



(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2016-0031588

(43) 공개일자 2016년03월23일

(51) 국제특허분류(Int. Cl.)

G06F 21/56 (2013.01)

(21) 출원번호 10-2014-0120942

(22) 출원일자 2014년09월12일

심사청구일자 2014년09월12일

(71) 출원인

고려대학교 산학협력단

서울특별시 성북구 안암로 145, 고려대학교 (안암동5가)

(72) 발명자

김휘강

서울 서초구 바우피로 91, 111동 1105호 (양재동, 우성아파트)

장재욱

서울특별시 영등포구 가마산로 540, 104동 107호 (신길동, 바다마을아파트)

(뒷면에 계속)

(74) 대리인

김동용

전체 청구항 수 : 총 10 항

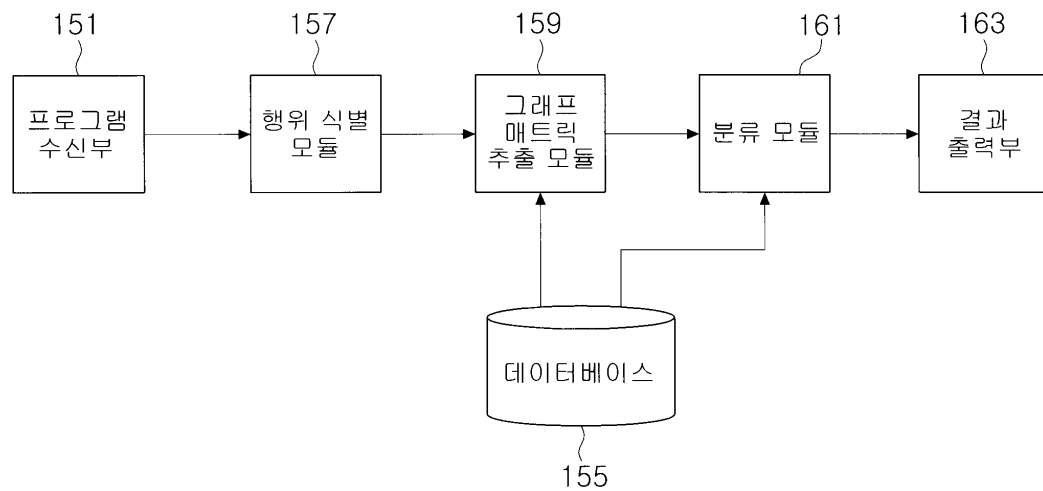
(54) 발명의 명칭 악성코드 분류 장치 및 악성코드 분류 방법

(57) 요약

본 발명은 대상 프로그램으로부터 악성 행위와 관련된 지정된 시스템 콜과 이웃 시스템 콜을 추출하고 추출된 시스템 콜들을 이용하여 시스템 콜 그래프를 생성하는 행위 식별 모듈 및 시스템 콜 그래프로부터 사회망 분석(Social Network Analysis : SNA) 기법에 적용가능한 그래프 매트릭 특성을 추출하는 그래프 매트릭 생성 모듈을 포함하는 악성코드 분류 장치 및 악성코드 분류 방법에 관한 것이다.

본 발명의 사회망 분석 기법의 적용으로 악성코드를 용이하게 식별 및 분류할 수 있는 효과가 있다.

대표도 - 도2



100

(72) 발명자

윤재성

경기도 고양시 일산동구 성현로 659번길 186, D동
402호(사리현동, SL그린빌)

우지영

충남 아산시 배방읍 용연로 37, 205동 603호 (용연
마을휴먼시아2단지아파트)

이 발명을 지원한 국가연구개발사업

과제고유번호 H0301-14-1004

부처명 미래창조과학부

연구관리전문기관 정보통신산업진흥원

연구사업명 대학 IT 연구센터 육성지원

연구과제명 제어시스템 보안연구

기 여 율 1/1

주관기관 고려대 산학협력단

연구기간 2014.01.01 ~ 2014.12.31

명세서

청구범위

청구항 1

대상 프로그램으로부터 악성 행위와 관련된 지정된 시스템 콜과 이웃 시스템 콜을 추출하고 추출된 시스템 콜들을 이용하여 시스템 콜 그래프를 생성하는 행위 식별 모듈; 및

상기 시스템 콜 그래프에서 사회망 분석(Social Network Analysis : SNA) 기법에 적용가능한 그래프 매트릭 특성을 추출하는 그래프 매트릭 추출 모듈;을 포함하는,

악성코드 분류 장치.

청구항 2

제1항에 있어서,

상기 행위 식별 모듈은 상기 대상 프로그램을 수행하여 시스템 콜 시퀀스를 생성하고 생성된 시스템 콜 시퀀스의 시스템 콜들과 미리 정의된 악성 시스템 콜의 비교로 상기 지정된 시스템 콜을 추출하고 상기 추출된 시스템 콜과 1 홉(One-Hop) 이웃 시스템 콜을 추출하여 하나 이상의 그래프 컴포넌트를 가지는 시스템 콜 그래프를 생성하는,

악성코드 분류 장치.

청구항 3

제1항에 있어서,

상기 그래프 매트릭은 평균 진입 차수 중심성(average in-degree centrality), 차수가 1인 시스템 콜의 비율 및 평균 거리(average distance) 중 하나 이상을 포함하는,

악성코드 분류 장치.

청구항 4

제3항에 있어서,

악성 클래스 내에서 추출된 그래프 매트릭 특성을 이용하여 상기 대상 프로그램에 대해 정상 프로그램인지 또는 악성 프로그램인지를 분류하는 분류 모듈;을 더 포함하는,

악성코드 분류 장치.

청구항 5

제4항에 있어서,

상기 분류 모듈은 상기 대상 프로그램을 상기 악성 클래스 내의 악성 패밀리 중 하나로 분류하거나 정상 프로그램으로 분류하며,

상기 그래프 매트릭은 상기 악성 패밀리들 각각 및 정상 프로그램 사이를 분류할 수 있는 매트릭인,

악성코드 분류 장치.

청구항 6

제4항에 있어서,

상기 그래프 매트릭은 평균 진입 차수 중심성, 차수가 1인 시스템 콜의 비율 및 평균 거리를 모두 포함하고,

상기 분류 모듈은 각 악성 클래스별로 부여되는 평균 진입 차수 중심성, 차수가 1인 시스템 콜의 비율 및 평균 거리의 중요도를 상이하게 할당하여 상기 대상 프로그램을 정상 프로그램과 악성 패밀리들 중 하나로 분류하는,

악성코드 분류 장치.

청구항 7

제6항에 있어서,

상기 분류 모듈은 악성 클래스내에서 악성코드 샘플들과 정상 프로그램 샘플들을 이용하여 상기 중요도를 미리 결정하고 상기 중요도에 기초하여 상기 대상 프로그램을 악성 클래스 내의 악성 패밀리들 중 하나 또는 정상 프로그램으로 분류하며,

악성 클래스별 각 그래프 매트릭의 중요도는 다른 클래스별 각 그래프 매트릭의 중요도와 상이한,

악성코드 분류 장치.

청구항 8

대상 프로그램으로부터 악성 행위와 관련된 지정된 시스템 콜과 이웃 시스템 콜을 추출하고 추출된 시스템 콜들을 이용하여 시스템 콜 그래프를 생성하는 단계; 및

상기 시스템 콜 그래프로부터 사회망 분석(Social Network Analysis : SNA) 기법에 적용가능한 그래프 매트릭 특성을 추출하는 단계;를 포함하는,

악성코드 분류 방법.

청구항 9

제8항에 있어서,

악성 클래스 내에서 추출된 그래프 매트릭 특성을 이용하여 상기 대상 프로그램에 대해 정상 프로그램인지 또는 악성 프로그램인지를 분류하는 단계;를 더 포함하는,

악성코드 분류 방법.

청구항 10

제9항에 있어서,

상기 그래프 매트릭은 평균 진입 차수 중심성, 차수가 1인 시스템 콜의 비율 및 평균 거리(average distance)를 모두 포함하고,

상기 분류 단계는 각 악성 클래스별로 부여되는 평균 진입 차수 중심성, 차수가 1인 시스템 콜의 비율 및 평균 거리의 중요도를 상이하게 할당하여 상기 대상 프로그램을 정상 프로그램과 악성 패밀리들 중 하나로 분류하는,

악성코드 분류 방법.

발명의 설명

기술 분야

[0001]

본 발명은 악성코드 분류 장치 및 악성코드 분류 방법에 관한 것으로서, 구체적으로는 사회망 분석방법을 적용하여 악성코드의 탐지 및 분류를 가능케 하는 악성코드 분류 장치 및 악성코드 분류 방법에 관한 것이다.

배경 기술

[0002]

백신제품 벤치마킹 테스트 전문기관인 AV-TEST의 분석보고서에 따르면, 2012년부터 2013년 동안에 변종 포함 4700만개의 새로운 악성코드가 출현하였다. 악성코드 제작자는 악성행위를 위한 핵심코드를 재사용하면서 악성코드의 변종을 쉽게 만든다.

[0003]

백신회사는 하루에도 수 천개의 신종 악성코드를 주로 시그니처(Signature) 기반의 탐지방법으로 분석하고 있다. 백신회사는 악성코드의 인스트럭션(instruction) 코드 및/또는 바이너리(binary) 코드로부터 악성코드의 고유 특징을 추출하여 시그니처를 생성 및 데이터베이스에 저장한다.

[0004]

한편 다형화(Polymorphism)와 변형성(Metamorphism) 기술이 적용된 악성코드는 시그니처 기반의 악성코드 탐지

기법을 쉽게 우회할 수 있고, 악성코드 분석가는 이러한 악성코드의 시그니처를 추출하기 위해서는 언패킹 및 난독화 해독과정을 선행하여야 한다. 시그니처 기반의 탐지기법은 분석가가 끊임없이 데이터베이스를 업데이트 해야하고 알려지지 않은(Zero-day) 공격의 경우 탐지가 용이치 않고 악성코드 행위는 유지된 채 악성코드의 일부 또는 악성코드의 전체를 변경한 악성코드의 경우 탐지율이 떨어지는 문제가 있다.

[0005] 이러한 시그니처 기반의 탐지기법을 해결하기 위한 선행연구는 주로 인스트럭션 셋 기반, API/시스템 콜 기반, 콜 그래프 매칭기반의 악성코드 정적/동적 분석방법으로 분류될 수 있다.

[0006] 인스트럭션 셋 기반 및 API 기반의 악성코드 정적 분석방법은 시그니처 추출을 위해 난독화 등 안티 악성코드 분석기술을 극복해야 하고 자동화에 제한이 있어 기하급수적으로 증가하는 악성코드의 대응에 한계가 있다. 또한 인스트럭션 셋 기반 및 API/시스템 콜 기반의 동적 분석방법은 주로 악성코드의 탐지에 관한 것으로 악성코드의 분류에 대해서는 한계가 존재한다.

[0007] 콜 그래프 매칭기반의 분석방법은 변종을 포함한 악성코드의 구조적인 유사도 탐지가 가능하나 유사도 비교를 위해서는 노드와 간선(에지)으로 구성된 각 악성코드의 콜 그래프 전체를 비교해야 하기 때문에 높은 연산능력이 필요하고 악성코드의 구조적인 정보 표현에 한계가 있다.

[0008] 이와 같이 선행 연구들에서의 한계점을 보완하기 위해 새로운 접근 방법으로 악성코드의 탐지 및 분류를 가능케 하는 악성코드의 탐지 및 분류 방법이 필요하다.

발명의 내용

해결하려는 과제

[0009] 본 발명은 상술한 문제점을 해결하기 위해서 안출한 것으로서, 사회망 분석 방법론에서 이용되는 그래프 매트릭을 활용하여 악성코드의 탐지 및 분류를 가능하게 하는 악성코드 분류 장치 및 악성코드 분류 방법을 제공하는 데 그 목적이 있다.

과제의 해결 수단

[0010] 상기와 같은 목적을 달성하기 위한 악성코드 분류 장치는 대상 프로그램으로부터 악성 행위와 관련된 지정된 시스템 콜과 이웃 시스템 콜을 추출하고 추출된 시스템 콜들을 이용하여 시스템 콜 그래프를 생성하는 행위 식별 모듈 및 시스템 콜 그래프로부터 사회망 분석(Social Network Analysis : SNA) 기법에 적용가능한 그래프 매트릭 특성을 추출하는 그래프 매트릭 추출 모듈을 포함한다.

[0011] 또한 상기와 같은 목적을 달성하기 위한 악성코드 분류 방법은 대상 프로그램으로부터 악성 행위와 관련된 지정된 시스템 콜과 이웃 시스템 콜을 추출하고 추출된 시스템 콜들을 이용하여 시스템 콜 그래프를 생성하는 단계 및 시스템 콜 그래프로부터 사회망 분석(Social Network Analysis : SNA) 기법에 적용가능한 그래프 매트릭 특성을 추출하는 단계를 포함한다.

발명의 효과

[0012] 상기와 같은 본 발명에 따른 악성코드 분류 장치 및 악성코드 분류 방법은 사회망 분석 방법론에서 이용되는 그래프 매트릭을 활용하여 악성코드의 탐지 및 분류를 가능하게 하는 효과가 있다.

도면의 간단한 설명

[0013] 도 1은 악성코드 분류 장치의 예시적인 하드웨어 블록도를 도시한 도면이다.

도 2는 악성코드 분류 장치의 예시적인 기능 블록도를 도시한 도면이다.

도 3은 악성코드 탐지 및 분류를 위한 흐름도를 도시한 도면이다.

도 4는 특정 악성 클래스 내에서 분류되는 패밀리들의 진입 차수 분포(In-degree distribution)를 나타내는 도면이다.

도 5는 도 4의 도면에 대한 로그 스케일의 진입 차수 분포(In-degree distribution)를 나타내는 도면이다.

도 6은 세 가지 악성코드 클래스(adware, trojan, worm) 각각에 대해 패밀리별로 진입 차수가 1인 콜의 분포(Portion of call with in-degree of 1) 및 진출 차수가 1인 콜의 분포(Portion of call with out-degree of

1)를 도시한 도면이다.

도 7은 특정 악성 클래스 내에서 패밀리 별 평균 진입 차수 중심성(Average in-degree centrality)을 나타내는 도면이다.

도 8은 특정 악성 클래스 내에서의 전체 클러스터링 계수(Overall clustering coefficient)의 분포를 도시한 도면이다.

도 9는 특정 악성 클래스 내에서 패밀리 별 평균 거리(Average distance)의 분포를 도시한 도면이다.

도 10은 악성 클래스와 정상 프로그램의 분류를 위한 각 그래프 매트릭별 정보 획득량(Information gain)을 도시한 도면이다.

발명을 실시하기 위한 구체적인 내용

- [0014] 상술한 목적, 특징 및 장점은 첨부된 도면을 참조하여 상세하게 후술 되어 있는 상세한 설명을 통하여 더욱 명확해 질 것이며, 그에 따라 본 발명이 속하는 기술분야에서 통상의 지식을 가진 자가 본 발명의 기술적 사상을 용이하게 실시할 수 있을 것이다. 또한, 본 발명을 설명함에 있어서 본 발명과 관련된 공지 기술에 대한 구체적인 설명이 본 발명의 요지를 불필요하게 흐릴 수 있다고 판단되는 경우에 그 상세한 설명을 생략하기로 한다. 이하, 첨부된 도면을 참조하여 본 발명에 따른 바람직한 실시 예를 상세히 설명하기로 한다.
- [0015] 도 1은 악성코드 분류 장치(100)의 예시적인 하드웨어 블록도를 도시한 도면이다.
- [0016] 도 1에 따르면 악성코드 분류 장치(100)는 통신 인터페이스(101), 입력 인터페이스(103), 출력 인터페이스(105), 메모리(107), 하드디스크(109), 하나 이상의 프로세서(111) 및 시스템 버스/제어 버스(113)를 포함한다. 도 1에 따른 악성코드 분류 장치(100)는 PC(Personal Computer), 인터넷 망을 통해 액세스 가능한 서버(Server), 노트북 등일 수 있다.
- [0017] 각 하드웨어 블록들을 간단히 살펴보면, 통신 인터페이스(101)는 악성코드 분류 장치(100) 외부의 장치와 통신할 수 있도록 구성된다. 통신 인터페이스(101)는 유선 또는 무선 통신을 수행하도록 구성되며 예를 들어 무선 랜이나 유선 랜에 연결될 수 있도록 하는 맥(MAC) 칩을 포함한다.
- [0018] 입력 인터페이스(103)는 악성코드 분류 장치(100)를 제어하기 위한 사용자 입력을 수신한다. 입력 인터페이스(103)는 키보드, 마우스 등을 포함하여 본 악성코드 분류 장치(100)를 제어할 수 있다. 예를 들어 입력 인터페이스(103)는 악성코드인지 또는 특정 악성코드 클래스 내에서 어떤 패밀리인지를 분류할 대상 프로그램의 선택, 대상 프로그램에 대한 탐지 및 분류 시작 및 종료를 결정할 수 있는 사용자 입력을 수신한다.
- [0019] 출력 인터페이스(105)는 악성코드 분류 장치(100)에 의해서 탐지 및 분류된 것을 알 수 있는 처리 결과를 출력한다. 출력 인터페이스(105)는 디스플레이, 스피커 등으로 포함한다.
- [0020] 메모리(107)는 휘발성 메모리 및/또는 비휘발성 메모리를 포함한다. 메모리(107)는 각종 데이터와 프로그램을 저장한다. 휘발성 메모리는 각종 데이터와 프로그램의 조각(segment)을 임시로 저장하고 비휘발성 메모리는 셋업 또는 부팅에 필요한 각종 셋업 데이터와 부팅 프로그램 등을 영구히 저장하고 있다.
- [0021] 하드디스크(109)는 대용량 저장매체로서 각종 데이터를 저장하고 각종 프로그램들을 저장한다. 하드디스크(109)는 적어도 본 발명에 따른 악성코드의 탐지 및 분류를 위한 악성코드 분류 방법을 수행하는 프로그램을 포함한다. 하드디스크(109)는 또한 악성코드 분류 방법의 프로그램에 이용될 데이터들을 포함한다. 이러한 데이터들은 특정 프로그램에 의해서 액세스 및 관리될 수 있다. 악성코드 분류에 이용되는 데이터들과 이 특정 프로그램은 데이터베이스(155)로 바람직하게 구성된다.
- [0022] 프로세서(111)는 비휘발성 메모리나 하드디스크(109) 등에 저장된 프로그램의 프로그램 코드를 로딩하여 프로그램 코드를 실행시킬 수 있다. 프로세서(111)는 프로그램 코드의 명령어(instruction)를 실행할 수 있는 실행 유닛(Execution Unit)을 포함하여 악성코드 분류를 위한 프로그램, 악성코드 분류에 이용되는 각종 데이터를 트래킹하기 위한 프로그램 등을 실행할 수 있다.
- [0023] 시스템 버스/제어 버스(113)는 각 하드웨어 블록 사이에 (제어) 데이터를 송수신할 수 있도록 구성된다. 시스템 버스/제어 버스(113)는 병렬 버스이거나 시리얼 버스 등일 수 있다.

- [0024] 도 2는 악성코드 분류 장치(100)의 예시적인 기능 블록도를 도시한 도면이다.
- [0025] 도 2에 따르면 악성코드 분류 장치(100)는 프로그램 수신부(151), 데이터베이스(155), 행위 식별 모듈(157), 그래프 매트릭 추출 모듈(159), 분류 모듈(161) 및 결과 출력부(163)를 포함한다. 악성코드 분류 장치(100)는 도 1에서 살펴본 하드웨어 상에서 수행되고 바람직하게는 하드디스크(109) 등에 저장된 프로그램 코드를 프로세서(111)에 로딩하여 프로그램 코드에 따라 각 하드웨어 블록을 제어하여 수행된다.
- [0026] 각 기능블록을 살펴보기 전에, 본 발명에 따른 주요 기술적 특징을 여기서 살펴보도록 한다.
- [0027] 네트워크 분석 및 그래프 이론과 같은 사회 과학 및 사회 과학 이론 분야에 사회망 분석(Social Network Analysis : SNA) 기법이 활용되고 있다. SNA 기법은 그래프 형태로 해결 대상 문제를 네트워크 구조로 변환하고 그래프 분야에서 잘 알려진 그래프 알고리즘을 이용하여 문제를 해결한다.
- [0028] 그래프 이론(Graph Theory)은 추상화 개념(abstract concepts)을 제공하고 그래프 분석을 위한 방법을 제공한다. 사회 과학 분야의 SNA는 사회 구조를 멤버와 멤버를 연결하는 에지를 가지는 네트워크(그래프, 이하 혼용하여 사용하도록 함)로 추상화한다. 이러한 SNA는 다양한 분야의 문제를 해결할 수 있도록 한다.
- [0029] 사회 과학 분야에 적용되는 SNA에 유사하게, 악성코드(맬웨어(malware)라고도 지칭함)가 수행될 때 악성코드는 미리 정의되는 세트의 시스템 콜들을 순차적으로 호출한다. 이러한 시스템 콜들은 일련의 시스템 콜 시퀀스로 에뮬레이션(emulation)을 통해 기록될 수 있다. 시스템 콜 시퀀스로부터 시스템 콜 그래프가 본 발명에 따라 구성된다.
- [0030] 시스템 콜 그래프는 방향성 그래프(directed graph) $G = \{V, E\}$ 로 정의되고, V 는 노드의 세트를 나타내고 E 는 에지의 세트를 나타낸다. 노드는 불러진(invoked) 시스템 콜을 나타내고 에지는 시스템 콜 시퀀스에 의해 결정되는 노드 간 연결을 나타낸다. $E = \{ \langle v_i, v_j \rangle \mid v_i, v_j \in V \}$ 에서 에지의 v_i 는 호출하는 콜러(caller)를 나타내고 v_j 는 호출되는 콜리(callee)를 나타낸다.
- [0031] 다시 본 발명에 따라 구현되는 시스템 콜 그래프를 정의하면, 시스템 콜 그래프는 4개의 원소로 정의되는 방향성 그래프($g = \{ V_g, E_g, L_g, I_g \}$)이다. V_g 는 시스템 콜을 나타내는 노드(버텍스(vertex)라고도 지칭됨) 세트이고, $E_g \subseteq V_g \times V_g$ 는 노드 간 호출 관계를 나타내는 에지 세트이고, L_g 는 각 시스템 콜을 식별할 수 있는 라벨(label) 세트를 나타내고, I_g 는 라벨을 노드에 할당하는 라벨링 함수를 나타낸다.
- [0032] 프로그램의 실행에 따라 콜 시퀀스가 생성되면 본 발명은 이를 SNA 툴을 통해 사회망 특성 분석을 위해 시스템 콜 그래프로 변환한다.
- [0033] 특정 프로그램을 수행시키면 이 프로그램은 일련의 시스템 콜을 호출한다. 예를 들어 특정 파일을 열고 기록하기 위해서 프로그램은 연속적으로 NtCreateFile, NtCreateSection, NtMapViewofSection 및 NtWriteFile을 호출한다. 중간 NtCreateSection 및 NtMapViewofSection은 소스 및 목적지 시스템 콜 사이의 다리(bridge) 역할을 수행하는 중간 시스템 콜로서 출발지나 목적지 혹은 그 사이에 위치한 다른 시스템 콜과 상호 작용(interaction)한다.
- [0034] 이러한 시스템 콜 시퀀스는 시스템 콜 그래프로 표현될 수 있다. 앞서 살펴본 예에서 시스템 콜 시퀀스는 $\{ \langle \text{NtCreateFile}, \text{NtCreateSection} \rangle, \langle \text{NtCreateSection}, \text{NtMapViewofSection} \rangle, \langle \text{NtMapViewofSection}, \text{NtWriteFile} \rangle \}$ 로 변환되고 이로부터 시스템 콜 그래프를 구성할 수 있다.
- [0035] 본 발명은 SNA 기법에서 적용가능한 특성으로 악성코드에서 발견되는 시스템 콜의 영향력을 측정하기 위해서 중심성 측정(centrality measure) 및 응집성 측정(cohesion measure) 방법을 이용한다. 중심성 및 응집성 모두

그래프 또는 네트워크의 특성을 정량할 수 있는 지표이다. 또한 SNA 기법에서 일반적으로 활용하는 차수 분포(degree distribution) 또한 그래프 또는 네트워크의 특성을 정량화할 수 있는 지표이다.

[0036] 중심성 측정 방식으로 중심성은 차수 중심성(degree centrality) 및 중계 중심성(betweenness centrality) 등이 있고 응집성 측정 방식으로 응집성은 클러스터링 계수(clustering coefficient), 평균 거리(average distance), 네트워크 밀도(network density) 및 컴포넌트 비율(component ratio) 등이 있다. 그 외 차수 분포(degree distribution)가 그래프의 특성을 정량화하기 위한 지표로 활용될 수 있다.

[0037] 중심성은 그래프의 노드 관점에서의 영향력을 나타내는 지표이고 응집성은 전체 그래프(즉 네트워크) 관점에서 영향력을 나타내는 지표이다. 이러한 중심성 및 응집성의 일부 특성들은 본 발명에 따라 악성코드의 탐지 및 분류를 위해서 이용된다.

[0038] 나열된 특성들을 포함하여 본 발명에서 이용된 특성들을 간단히 살펴보면 차수 분포(degree distribution)는 네트워크 구조를 알 수 있도록 하고 가장 기본적인 네트워크(그래프) 특성을 나타낸다. 차수 분포는 전체 네트워크 상에서 차수들의 분포 비율을 나타낸다. 대부분의 노드들은 낮은 차수를 가지는 반면 작은 수의 노드들은 높은 차수를 가진다. 이렇게 차수가 높은 노드들은 허브(hub)로 알려져 있다. 높은 차수의 노드들은 높은 영향력을 가진다. 로그 스케일로 차수 분포를 직선(straight line)으로 대략 표현될 때 이 분포는 멱함수(power law) 분포를 따른다고 지칭될 수 있다.

[0039] 차수 중심성은 방향성 연결 관점에서 노드가 다른 노드와 어떻게 연결되었는지를 나타내는 척도(기준)이다. 즉 차수 중심성은 노드가 다른 노드에 대해서 가지는 링크(에지)의 개수를 나타낸다. 방향성 그래프는 진입 차수(in-degree) 중심성과 진출 차수(out-degree) 중심성을 가진다. 진입 차수 중심성은 노드로 향하는 에지의 개수를 나타내고 진출 차수 중심성은 노드에서 다른 노드로 향하는 에지의 개수를 나타낸다. 진입 차수 중심성 관점에서 많은 에지를 수신하는 노드는 지배적인(prominent) 것으로 특징화될 수 있고 진출 차수 중심성 관점에서 많은 에지를 가지는 노드는 영향력이 있다는 것으로 특징화될 수 있다.

[0040] 중계 중심성은 그래프에서 다른 노드간 루트상에서 특정 노드가 브리지로 역할을 담당하는 정도에 대한 척도이다. 중계 중심성은 다른 노드간의 통신을 제어할 위한 노드의 잠재적인 인덱스로서 사용된다.

[0041] 노드 i에 대한 중계 중심성 X_i 는

수학식 1

$$X_i = \sum_{s,t} \frac{n_{st}^i}{g_{st}}$$

[0042]

[0043] 여기서, n_{st}^i 는 node i를 통해 노드 s로부터 노드 t로의 최단(shortest) 패스(path)의 개수를 나타내고

g_{st} 는 노드 s로부터 노드 t까지의 모든 최단 패스의 개수를 나타낸다.

[0044] 클러스터링 계수는 한 노드의 이웃(neighbor) 노드인 두 노드가 서로 이웃일 평균 가능성을 나타내는 척도이자 그래프 상에서 삼각형(triangles)의 밀도를 나타내는 척도이다.

[0045] 평균 거리는 그래프 상에서 모든 노드 페어(pair) 사이에서 최단 패스의 개수에 대한 모든 노드 페어 사이의 최단 패스의 에지 개수의 비율을 나타낸다.

[0046] 네트워크 밀도는 모든 노드 페어 사이의 에지의 총 개수에 대한 그래프에서의 에지의 개수 비율을 나타낸다. 다중 에지 또는 셀프 에지가 그래프에 없는 것으로 가정하면, 네트워크 밀도(p)는

수학식 2

$$p = \frac{m}{nC_2} = \frac{2m}{n(n-1)} = \frac{k}{n-1}$$

[0047]

[0048]

로 표현된다. 여기서 n 은 노드의 개수를 나타내고, m 은 에지의 개수를 나타내고 k 는 그래프에서의 평균 차수를 나타낸다. 노드의 개수가 무한대가 됨에 따라 네트워크 밀도는 k/n 이 되고 평균 차수는 상수(constant)가 된다.

[0049]

컴포넌트 비율은 노드의 개수에 대한 그래프 컴포넌트의 개수의 비로 정의된다. c 가 컴포넌트의 개수를 나타낼 때, 컴포넌트 비율은 $(c-1)/(n-1)$ 로 정의되고 n 은 노드의 개수를 나타낸다.

[0050]

정의를 통해서 살펴본 차수 분포, 차수 중심성, 중계 중심성, 클러스터링 계수, 평균 거리, 네트워크 밀도 및 컴포넌트 비율 등은 일반적으로 사회망 분석(Social Network Analysis : SNA) 기법에서 적용가능한 그래프 매트릭(metric)이다. 이러한 그래프 매트릭은 각종 악성코드들의 분석을 통해서 기존 사회망 분석뿐 아니라 악성코드의 탐지 및 분류를 위해서도 활용 가능함을 알 수 있었다.

[0051]

여기서, 악성코드의 분석을 통해서 사회망 분석 기법에서 적용되는 그래프 매트릭들 중에서 악성코드의 탐지 및 분류에 이용될 그래프 매트릭을 결정하도록 한다. 실제로 입수 가능한 악성코드들과 정상 프로그램들을 이용하여 악성코드의 탐지 및 분류에 이용될 그래프 매트릭이 결정되었다.

[0052]

분석에 이용된 악성코드 클래스는 애드웨어(adware), 트로이잔(trojan) 및 웜(worm)으로 분류되고, 애드웨어 클래스 내에는 Win32.FakeInstaller, Win32.ScreenSaver, Win32.SideTab의 패밀리로 분류되고, 트로이잔 클래스 내에는 Win32.LIac, Win32.Pakes, Win32.Regrun의 패밀리로 분류되고 웜 클래스 내에는 Win32.Mydoom, Win32.Mytob, Win32.Zwr로 분류된다. 각 패밀리에는 적어도 수십개 이상의 악성코드와 동일한 패밀리의 변종이 존재한다. 악성코드 클래스 및 패밀리는 물론 변경될 수 있다.

[0053]

차수 분포 : 악성코드의 시스템 콜 분석을 통해 악성코드에서 발견되는 높은 진입 차수를 가지는 시스템 콜은 주로 로컬 프로시저 콜을 생성하는 것과 동기화에 관련된다는 것임을 발견하였다. 반면에 정상 프로그램에서 진입 차수가 1인 시스템 콜은 파일 생성, 파일 오픈, 섹션 생성 등에 관련된 시스템 콜임을 알 수 있었다. 빈도(frequency)나 시퀀스로부터 결정되는 시스템 콜 패턴은 프로그램의 행위(behavior)를 결정하기 때문에, 악성 행위의 수행시에 악성코드는 이 악성코드에서 발견되는 시스템 콜을 정상 프로그램에서 발견되는 시스템 콜보다 더 많이 호출한다. 따라서 만일 악성코드에서 발견되는 높은 차수의 시스템 콜을 시스템 콜 그래프로부터 삭제하면 악성행위를 나타내는 네트워크(그래프)는 붕괴될 수 있다. 여기서 악성코드에서 발견되는 시스템 콜들이 차수 분포와 같은 시스템 콜 그래프의 네트워크 구조에 상당한 영향을 가진다고 가정할 수 있다.

[0054]

도 4는 특정 악성 클래스 내에서 분류되는 패밀리들의 악성코드 각각의 시스템 콜들의 진입 차수 분포(in-degree distribution)를 나타내고 있다. 도 5는 도 4에 대한 로그 스케일의 진입 차수 분포(in-degree distribution)를 나타낸다. 이 특정 악성 클래스는 예를 들어 애드웨어(adware)일 수 있고, 각각의 패밀리들은 예를 들어 Win32.FakeInstaller, Win32.ScreenSaver 및 Win32.SideTab을 나타낸다.

[0055]

도 4 및 도 5에서 알 수 있는 바와 같이, 차수가 1인 시스템 콜의 부분(portion)은 악성코드 패밀리들 사이에서 상당히 다른 것을 알 수 있다. 따라서 차수가 1인 시스템 콜의 비율 또는 부분은 악성코드 패밀리를 분류하기 위한 좋은 특성이 된다. 그리고 악성코드의 시스템 콜의 차수 분포는 멱함수(power-law) 분포를 대략적으로 따를 수 있다.

[0056]

도 4에서 알 수 있는 바와 같이 진입 차수가 1인 시스템 콜의 부분(비율)이 진입 차수가 1을 초과하는 시스템 콜의 부분(비율)보다 더 크고 비율이 악성코드 패밀리별로 상이(도 5 참조)하기 때문에 진입 차수가 1인 시스템 콜의 비율의 차이는 악성코드 분류를 위한 주요 매트릭으로 이용될 수 있다. 진출 차수가 1인 시스템 콜의 비율 또한 진입 차수가 1인 시스템 콜의 분포와 유사하므로, 진출 차수가 1인 시스템 콜의 비율 또한 주요 매트릭으로 이용될 수도 있다.

[0057]

도 6은 세 가지 악성코드 클래스(adware, trojan, worm) 각각에 대해 패밀리별로 진입 차수가 1인 시스템 콜의 비율 및 진출 차수가 1인 시스템 콜의 비율을 도시한 도면이다. 도 6에서 알 수 있는 바와 같이 진입 차수가 1

인 시스템 콜의 비율 및 진출 차수가 1인 시스템 콜의 비율은 클래스 내에서 다른 패밀리를 구분할 수 있는 주요 측도로서 이용될 수 있다.

- [0058] 평균 차수 중심성 : 악성코드의 시스템 콜의 차수 중심성을 여기서 살펴보면, 각 시스템 콜들의 진입 차수의 합은 진출 차수의 합과 동일하다. 평균 차수 중심성(Average in-degree centrality)은 이 두 차수의 중심성 평균의 두배가 된다. 평균 진입 차수 중심성만 살펴보면, 평균 진입 차수 중심성은 악성코드 클래스 내의 악성 패밀리들을 서로 분류할 수 있음을 알 수 있었다.
- [0059] 도 7은 특정 악성 클래스 내에서 패밀리 별 평균 진입 차수 중심성(Average in-degree centrality)을 나타내는 도면인데, 이 특정 악성 클래스는 애드웨어를 나타낸다. 도 7에서 알 수 있는 바와 같이 평균 진입 차수 중심성은 여러 동일 또는 변종의 악성코드들을 가지는 하나의 패밀리를 다른 패밀리와 구별할 수 있도록 한다. 또한 평균 진입 차수 중심성은 또한 트로이잔(trojan)과 웜(worm) 클래스 내에서 패밀리 분류를 위한 좋은 지표가 됨을 알 수 있었다. 이와 같이, 평균 진입 차수 중심성(Average in-degree centrality)만 살펴보면, 평균 진입 차수 중심성은 악성코드 클래스 내의 악성 패밀리들을 서로 분류할 수 있음을 알 수 있었다.
- [0060] 중계 중심성(Betweenness Centrality) : 평균 중계 중심성을 이용하여 한 악성 클래스 내의 패밀리 분류 여부를 판단한 결과, 패밀리별 분류가 용이치 않음을 알 수 있었다. 따라서 중계 중심성은 일반적으로 악성 클래스 내에서 패밀리 분류를 위한 좋은 지표가 되지 못한다.
- [0061] 응집성 측도 : 다음으로 전체 클러스터링 계수, 평균 거리, 네트워크 밀도, 밍 컴포넌트 비율을 사용하여 악성코드의 시스템 콜 그래프의 구조 정보를 획득하였다.
- [0062] 그 획득 결과 전체 클러스터링 계수(전체 시스템 콜 개수로 클러스터링 계수를 평균화한 값)은 도 8에서 알 수 있는 바와 같이 한 악성 클래스(애드웨어) 내에서 패밀리를 분류하기 위한 좋은 지표가 되지 않았다.
- [0063] 반면에 평균 거리는 애드웨어, 트로이잔 또는 웜 악성 클래스 내에서 각 패밀리를 분류하기 위한 좋은 지표(도 9 참조)가 됨을 알 수 있었다. 도 9의 악성 클래스는 애드웨어를 나타낸다. 그러나 네트워크 밀도 및 컴포넌트 비율은 분석 결과 악성 클래스 내에서 패밀리를 구분하기 위한 용도로 적합하지 않다.
- [0064] 이와 같은 그래프 매트릭들에 대한 기존의 악성코드의 분석을 통해서 적어도 평균 진입 차수 중심성, 진입 차수가 1인 시스템 콜의 비율(또는 부분) 및 평균 거리는 특정 대상 프로그램을 악성코드의 클래스 내에서 패밀리로 또는 정상 프로그램으로 분류할 수 있는 중요 지표(측도)가 됨을 알 수 있었다.
- [0065] 기존 악성코드에서 사회망 분석에 이용되는 그래프 매트릭에 대한 분석을 토대로 도 2의 블록들을 살펴보면, 프로그램 수신부(151)는 통신 인터페이스(101)로부터 또는 입력 인터페이스(103)를 경유하여 하드디스크(109)로부터 악성코드 탐지 및 분류 대상인 프로그램(이하 '대상 프로그램'이라고 지칭할 수 있음)을 수신한다. 수신된 프로그램은 행위 식별 모듈(157)로 전달된다.
- [0066] 행위 식별 모듈(157)은 프로그램 수신부(151)를 통한 대상 프로그램으로부터 악성 행위와 관련된 미리 지정된 시스템 콜과 이 지정된 시스템 콜에 이웃한 이웃 시스템 콜을 추출하고 추출된 시스템 콜을 이용하여 시스템 콜 그래프를 생성한다.
- [0067] 구체적으로, 행위 식별 모듈(157)은 대상 프로그램을 지정된 시간 동안(예를 들어 30초) 프로세서(111)에서 수행시킨다. 그리고 행위 식별 모듈(157)은 대상 프로그램이 수행되는 동안 호출되는 시스템 콜 시퀀스를 생성한다. 이러한 시스템 콜 시퀀스의 생성은 툴을 이용하여 이루어질 수 있고 예를 들어 알려져 있는 NtTrace를 이용하여 이루어진다.
- [0068] 행위 식별 모듈(157)은 데이터베이스(155)에 저장될 수 있고 미리 지정된 악성 시스템 콜 사전에 메모리(107) 등에 로딩하여, 악성 시스템 콜 사전의 악성 시스템 콜들과 시스템 콜 시퀀스의 시스템 콜들을 비교한다. 순차적인 비교에 따라 행위 식별 모듈(157)은 악성 시스템 콜 사전에 미리 지정된 시스템 콜이 전체 시스템 콜 시퀀스 내에 존재하는 경우 해당 시스템 콜을 추출하고 또한 해당 시스템 콜의 전후에 위치하는 시스템 콜, 즉 1 홉 이웃 시스템 콜,을 추출하여 하나 이상의 그래프 컴포넌트를 가지는 시스템 콜 그래프를 생성한다.
- [0069] 악성 시스템 콜 사전은 미리 데이터베이스(155)에 저장되어 있을 수 있고 악성코드 클래스별로 미리 준비되어 있을 수 있다. 예를 들어 애드웨어 클래스를 위한 악성 시스템 콜 사전과 트로이잔 클래스를 위한 악성 시스템 콜 사전과 웜 클래스를 위한 악성 시스템 콜 사전은 서로 다를 수 있고 각 클래스의 시스템 콜의 분석을 통해서 결정될 수 있다.

- [0070] 행위 식별 모듈(157)에 의해서 적어도 시스템 콜 그래프는 클래스별 악성 행위를 유발할 것으로 예상되는 시스템 콜과 이웃 시스템 콜로 구성된 그래프 컴포넌트가 하나 이상 존재한다.
- [0071] 그래프 매트릭 추출 모듈(159)은 행위 식별 모듈(157)에 의해서 생성된 시스템 콜 그래프로부터 사회망 분석 기법에 일반적으로 적용 가능한 그래프 매트릭의 특성을 해당 시스템 콜 그래프로부터 추출한다. 그래프 매트릭 추출 모듈(159)에서 이용되는 그래프 매트릭은 평균 진입 차수 중심성, 차수가 1인 시스템 콜의 비율(또는 부분), 평균 거리 중에서 하나 또는 그 조합을 포함한다. 차수가 1인 시스템 콜의 비율은 바람직하게 진입 차수가 1인 시스템 콜의 비율일 수 있다. 이러한 (평균 진입) 차수 중심성, 진입 차수가 1인 비율, 및 평균 거리는 특정 악성 클래스 내의 패밀리를 구별하기에 적합한 매트릭이다. 그래프 매트릭에 따라 그래프 매트릭 추출 모듈(159)은 대상 프로그램의 시스템 콜 그래프로부터 그래프 매트릭 특성을 추출하고 그래프 매트릭 특성은 예를 들어 하나 혹은 복수의 값으로 표현될 수 있다.
- [0072] 그래프 매트릭 추출 모듈(159)은 예를 들어 네트워크(그래프) 특성을 추출할 수 있는 툴을 이용하여 구성될 수 있다. 이 툴은 예를 들어 Ucinet 6일 수 있다.
- [0073] 분류 모듈(161)은 분석을 위한 특정 하나의 악성 클래스 내에서 추출된 그래프 매트릭 특성을 이용하여 대상 프로그램이 정상(benign) 프로그램인지 또는 악성 프로그램(코드)인지를 분류한다. 분류 모듈(161)은 나아가 악성 클래스 내의 악성 패밀리들 중 하나로 대상 프로그램을 분류하거나 정상 프로그램으로 분류할 수 있다. 분류 모듈(161)에서 이용되는 그래프 매트릭은 악성 패밀리들 각각과 정상 프로그램 사이를 분류할 수 있는 매트릭으로서, 예를 들어 평균 진입 차수 중심성, 차수가 1인 시스템 콜의 비율(또는 부분), 평균 거리 중 하나 또는 그 조합으로 구성된다.
- [0074] 분류 모듈(161)은 평균 진입 차수 중심성, 차수가 1인 시스템 콜의 비율 및 평균 거리의 조합으로 악성 클래스별로 특정 악성 패밀리로 또는 정상 프로그램으로 분류할 수 있는 데, 악성 클래스별로 평균 진입 차수 중심성, 차수가 1인 시스템 콜의 비율 및 평균 거리에 부여되는 중요도를 상이하게 할당하여 대상 프로그램을 분류할 수 있다.
- [0075] 행위 식별 모듈(157), 그래프 매트릭 추출 모듈(159) 및 분류 모듈(161)은 대상 프로그램에 대해서 각각의 악성 클래스에서 대상 프로그램이 정상 프로그램인지를 또는 특정 악성 클래스 내의 특정 악성 패밀리로 분류할 수 있다. 행위 식별 모듈(157), 그래프 매트릭 추출 모듈(159) 및 분류 모듈(161)은 각 악성 클래스별로 상이하게 동작할 수 있다. 예를 들어 행위 식별 모듈(157)에서 이용되는 악성 시스템 콜 사전은 악성 클래스별로 상이하고 그래프 매트릭 추출 모듈에서 이용되는 그래프 매트릭은 악성 클래스별로 서로 상이할 수 있고 분류 모듈(161)에서 할당되는 그래프 매트릭 별 중요도는 악성 클래스별로 상이할 수 있다.
- [0076] 분류 모듈(161)은 머신 러닝 알고리즘을 적용한 툴 또는 프로그램 일 수 있다. 예를 들어 분류 모듈(161)은 확률(probabilistic) 기반, 룰(rule) 기반, 기능(function) 기반, 인스턴스(instance) 기반, 트리 기반(tree-based) 등의 머신 러닝 알고리즘을 적용한 분류기(classifier)일 수 있다. 분류 모듈(161)은 자동으로 그래프 매트릭 추출 모듈(159)로부터 전달된 그래프 매트릭 특성을 이용하여 특정 악성 클래스 내의 특정 패밀리로 분류하거나 정상 프로그램으로 분류할 수 있다. 분류 모듈(161)은 다양한 알고리즘이 이미 알려져 있어서 상세한 설명은 생략한다.
- [0077] 결과 출력부(163)는 분류 모듈(161)을 통해 분류된 대상 프로그램의 분류를 분석 결과로서 출력 인터페이스(105)나 통신 인터페이스(101)를 통해 출력한다.
- [0078] 앞서 설명한 분류 모듈(161)은 악성 클래스별로 적용이 되고, 트레이닝 모듈(도면 미도시)과 테스트 모듈(도면 미도시)을 포함한다. 트레이닝 모듈은 학습용의 정상 프로그램 샘플과 악성코드 샘플(프로그램 수신부(151)로부터 수신될 수 있음)을 이용하여 분류 모듈(161)을 학습시키고, 학습을 통해 결정된 각 그래프 매트릭의 중요도는 테스트 모듈에 적용되어 이전에 학습이 되지 않은 정상 프로그램 샘플과 악성코드 샘플을 통해 검증과정을 거친다. 이후 트레이닝과 검증 과정 후에 분류 모듈(161)은 프로그램 수신부(151)을 통해 수신된 대상 프로그램이 정상(benign) 프로그램인지 또는 악성 프로그램(코드)인지를 바람직하게 분류한다.
- [0079] 머신 러닝 기반의 분류 모듈(161)은 정확한 샘플들을 통해 미리 학습될 필요가 있고 트레이닝 모듈은 이러한 학습에 이용될 샘플들과 각 샘플의 클래스와 패밀리 정보를 제공함으로써 분류 모듈(161)을 학습시킨다.
- [0080] 트레이닝 모듈에 의한 분류 모듈(161)의 학습시에 또한 각각의 그래프 매트릭 별로 상대적 중요도(relative importance)를 달리 적용할 필요가 있다. 특히 악성 클래스내 패밀리와 정상 프로그램 사이의 분류를 잘

할 수 있는 그래프 매트릭에 더 많은 가중치를 부여할 필요가 있다.

[0081] 정보 획득량(Information Gain)으로서 각 그래프 매트릭의 중요도를 산출한 결과, 도 10과 같은 정보 획득량을 얻을 수 있었다. 정보 획득량(IG(G,A))은

수학식 3

$$[0082] \quad IG(C,A)=H(C)-H(C|A)$$

[0083] 로 산출되고, H(C)는

수학식 4

$$[0084] \quad H(C)=-\sum_{c \in C} p(c) \log_2 p(c)$$

[0085] 로 산출되며, H(C|A)는

수학식 5

$$[0086] \quad H(C|A)=-\sum_{a \in A} p(a) \sum_{c \in C} p(c|a) \log_2 p(c|a)$$

[0087] 로 산출된다. 여기서 상대적 중요도인 정보 획득량, IG(C,A),은 그래프 매트릭 A가 주어졌을 때 주어진 악성 클래스 C에서 엔트로피(entropy) 감소 양의 척도를 나타낸다. 엔트로피 감소 양은 그래프 매트릭 A를 추가함에 따라 획득된 추가적인 정보를 나타낸다. 수학식 3 내지 5에서, H(C)와 H(C|A)는 클래스 C에서 특징인 그래프 매트릭 A의 추가 이전과 추가 이후의 엔트로피를 각각 나타낸다.

[0088] 도 10에서 알 수 있는 바와 같이 특정 그래프 매트릭은 악성코드의 탐지 및 분류에 다른 그래프 매트릭에 비해 높은 영향력을 가지는 것을 알 수 있다. 특히 (평균) 진입 차수 중심성과 진입 차수가 1인 시스템 콜의 비율 및 진출 차수가 1인 시스템 콜의 비율이 다른 그래프 매트릭에 비해서 분류 영향력이 큰 것임을 알 수 있다.

[0089] 도 10에서 또한 알 수 있는 바는, 악성 클래스(애드웨어, 트로이잔, 웜 등) 별로 동일한 그래프 매트릭의 중요도(IG(C,A))가 서로 다른 것을 알 수 있다. 이는 악성 클래스의 특성에 따른 것으로 이러한 중요도는 미리 결정될 수 있다. 미리 결정되는 중요도는 트레이닝에 의해서 바람직하게 학습되나 필요시에는 입력 인터페이스(103)를 통해 수신될 수도 있다. 그리고 이 중요도는 새로운 악성코드가 특정 악성 클래스 내의 특정 패밀리에 추가됨에 따라 업데이트될 수 있다.

[0090] 즉, 트레이닝 모듈은 분류 모듈(161)로 악성 클래스별 트레이닝 세트를 분류 모듈(161)로 제공하고, 분류 모듈(161)은 하나의 악성 클래스내에서의 트레이닝 세트에 포함된 악성코드 샘플과 정상 프로그램 샘플을 이용하여 그래프 매트릭별 중요도를 미리 결정한다. 이후 분류 모듈(161)은 미리 결정된 그래프 매트릭 별 중요도에 기초하여 프로그램 수신부(151)를 통해 수신된 대상 프로그램이 정상 프로그램인지 또는 악성코드인지를 분류하고 만일 악성코드인 경우 어떠한 악성 패밀리인지를 분류할 수 있다.

[0091] 데이터베이스(155)는 프로그램의 분류를 위해 필요한 각종 데이터를 저장한다. 데이터베이스(155)는 예를 들어 악성 클래스별 악성 시스템 콜 사전, 악성 클래스별 악성 패밀리 리스트, 각 악성 패밀리별 그래프 매트릭 특성 분포 등을 저장할 수 있다. 데이터베이스(155)는 그래프 매트릭 추출 모듈(159) 및 분류 모듈(161)에 의해서 액세스될 수 있고 액세스한 모듈(159,161)로 악성 시스템 콜 사전이나 패밀리별 그래프 매트릭 특성 분포를 필요시 전달할 수 있다.

[0092] 이상 도 2를 통해서 살펴본 바와 같이, 본 발명에 따른 악성코드 분류 장치(100)는 사회망 분석 기법에서 널리

활용되는 그래프 매트릭을 이용해서 분류 및 탐지 대상인 프로그램이 악성인지 또는 특정 악성 패밀리인지를 결정할 수 있도록 구성된다.

- [0093] 실험 결과 99% 이상의 정확도로 대상 프로그램을 분류할 수 있었고 하나의 프로그램의 탐지 및 분류에 약 20초 내외의 시간이 걸렸음을 알 수 있었다.
- [0094] 이와 같이 본 발명에 따른 악성코드 분류 장치(100)는 사회망 분석 분야에서 이용되는 그래프(네트워크) 특징 중 적은 수의 특징만을 이용해서 간단하게 악성코드의 탐지 및 분류가 가능하다. 또한 본 발명은 사회망 분석에 관련되어 단지 통계학적 값만이 필요하고 다른 추가적인 계산상 오버헤드가 부가되지 않는다. 또한 본 발명은 애플리케이션 등을 통한 동적 분석 기법뿐 아니라 정적 분석 기법에도 용이하게 적용할 수 있다.
- [0095] 도 3은 악성코드 탐지 및 분류를 위한 흐름도를 도시한 도면이다.
- [0096] 도 3의 흐름도 중 전부 또는 일부는 바람직하게 프로그램으로 구현될 수 있고 이 프로그램은 프로세서(111)상에서 수행될 수 있다.
- [0097] 이미 도 2를 통해서 악성코드 탐지 및 분류에 대해서 상세히 살펴보았으므로 여기서는 간단하게 살펴보도록 한다.
- [0098] 먼저 악성코드 분류 장치(100)는 탐지 및 분류 대상인 대상 프로그램을 통신 인터페이스(101)나 입력 인터페이스(103)를 경유하여 수신(S110)한다.
- [0099] 이후 악성코드 분류 장치(100)는 대상 프로그램으로부터 하나의 악성 클래스의 악성 행위와 관련된 지정된 시스템 콜과 이웃 시스템 콜을 추출하고 추출된 시스템 콜들을 이용하여 시스템 콜 그래프를 생성(S120)한다.
- [0100] 시스템 콜 그래프의 생성 이후에, 악성코드 분류 장치(100)는 시스템 콜 그래프로부터 사회망 분석 기법에 적용 가능한 그래프 매트릭 특성을 추출한다. 이 그래프 매트릭은 평균 진입 차수 중심성, 차수가 1인 시스템 콜의 비율 및 평균 거리 중 하나 또는 조합으로 구성된다. 바람직하게는 그래프 매트릭은 이 3가지 특성 모두를 포함한다.
- [0101] 추출된 그래프 매트릭 특성을 이용하여 대상 프로그램에 대해 정상 프로그램인지 또는 악성 프로그램인지를 악성코드 분류 장치(100)가 하나의 악성 클래스 내에서 분류(S140)한다.
- [0102] 여기서, 악성코드 분류 장치(100)는 악성 클래스별로 대상 프로그램을 분류하고 악성 클래스별 분류 결과를 통합한 분석 결과를 생성할 수 있다. 예를 들어 악성코드 분류 장치(100)는 악성 클래스별로 단계 S120 내지 단계 S140을 수행하여 모든 악성 클래스에 대해서 정상 프로그램으로 결정된 경우, 대상 프로그램을 정상 프로그램으로 분류할 수 있다. 또는 하나 이상의 악성 클래스 내에서 악성 패밀리로 분류된 경우, 해당 악성 패밀리 정보를 분석 결과로서 구성할 수 있다. 만일 복수의 악성 클래스에서 각각 특정 악성 패밀리로 분류된 경우에는 이를 통합하여 분석 결과로 구성할 수도 있다. 또는 악성코드 분류 장치(100)는 가장 높은 확률을 가지는 단일의 악성 패밀리로 대상 프로그램을 분류할 수 있다.
- [0103] 이후 악성코드 분류 장치(100)는 통신 인터페이스(101)나 출력 인터페이스(105)를 통해 분석 결과를 출력(S150)한다.
- [0104] 이상으로 살펴본 본 발명의 악성코드 분류 장치 및 방법은 바람직하게는 PC를 타겟으로 하는 악성코드를 탐지하고 분류할 수 있다.
- [0105] 이상에서 설명한 본 발명은, 본 발명이 속하는 기술 분야에서 통상의 지식을 가진 자에게 있어 본 발명의 기술적 사상을 벗어나지 않는 범위 내에서 여러 가지 치환, 변형 및 변경이 가능하므로 전술한 실시 예 및 첨부된 도면에 의해 한정되는 것이 아니다.

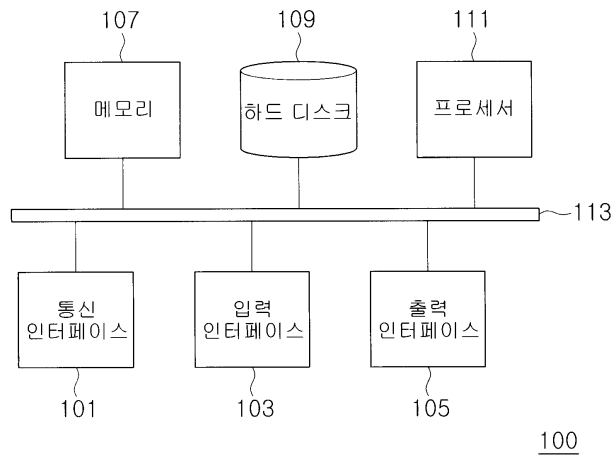
부호의 설명

- [0106] 100 : 악성코드 분류 장치
101 : 통신 인터페이스 103 : 입력 인터페이스

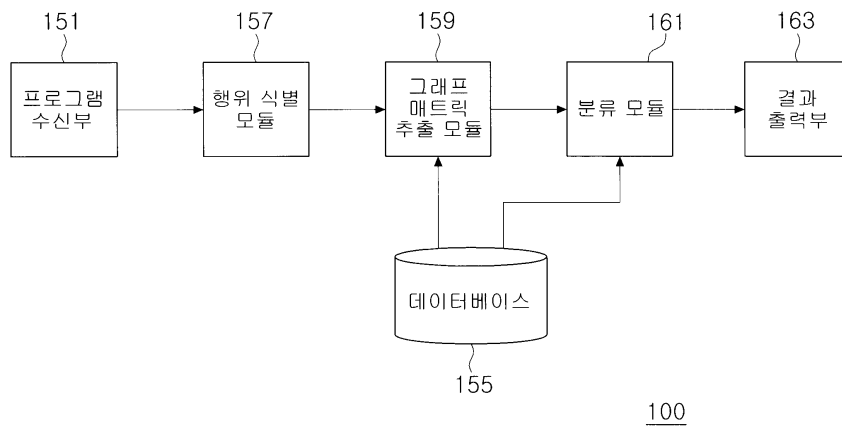
105 : 출력 인터페이스 107 : 메모리
 109 : 하드디스크 111 : 프로세서
 113 : 시스템 버스/제어 버스
 151 : 프로그램 수신부 155 : 데이터베이스
 157 : 행위 식별 모듈 159 : 그래프 매트릭 추출 모듈
 161 : 분류 모듈 163 : 결과 출력부

도면

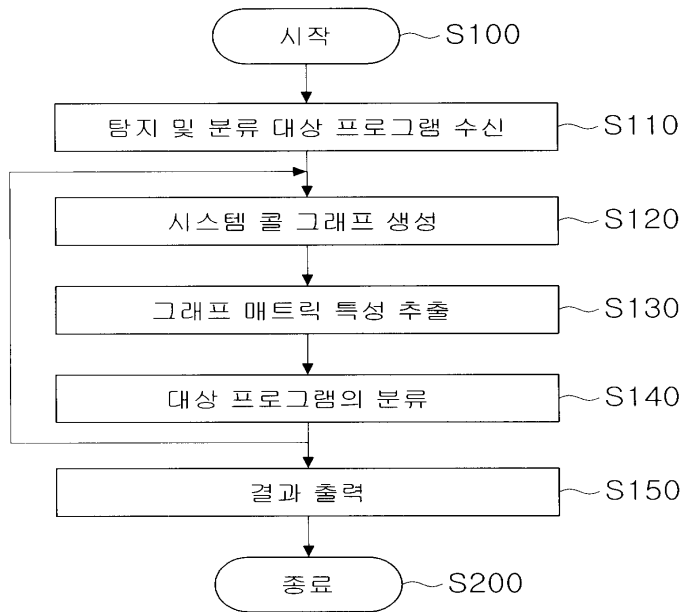
도면1



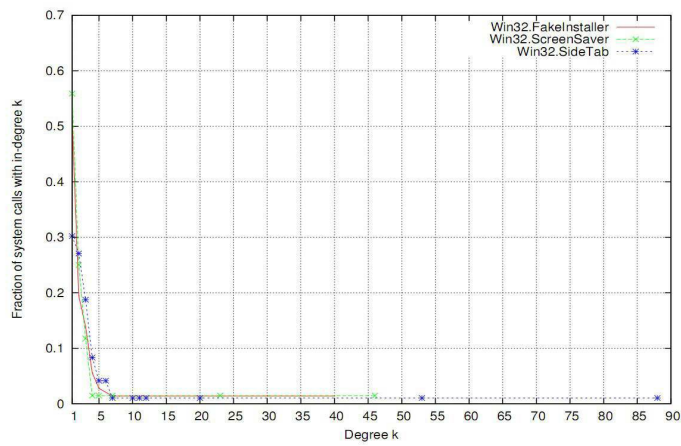
도면2



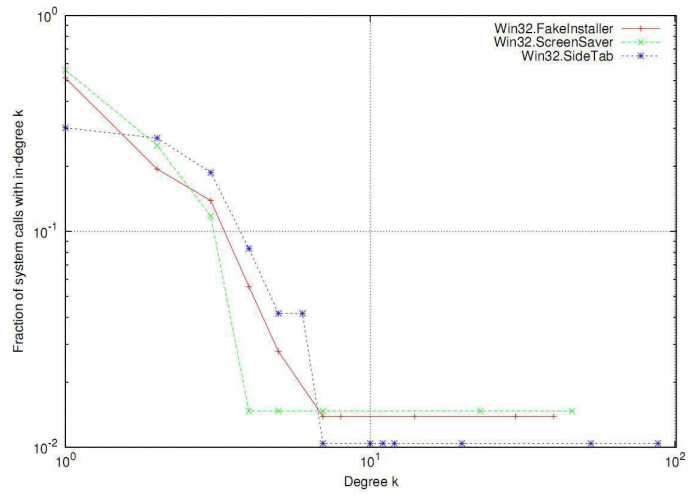
도면3



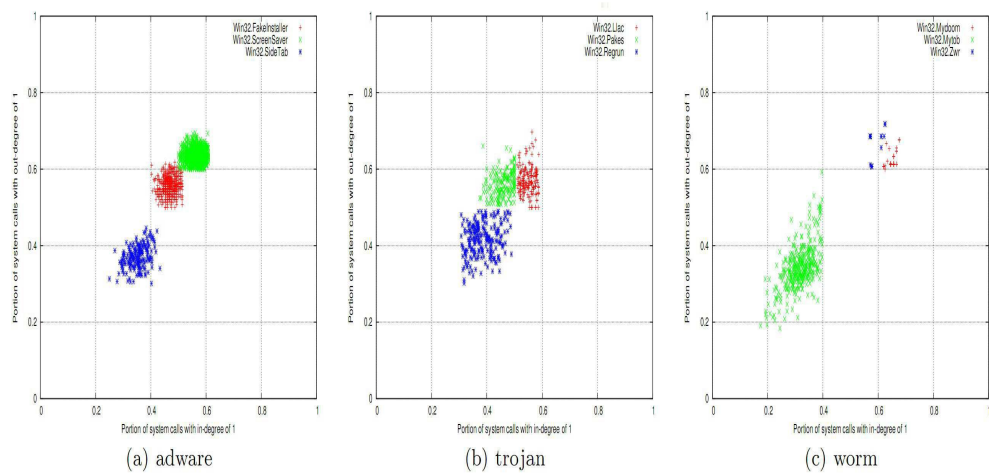
도면4



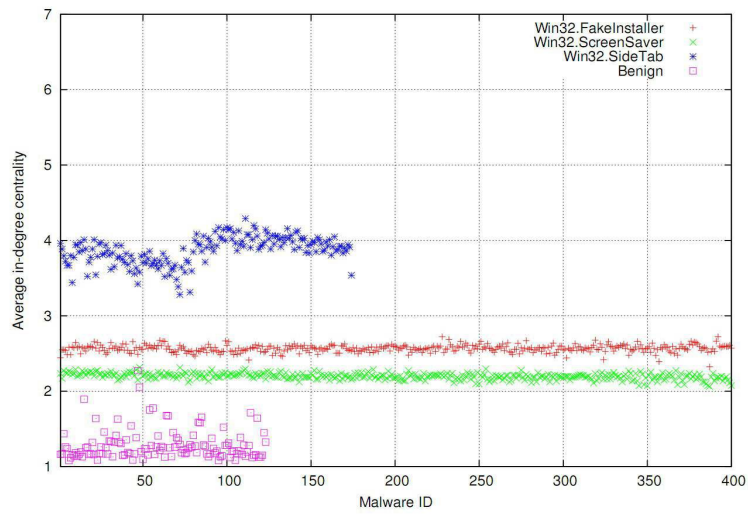
도면5



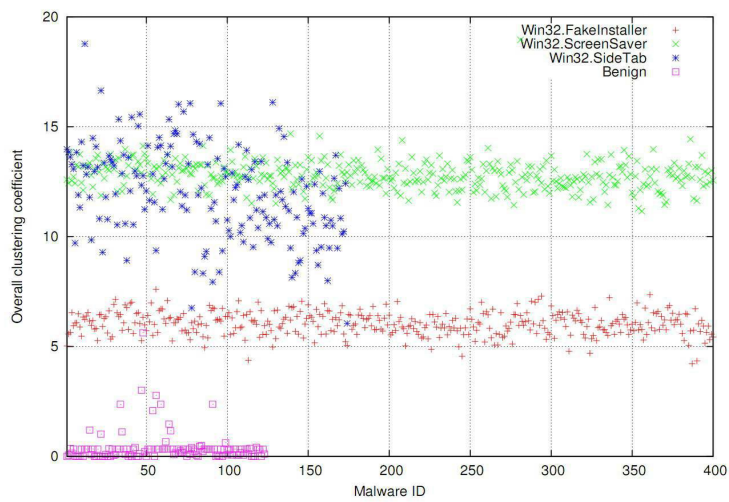
도면6



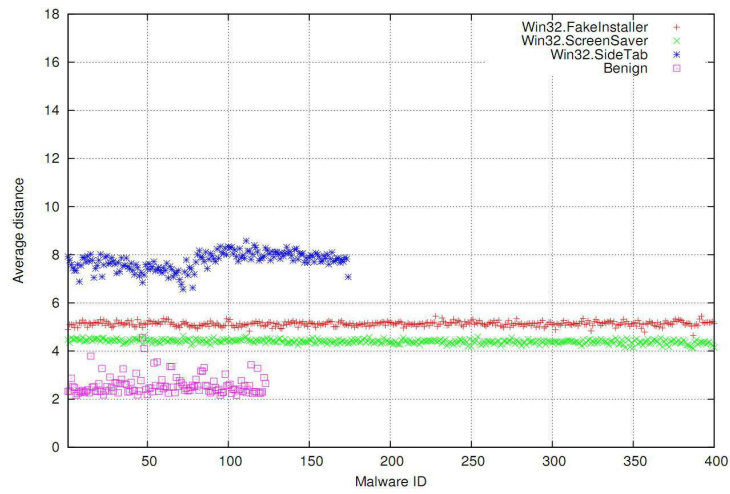
도면7



도면8



도면9



도면10

Graph metric	adware + benign	trojan + benign	worm + benign
Average in-degree centrality	0.995	0.440	0.667
Average weighted in-degree centrality	0.437	0.387	0.755
Portion of calls with in-degree of 1	0.964	0.614	0.681
Portion of calls with out-degree of 1	0.817	0.429	0.789
Average distance	0.651	0.279	0.691
Portion of calls with weighted in-degree of 1	0.257	0.309	0.610
Portion of calls with weighted out-degree of 1	0.436	0.434	0.763