아닌 섹션 또는 실행가능하지 않으면서 코드인 섹션의 개수
$V_{F,3}$	섹션명이 출력 불가능한 섹션의 개수
$V_{F,4}$	실행가능한 섹션이 없는 경우 '1'
$V_{F,5}$	모든 섹션의 합이 파일 크기보다 큰 경우 '1'
$V_{F,6}$	PE 시그니처의 위치가 DOS_HEADER의 크기보다 작은 경우 1
$V_{F,7}$	PE 시그니처 위치가 512byte 보다 큰 경우 1
$V_{F,8}$	섹션 Entrypoint가 실행 파일내 섹션에 포함되지 않는 경우
$V_{F,9}$	Entrypoint 위치의 섹션이 코드가 아닌 경우
$V_{F,10}$	Data Directory의 개수가 16개가 아닌 경우

여기서, γ는 전체 결과 값의 크기를 증가시켜, 특정 WED 간의 차이를 쉽게 확인하기 위한 상수 값이며, W_i는 각 특징값에 대한 가중치로, 이는 휴리스틱하게 선정된다.

특정 파일 F의 악성 여부를 판단하기 위해서는 2개의 WED-d(CV_F, ACV_N)와 d(CV_F, ACV_M)-를 계산하고, 그 결과, 거리가 가까운 것을 해당 파일의 탐지 결과로 제시한다. 즉, 정상 파일에 대한 ACV와의 거리보다 악성 파일에 대한 ACV와 거리가 가깝다면, 해당 파일 F를 악성 파일로 판단하는 것이다.

5. 실험 결과

실험용 파일은, 정상 파일은 마이크로소프트사의 윈도우 XP의 "System32" 및 "Program Files" 디렉토리 내에 존재하는 파일과, 인터넷 상에서 수집한 589개 파일을 대상으로 하였으며, 악성 파일은 (주)안철수연구소에서 획득한 악성파일 946개를 대상으로 수행하였다.



(그림 3) 악성파일 및 정상파일 ACV

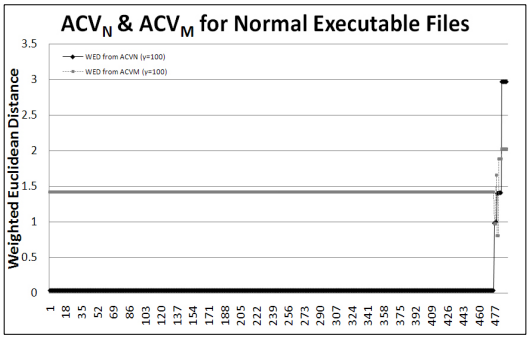
이때 임의로 선정한 정상파일 100개와 악성 파일 100개를 이용하여, 악성 여부 판단의 기준이 되는 $ACV_N(100)$과 $ACV_M(100)$을 도출하였다. 그 결과는 (그림 3)과 같다.

이후, ACV_N과 ACV_M을 이용하여 나머지 489개의 정상파일과, 846개의 악성 프로그램에 대해 악성 여부 판단 결과를 측정하였다. 실험에 사용한 파일 및 파라미터는 <표 2>와 같다.

〈표 2〉 실험 파라미터

파라미터	Values
γ (상수값)	100
Wi(가중치)	{1, 1, 10, 1, 10, 10, 10, 10, 1, 10}

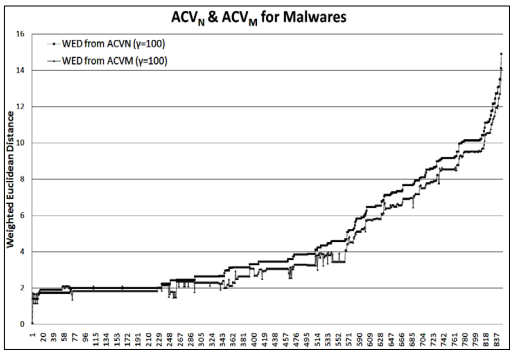
(그림 4)는 정상파일 489개에 대해 ACV_N and ACV_M와의 WED를 계산한 것이다. (그림 4)에서 확인 가능한 바와 같이 대부분의 정상 파일들이 ACV_N (점선)과 그 거리가 근접하고 있음을 알 수 있다.



(그림 4) 정상파일의 ACV_N 및 ACV_M

(그림 5)는 악성파일 846개에 대해 ACV_N and ACV_M와의 WED를 계산한 것이다. (그림 5)에서 확인 가능한 바와 같이 대부분의 악성 파일들이 ACV_M (실선)과 그 거리가 근접하고 있음을 알 수 있다.

총 846개의 악성 파일에 대하여 72개를 제외한 774개의 파일을 악성으로 판단하였으며, 489개의 정상 파일 중 10개를 제외한 479개의 파일을 정상으로 판단하였다. 이는 91.49%의 탐지 성능을 보이는 것으로, 이때 8.51%의 미탐율과 2.04%의 오탐율을 나타냈다. 실험결과를 정리하면 <표 3>과 같다.



(그림 5) 악성파일의 ACV_N 및 ACV_M

〈표 3〉 실험결과(탐지율)

	개 수	악성파일 판단	정상파일 판단
악성 파일	846	774 (91.49% : 탐지율)	72 (8.51% : 미탐율)
정상 파일	489	10 (2.04% : 오탐율)	479 (97.96%)

6. 결 론

본 논문에서는 분석방해 기법이 적용된 악성파일 탐지를 위해 PE파일 헤더를 분석하여, PE헤더 특징벡터를 정의하고, 학습을 통해 획득한 ACV를 이용하여 임의 파일의 CV와 WED (Weighted Euclidean distance)를 구하고, 이를 바탕으로 악성 여부를 판단하는 기법을 제안하였다.

제안된 기법을 이용한 실험 결과 낮은 오탐율과 높은 탐지 확률로 분석방해 기법이 적용된 악성파일을 탐지할 수 있음을 확인하였다.

앞으로는 제안된 기법의 성능분석과 오탐 및 미탐의 원인을 분석하여 탐지성능 향상을 위한 연구가 시도될 것이며, 악성파일과 정상파일간의 거리계산을 Euclidean distance가 아닌 다른 클러스터링 알고리즘들을 이용하여 보다 효율적인 방안을 개발할 계획이다.

참 고 문 헌

[1] Schneier Bruce, “Security-FOCUS : Attack Trends”, QUEUE, 2005.

[2] Korea Internet Security Center, “Korea internet Incident Trend Report”, KISA, 2007.

[3] Idike Nwokedi and P. Mathur Aditya, “A Survey of Malware Detection Techniques”, Technical Report, Purdue University, 2007.

[4] Y. M. Wang, D. Beck, B. Vo, R. Roussev, and C. Verbowski, “Detecting stealth software with strider ghostbuster”, In Proceedings of the 2005 International Conference on Dependable Systems and Networks”, pp. 368-377, 2005.

[5] Harbour Nick, “Stealth Secrets of the Malware Ninjas”, Blackhat 2007, 2007.

[6] Wikipedia, “Euclidean Distance”, http://www.en.wikipedia.org/wiki/Euclidean_metric.

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

[8] 이호동, “Windows 시스템 실행파일의 구조와 원리”, 한빛미디어, 2005.

[9] Nwokedi Idika, Aditya P. Mathur, “A Survey of Malware Detection Techniques”, Purdue University, 2007.

[10] A. Mori, T. Izumida, T. Sawada, and T. Inoue, A tool for analyzing and detecting malicious mobile code. In Proceedings of the 28th International Conference on Software Engineering, pp. 831-834, 2006.

[11] K. Wang and S. J. Stolfo, Anomalous payload-based network intrusion detection. In Proceedings of the 7th International Symposium on(RAID), pp. 201-222, 2004.

[12] R. Sekar, T. Bowen, and M. Segal, On preventing intrusions by process behavior monitoring. USENIX Intrusion Detection Workshop, 1999.

[13] C. Ko, G. Fink, and K. Levitt, Automated detection of vulnerabilities in privileged programs by execution monitoring. In Proceedings of the 10th Annual Computer Security Applications Conference, pp. 134-144, 1994.
[14] C. Kreibich and J. Crowcroft, Honeycombcreating intrustion detection signatures using honeypots. In 2nd Workshop on Hot Topics in Network, 2003.
[15] W. Lee and S. Stolfo, Data mining aroaches for intrusion detection. In Proceedings of the 7th USENIX Security Symposium, 1998.
[16] W. Li, K. Wang, S. Stolfo and B. Herzog, Fileprints : Identifying file types by ngram analysis. 6th IEEE Information Assurance Workshop, 2005.
[17] C. M. Linn, M. Rajagopalan, S. Baker, C. Collberg, S. K. Debray, and J. H. Hartman, Protecting against unexpected system calls. Usenix Security Symposium, 2005.

최 양 서

1996년 강원대학교 전자계산학과 (공학사)
2000년 서강대학교 컴퓨터공학과 (공학석사)
2000년~현재 한국전자통신 연구원 선임연구원

김 익 균

1994년 경북대학교 컴퓨터공학과 (공학사)
1996년 경북대학교 컴퓨터공학과 (공학석사)
1996년~1999년 한국전자통신 연구원
2000년~2001년 (주)팍스콤 선임연구원
2004년~2005년 Purdue University 객원연구원
2001년~현재 한국전자통신연구원 선임연구원

오 진 태

1986년~1990년 경북대학교 전자공학과(공학사)
1990년~1992년 경북대학교 전자공학과(석사)
1992년~1998년 한국전자통신 연구원 선임연구원
1998년~1999년 MinMax Tech. 연구원
1999년~2001년 Engedi Networks Inc. Director
2001년~2003년 Winnow Networks Inc. CTO, 부사장, Cofounder
2003년~현재 한국전자통신연구원 선임연구원, 보안게이트웨이연구팀 팀장

류 재 철

1988년 Iowa State University 전산학과(석사)
1990년 Northwestern University 전산학과(박사)
1991년~현재 충남대학교 정보통신공학부 교수
1997년~현재 한국정보보호학회 이사
2001년~현재 국가정보원 정보보호시스템 인증위원회 위원
2003년~현재 인터넷침해대응기술연구센터장