기술명 : API 호출 시퀀스를 이용한 악성코드의 기능 분석 방법 및 장치

IPC : G06F 21/56

발명자 : 고려대학교 문주연

요 약
본 실시예들은 API 호출에 사용되는 매개변수를 고려하여 정의된 액션 룰을 기준으로 하위 레벨의 제1 행위 정보 를 상위 레벨의 제2 행위 정보로 추상화하고, 시퀀스에 따른 기능 룰을 기준으로 상위 레벨의 제2 행위 정보로부 터 기능 정보를 추정함으로써, 악성코드의 주요 목적 또는 핵심 기능을 예측할 수 있는 악성코드의 기능 분석 장 치를 제공한다. - 도1 (72) 발명자 김태규 경기도 성남시 분당구 판교로 333 (삼평동) 김휘강 서울특별시 성북구 안암로 145, 로봇융합관 313호 (안암동5가) 송현민 서울특별시 성북구 문주연 서울특별시 성북구 안암로 145, 로봇융합관 313호 (안암동5가) 이수연 서울특별시 성북구 안암로 145, 로봇융합관 313호 (안암동5가) 김혜민 서울특별시 성북구 안암로 145, 로봇융합관 313호 (안암동5가) 유정도 서울특별시 성북구 안암로 145, 로봇융합관 313호 (안암동5가)

청구범위
청구항 1
하나 이상의 프로세서; 및 상기 하나 이상의 프로세서에 의해 실행되는 하나 이상의 프로그램을 저장하는 메모리를 포함하며, 상기 프로세서는 악성코드를 분석하여 하위 레벨의 제1 행위 정보를 추출하고, 기 설정된 액션 룰을 기준으로 상기 하위 레벨의 제1 행위 정보를 상위 레벨의 제2 행위 정보로 추상화하고, 기 설정된 기능 룰을 기준으로 상기 상위 레벨의 제2 행위 정보로부터 기능 정보를 추정하며, 상기 하위 레벨의 제1 행위 정보는 애플리케이션 프로그래밍 인터페이스(Application Programming Interface, API) 호출 명령에 관한 제1 시퀀스 및 매개변수를 포함하고, 상기 상위 레벨의 제2 행위 정보는 상기 하위 레벨의 제1 행위 정보를 추상화한 악성코드의 기본 행위를 순서에 따라 배치한 제2 시퀀스를 포함하고, 상기 상위 레벨의 제2 행위 정보는 동일한 애플리케이션 프로그래밍 인터페이스 호출 명령을 판독하더라도 상기 매개변수에 따라 다른 기본 행위로 해석되며, 상기 프로세서가 상기 제1 행위 정보에 대해서 파일 생성(CreateFile) 함수를 호출하면서 상기 파일 생성 함수 의 매개변수로 MBR(Master Boot Record) 영역의 경로를 입력하면, 상기 제2 행위 정보를 상기 MBR 영역에 접근 하는 행위로 추상화하는 것을 특징으로 하는 악성코드의 기능 분석 장치.

청구항 2
제1항에 있어서, 상기 프로세서는 상기 악성코드를 실행하지 않고 테스트하는 정적 분석 또는 상기 악성코드를 실행하여 테스트 하는 동적 분석을 수행하여 상기 하위 레벨의 행위 정보를 추출하는 것을 특징으로 하는 악성코드의 기능 분석 장치.

청구항 3
삭제

청구항 4
삭제

청구항 5
제1항에 있어서, 상기 기능 정보는 (i) 광고, (ii) 백신 탐지, (iii) 디버깅 방해, (iv) 샌드박스 확인, (v) 권한 획득, (vi) 원격 제어, (vii) 코드 주입, (viii) 다운로드, (ix) 파일 설치, (x) 정보 탈취, (xi) 키 후킹, (xii) 마스터 부트 영역 변경, (xiii) 암호화, (xiv) 은폐, (xv) 복제, 또는 이들의 조합으로 구분되는 것을 특징으로 하는 악성코드의 기능 분석 장치.

청구항 6
제1항에 있어서, 상기 액션 룰은 상기 하위 레벨의 제1 행위 정보 및 상기 상위 레벨의 제2 행위 정보 간의 관계를 정한 것이고, 상기 기능 룰은 상기 상위 레벨의 제2 행위 정보 및 상기 기능 정보 간의 관계를 정한 것을 특징으로 하는 악성 코드의 기능 분석 장치.

청구항 7
제1항에 있어서, 상기 프로세서는 상기 액션 룰에 등록된 제1 행위 정보와 상기 추출된 제1 행위 정보 간의 유사도를 측정하고, 상기 측정한 유사도를 기준으로 상기 제2 행위 정보를 추출하는 것을 특징으로 하는 악성코드의 기능 분석 장치.

청구항 8
제1항에 있어서, 상기 프로세서는 상기 추출된 제1 행위 정보로부터 애플리케이션 프로그래밍 인터페이스 호출 명령에 관한 문자열을 추출하고, 상기 추출한 문자열을 배열한 문자열 리스트를 생성하고, 상기 애플리케이션 프로그래밍 인터페이스 호출 명령 과 숫자가 매칭된 매핑 테이블을 이용하여 상기 문자열 리스트에 속한 각 문자열에 대응하는 숫자로 변환하는 것을 특징으로 하는 악성코드의 기능 분석 장치.

청구항 9
제8항에 있어서, 상기 프로세서는 상기 변환된 숫자에 지역적 센시티브 해시 연산을 수행하여 유사도를 측정하는 것을 특징으로 하는 악성코드의 기능 분석 장치. 청구항 10 제8항에 있어서, 상기 프로세서는 상기 문자열 리스트에 대하여 해시 값으로 변환하고 두 개의 해시 값을 비트 단위(bit-wise)로 비교하여 유사도 를 측정하는 것을 특징으로 하는 악성코드의 기능 분석 장치. 청구항 11 제1항에 있어서, 상기 프로세서는 상기 추정한 제2 행위 정보 및 기 저장된 제2 행위 정보를 비교하여 상기 기능 정보를 추정하는 것을 특징으로 하는 악성코드의 기능 분석 장치. 청구항 12 프로세서에 의해 실행 가능한 컴퓨터 프로그램 명령어들을 포함하는 컴퓨터 판독 가능한 저장 매체에 기록된 컴 퓨터 프로그램으로서, 상기 컴퓨터 프로그램 명령어들이 컴퓨팅 디바이스의 적어도 하나의 프로세서에 의해 실 행되는 경우에, 악성코드를 분석하여 하위 레벨의 제1 행위 정보를 추출하는 단계; 기 설정된 액션 룰을 기준으로 상기 하위 레벨의 제1 행위 정보를 상위 레벨의 제2 행위 정보로 추상화하는 단 계; 및 기 설정된 기능 룰을 기준으로 상기 상위 레벨의 제2 행위 정보로부터 기능 정보를 추정하는 단계를 포함한 동 작들을 수행하며, 상기 하위 레벨의 제1 행위 정보는 애플리케이션 프로그래밍 인터페이스(Application Programming Interface, API) 호출 명령에 관한 제1 시퀀스 및 매개변수

를 포함하고, 상기 상위 레벨의 제2 행위 정보는 상기 하위 레벨의 제1 행위 정보를 추상화한 악성코드의 기본 행위를 순서에 따라 배치한 제2 시퀀스를 포함하고, 상기 상위 레벨의 제2 행위 정보는 동일한 애플리케이션 프로그래밍 인터페이스 호출 명령을 판독하더라도 상기 매개변수에 따라 다른 기본 행위로 해석되며, 상기 제2 행위 정보로 추상화하는 단계는, 상기 제1 행위 정보에 대해서 파일 생성(CreateFile) 함수를 호출하 면서 상기 파일 생성 함수의 매개변수로 MBR(Master Boot Record) 영역의 경로를 입력하면, 상기 제2 행위 정보 를 상기 MBR 영역에 접근하는 행위로 추상화하는 것을 특징으로 하는 컴퓨터 프로그램. 청구항 13 삭제

기 술 분 야

본 발명이 속하는 기술 분야는 악성코드의 목적 또는 핵심 기능을 예측하는 방법 및 장치에 관한 것이다.

배 경 기 술

이 부분에 기술된 내용은 단순히 본 실시예에 대한 배경 정보를 제공할 뿐 종래기술을 구성하는 것은 아니다. 정보화 기술이 발달함에 따라 네트워크 및 시스템에 대한 의존도가 증가하고 있으며 그에 따른 보안의 중요성도 커지고 있다. 사이버 공격자는 자신의 특정 목적을 달성하기 위하여 공격 대상 네트워크 및 시스템에 내재된 취 약점을 이용하여 교묘하고 지능화된 해킹기술로 공격을 행하고 있다. 이에 따라 네트워크 및 시스템에 내재된 취약성의 탐지를 위한 취약성 분석도구나 위험 분석도구와 같은 자동화 도구들이 개발되고 있으며, 사이버 공격 의 특징, 패턴, 유형 등에 대한 연구와 이에 따른 대응기술에 관한 연구가 활발히 진행되고 있다. 사이버 인프라를 이용하는 국방 등의 산업 분야에서 사이버 위협 인텔리전스(Cyber Threat Intelligence) 시스 템을 구축할 필요가 있고, 기존 악성코드로부터 위협뿐만 아니라 알려지지 않은 악성코드에 신속하게 대응하기 위해서는 악성코드가 어떤 목적을 가지고 동작하는지 파악할 필요가 있다.

해결하려는 과제

본 발명의 실시예들은 API 호출에 사용되는 매개변수를 고려하여 정의된 액션 룰을 기준으로 하위 레벨의 제1 행위 정보를 상위 레벨의 제2 행위 정보로 추상화하고, 시퀀스에 따른 기능 룰을 기준으로 상위 레벨의 제2 행 위 정보로부터 기능 정보를 추정함으로써, 악성코드의 주요 목적 또는 핵심 기능을 예측하는 데 발명의 주된 목 적이 있다. 본 발명의 명시되지 않은 또 다른 목적들은 하기의 상세한 설명 및 그 효과로부터 용이하게 추론할 수 있는 범 위 내에서 추가적으로 고려될 수 있다.

과제의 해결 수단

본 실시예의 일 측면에 의하면, 하나 이상의 프로세서, 및 상기 하나 이상의 프로세서에 의해 실행되는 하나 이 상의 프로그램을 저장하는 메모리를 포함하며, 상기 프로세서는 악성코드를 분석하여 하위 레벨의 제1 행위 정 보를 추출하고, 기 설정된 액션 룰을 기준으로 상기 하위 레벨의 제1 행위 정보를 상위 레벨의 제2 행위 정보로 추상화하고, 기 설정된 기능 룰을 기준으로 상기 상위 레벨의 제2 행위 정보로부터 기능 정보를 추정하는 것을 특징으로 하는 악성코드의 기능 추정 장치를 제공한다. 본 실시예의 다른 측면에 의하면, 프로세서에 의해 실행

가능한 컴퓨터 프로그램 명령어들을 포함하는 컴퓨터 판독 가능한 저장매체에 기록된 컴퓨터 프로그램으로서, 상기 컴퓨터 프로그램 명령어들이 컴퓨팅 디바이스의 적어도 하나의 프로세서에 의해 실행되는 경우에, 악성코드를 분석하여 하위 레벨의 제1 행위 정보를 추출하는 단계, 기 설정된 액션 룰을 기준으로 상기 하위 레벨의 제1 행위 정보를 상위 레벨의 제2 행위 정보로 추상화하 는 단계, 및 기 설정된 기능 룰을 기준으로 상기 상위 레벨의 제2 행위 정보로부터 기능 정보를 추정하는 단계 를 포함한 동작들을 수행하는 컴퓨터 프로그램을 제공한다.

발명의 효과

이상에서 설명한 바와 같이 본 발명의 실시예들에 의하면, API 호출에 사용되는 매개변수를 고려하여 정의된 액 션 룰을 기준으로 하위 레벨의 제1 행위 정보를 상위 레벨의 제2 행위 정보로 추상화하고, 시퀀스에 따른 기능 룰을 기준으로 상위 레벨의 제2 행위 정보로부터 기능 정보를 추정함으로써, 악성코드의 주요 목적 또는 핵심 기능을 예측할 수 있고, 다른 API를 호출한 변종 악성코드에 대하여 동일 기능을 수행하는지 여부를 판단할 수 있다. 여기에서 명시적으로 언급되지 않은 효과라 하더라도, 본 발명의 기술적 특징에 의해 기대되는 이하의 명세서에 서 기재된 효과 및 그 잠정적인 효과는 본 발명의 명세서에 기재된 것과 같이 취급된다.

도면의 간단한 설명

도 1 및 도 2는 본 발명의 일 실시예에 따른 악성코드의 기능 분석 장치의 동작을 예시한 도면이다. 도 3은 본 발명의 일 실시예에 따른 악성코드의 기능 분석 장치의 악성코드분석 모듈이 악성코드데이터베이스에 저장한 데이터를 예시한 도면이다. 도 4는 발명의 일 실시예에 따른 악성코드의 기능 분석 장치의 행위 추상화 모듈을 예시한 도면이다. 도 5는 발명의 일 실시예에 따른 악성코드의 기능 분석 장치의 기능 추정 모듈을 예시한 도면이다. 도 6은 발명의 일 실시예에 따른 악성코드의 기능 분석 장치가 정의한 악성코드기능을 예시한 도면이다. 도 7 내지 도 11은 발명의 일 실시예에 따른 악성코드의 기능 분석 장치가 각각의 악성코드의 기본 행위에 관하 여 정의한 제2 시퀀스를 예시한 도면이다. 도 12는 발명의 일 실시예에 따른 악성코드의 기능 분석 장치가 제1 시퀀스를 이용하여 행위의 유사도를 측정하 는 것을 예시한 도면이다. 도 13은 예시적인 실시예들에서 사용되기에 적합한 컴퓨팅 디바이스를 포함하는 컴퓨팅 환경을 예시하여 설명하 기 위한 블록도이다.

발명을 실시하기 위한 구체적인 내용

이하, 본 발명을 설명함에 있어서 관련된 공지기능에 대하여 이 분야의 기술자에게 자명한 사항으로서 본 발명 의 요지를 불필요하게 흐릴 수 있다고 판단되는 경우에는 그 상세한 설명을 생략하고, 본 발명의 일부 실시예들 을 예시적인 도면을 통해 상세하게 설명한다. 기존의 악성코드분석 방법은 개별 API(Application Programming Interface) 호출 여부만을 확인하여 악성코드 의 단편적인 행위 정보만을 이용한다. API 호출 정보만을 이용하는 경우 해당 API를 호출하는 정상 프로그램에 대해서 오탐을 발생시킬 위험이 높고, 동일한 행위를 수행하지만 다른 API를 호출하는 경우에 동일한 행위 정보 로 판단하기 어려운 문제가 있다. 본 실시예들은 악성코드의 행위 정보를 기반으로 악성코드의 최종 목적인 핵심 기능을 파악하여 악성코드의 동 작을 파악하고 효과적으로 악성코드에 대응할 수 있도록 정보를 제공한다. 본 실시

예들은 1차적으로 API 호출시 사용되는 매개변수 정보를 고려해 악성코드의 API 호출 데이터(제1 행위 정보)를 기본 행위 정보로 추상화하고, 추상화된 기본 행위 정보(제2 행위 정보)의 시퀀스를 이용하여 악성코드 의 주요 기능을 최종 판단한다. 본 실시예들은 악성코드의 주요 행위들에 대해 행위 별로 기본 행위를 정의하여, 다른 API를 호출하더라도 그 목적과 의미를 분석하여 동일한 기본 행위로 추상화할 수 있으며, 이를 통해 유사한 API 호출을 이용하는 변종 악성코드에 대해서도 동일 기능을 수행하는 것으로 판단할 수 있다 본 실시예들은 API의 호출 여부뿐만 아니라 API 호출에 쓰인 매개변수 정보를 이용해 해당 호출의 목적 및 의미를 담은 기본 행위 정보로 추상화한다. 예컨대, 동일한 파일 생성(create file) 행위라도 인터넷 관련 레지스트 리키 파일 생성과 exe 실행 파일 등을 구분한다. 기본 행위 정보 사이의 유기적 관계(시퀀스)를 이용하여 악성 코드의 기능을 분석한다. 본 명세서에서 사용된 용어를 정의하면 다음과 같다. API 호출 명령은 악성코드가 실행 중 호출하는 Windows API의 함수명과 호출시 사용되는 매개변수, 예컨대, 디 렉토리 또는 파일의 경로, 레지스트리, 프로세스명 등의 연관 정보를 포함한다. API 호출 시퀀스는 악성코드가 호출한 API들을 호출된 순서로 나열한 데이터이다. 순서는 시간의 흐름에 따르며 동시 또는 이시에 호출될 수 있고, 순차적 처리 또는 병렬적 처리를 포함할 수 있다. 기본 행위 정보는 API 호출 명령을 분석하여 특정 악성 행위를 나타내는 행위를 추상화한 정보이다. 액션 룰은 API 호출 명령을 악성 행위로 추상화시키기 위해 정의한 룰로, 악성코드의 API 호출 정보가 정의된 룰과 일치하면 해당하는 행위를 수행하는 것으로 판단 기준을 제공한다. 기본 행위 시퀀스는 액션 룰에 의해 추출된 기본 행위들을 시간 순으로 나열한 데이터, 이때 시간은 액션이 탐 지된 API 호출 정보의 시간을 의미할 수 있다. 기능 룰은 악성코드가 특정 기능을 포함하고 있는 경우 나타나는 기본 행위 시퀀스를 정의한 룰로, 액션 룰을 통해 추출된 기본 행위 시퀀스가 정의된 룰과 일치하면 해당하는 기능을 포함하는 것으로 판단 기준을 제공한다. 도 1 및 도 2는 본 발명의 일 실시예에 따른 악성코드의 기능 분석 장치의 동작을 예시한 도면이다. 악성코드의 기능 분석 장치는 하나 이상의 프로세서 및 하나 이상의 프로세서에 의해 실행되는 하나 이상의 프 로그램을 저장하는 메모리를 포함할 수 있다. 프로세서는 악성코드분석 모듈(210), 행위 추상화 모듈(220), 기 능 추정 모듈(230)의 동작을 제어할 수 있다. 도 1을 참조하면, 단계 S110에서 프로세서는 악성코드를 분석하여 하위 레벨의 제1 행위 정보를 추출한다. 단계 S120에서 프로세서는 기 설정된 액션 룰을 기준으로 상기 하위 레벨의 제1 행위 정보를 상위 레벨의 제2 행위 정보로 추상화한다. 단계 S130에서 프로세서는 기 설정된 기능 룰을 기준으로 상기 상위 레벨의 제2 행위 정보 로부터 기능 정보를 추정한다. 하위 레벨의 제1 행위 정보는 애플리케이션 프로그래밍 인터페이스(Application Programming Interface, API) 호출 명령에 관한 제1 시퀀스 및 매개변수를 포함한다. 상위 레벨의 제2 행위 정보는 하위 레벨의 제2 행위 정보를 추상화한 악성코드의 기본 행위를 순서에 따라 배치 한 제2 시퀀스를 포함한다. 상위 레벨의 제2 행위 정보는 동일한 애플리케이션 프로그래밍 인터페이스 호출 명 령을 판독하더라도 매개변수에 따라 다른 기본 행위로 해석될 수 있다. 악성코드분석 모듈(210)은 입력 받은 악성코드샘플에 대하여 정적 분석 및/또는 동적 분석을 수행하여 행위 정 보(윈도우 API 호출 정보 및 매개변수)를 포함한 다양한 악성코드 정보를 추출한다. 행위 추상화 모듈(220)은 API 함수 레벨의 행위 정보(윈도우 API 및 매개변수)를 악성코드의 기본 동작 단위인 기본 행위 정보로 추상화한다. 행위 추상화 모듈(220)은 악성코드의 행위 정보(윈도우 API 및 매개변수)를 이용 하여 데이터베이스에 저장된 액션 룰을 통해 악성코드가 어떤 동작을 수행하는지를 나타내는 기본 행위 정보로 추상화한다. 추출된 행위(디버거 탐지, 크립토 키 생성, 파일 이

동, 네트워크 연결, 시스템 정보 수집 등) 정보 와 알려진 악성코드의 기능 정보를 이용하여 각 기능을 탐지할 수 있는 기본 행위 시퀀스(제2 시퀀스)를 정의하 여 악성코드데이터베이스에 저장한다. 기능 추정 모듈(230)은 기본 행위 정보를 이용하여 악성코드의 주요 목표인 기능을 탐지한다. 기능 추정 모듈 (230)은 추출된 기본 행위 정보를 이용하여 데이터베이스에 기능 별로 저장된 기본 행위 시퀀스와 비교를 통해 기본 행위 시퀀스가 일치하는 기능이 있는지 확인한다. 도 3은 본 발명의 일 실시예에 따른 악성코드의 기능 분석 장치의 악성코드분석 모듈이 악성코드데이터베이스에 저장한 데이터를 예시한 도면이다. 프로세서 또는 악성코드분석 모듈(210)은 악성코드를 실행하지 않고 테스트하는 정적 분석 또는 악성코드를 실 행하여 테스트하는 동적 분석을 수행하여 하위 레벨의 행위 정보를 추출한다. 악성코드의 API 호출 정보(API 이 름, 매개변수, 디렉토리, 파일, 레지스트리)를 추출하여 악성코드데이터베이스에 저장한다. 도 3은 API 정보를 추출한 예시이다. 하나의 API당 하나의 행으로 도 3과 같이 저장된다. API 외에 파일과 레지 스트리 등의 정보도 각각 하나의 행으로 함께 저장된다. 도 4는 발명의 일 실시예에 따른 악성코드의 기능 분석 장치의 행위 추상화 모듈을 예시한 도면이다. 프로세서 또는 행위 추상화 모듈(220)은 액션 룰을 기준으로 하위 레벨의 제1 행위 정보를 상위 레벨의 제2 행 위 정보로 추상화한다. 액션 룰은 하위 레벨의 제1 행위 정보 및 상위 레벨의 제2 행위 정보 간의 관계를 정한 것이다. 프로세서 또는 행위 추상화 모듈(220)은 악성코드분석 모듈(210)에서 추출한 악성코드의 하위 레벨 행위 정보인 API 호출 정보를 데이터베이스에 저장된 액션 룰(220)과 매칭시켜 상위 레벨인 제2 행위 정보로 추상화한다. 제2 행위 정보 또는 기본 행위 정보는 '파일 이동', '프로세스 탐색', 'MBR영역에 접근' 등의 악성코드의 기본 적인 행위를 의미한다. 이 때, 악성코드의 행위는 악성코드가 동일한 API를 호출하더라도 매개변수에 따라 다른 액션으로 해석되어야 한다. 예를 들어, CreateFile API 함수는 일반적으로 매개변수로 파일의 경로가 같이 입력 되어 특정 경로에 파일을 생성하는데 사용된다. 반면, MBR(Master Boot Record) 영역의 경로를 입력하게 되면 파일을 여는 행위가 아니라 MBR 영역에 접근을 하는 행위로 해석되어야 한다. 액션 룰은 악성코드가 호출한 API와 그 매개변수를 함께 고려하여 악성코드가 어떤 행위를 수행하는 것인지 추 상화하기 위한 룰을 정의한 것이다. 악성코드의 특정 오브젝트에 대한 접근, 변경, 호출 여부를 바탕으로 생성하며, Windows API, 매개변수, 디렉토리, 파일, 레지스트리 등이 포함될 수 있다. 예컨대, 어떤 악성코드가 Windows API인 CreateFile을 호출하면서 해당 API의 매개변수로 '\\.\PhysicalDrive0'을 사용한다면 'MBR 영역에 접근' 행위로 정의한다. 도 3을 도시된 바와 같이, 두 번째 행의 CreateFile은 MBR 영역을 나타내는 경로('\\.\PhysicalDrive0')를 매개변수로 사용하므로 'MBR 영역에 접근' 행위로 추상화할 수 있다. 하위 레벨 행위 정보인 API 시퀀스(제1 시퀀스)를 상위 레벨 행위 정보인 기본 행위 정보로 추상화하면 기본 행 위 정보로 구성된 시퀀스(제2 시퀀스)가 만들어진다. 모든 API호출이 각각의 행위로 매칭되는 것은 아니므로, API 시퀀스의 길이와 액션 룰에 따라서 악성코드에서 발견되는 액션 시퀀스의 길이는 달라지게 된다. API 호출 순서에 따라 매칭되는 액션 정보를 시퀀스로 나타내고, 이를 기본 행위 시퀀스라 정의한다. 도 5는 발명의 일 실시예에 따른 악성코드의 기능 분석 장치의 기능 추정 모듈을 예시한 도면이다. 프로세서 또는 기능 추정 모듈(230)은 기 설정된 기능 룰을 기준으로 상위 레벨의 제2 행위 정보로부터 기능 정 보를 추정한다. 기능 룰은 상위 레벨의 제2 행위 정보 및 기능 정보 간의 관계를 정한 것이다. 프로세서는 추정 한 제2 행위 정보 및 기 저장된 제2 행위 정보를 비교하여 기능 정보를 추정한다. 기능 추정 모듈(230)은 기본 행위 시퀀스와 기능 룰을 매칭한다. 기능 룰은 '기본 행위 1->기본 행위 2->기본 행위 3' 차

례로 이어지면 기능 1을 가짐과 같은 형태로 정의되어 있다. 기본 룰에 관하여는 도 7 내지 도 11에 상세히 도시되어 있다. 도 6은 발명의 일 실시예에 따른 악성코드의 기능 분석 장치가 정의한 악성코드기능을 예시한 도면이다. 기능 정보는 (i) 광고, (ii) 백신 탐지, (iii) 디버깅 방해, (iv) 샌드박스 확인, (v) 권한 획득, (vi) 원격 제어, (vii) 코드 주입, (viii) 다운로드, (ix) 파일 설치, (x) 정보 탈취, (xi) 키 후킹, (xii) 마스터 부트 영역 변경, (xiii) 암호화, (xiv) 은폐, (xv) 복제, 또는 이들의 조합으로 구분될 수 있다. (i) 광고(Adware)는 이용자의 동의 없이 광고를 띄우는 기능이고, (ii) 백신 탐지(Anti-av)는 백신 프로그램을 탐지하여 우회, 종료시키는 기능이고, (iii) 디버깅 방해(Anti-dbg)는 디버깅을 방해하는 기능이고, (iv) 샌드 박스 확인(Anti-sandbox)은 실행 환경이 샌드박스인지 확인하는 기능이고, (v) 권한 획득(Backdoor)은 일반적인 인증 과정을 거치지 않고 권한을 획득하는 기능이고, (vi) 원격 제어(Botnet)는 감염 대상을 봇(Bot)으로 만들 어 네트워크를 통해 원격 접근이나 통제가 가능하게 만드는 기능이고, (vii) 코드 주입(Code-injection)은 프로 세스에 악의적 코드를 주입하는 기능이고, (viii) 다운로드(Downloader)는 외부로부터 악성코드를 다운로드하여 대상을 감염시키는 기능이고, (ix) 파일 설치(Dropper)는 파일을 설치 및 실행하는 기능이고, (x) 정보 탈취(Infostealer)는 인터넷 브라우저, ftp 클라이언트, 메일 등에 저장된 개인 정보를 탈취하는 기능이고, (xi) 키 후킹(Keylogger)은 키보드 입력 이벤트를 후킹하여 기록하는 기능이고, (xii) 마스터 부트 영역 변경(MBR- destroyer)은 하드디스크의 Master Boot Record 영역을 비정상적인 값으로 변경하는 기능이고, (xiii) 암호화 (Ransomware)는 사용자의 파일을 암호화하는 기능이고, (xiv) 은폐(Stealth)는 스스로를 은폐시키는 기능이고, (xv) 복제(Worm)는 스스로를 복제하여 네트워크 상에 전파시켜 다른 기기를 감염시키는 기능을 의미한다. 도 7 내지 도 11은 발명의 일 실시예에 따른 악성코드의 기능 분석 장치가 각각의 악성코드의 기본 행위에 관하 여 정의한 제2 시퀀스를 예시한 도면이다. 도 7의 (a)에 도시된 광고(Adware) 기능을 참조하면, 광고(Adware) 기능의 기본 행위 시퀀스는 설치 이후 웹 브 라우저를 열기 위한 행위 수행 후 정보 탈취나 브라우저 설정 변경을 진행한다. 먼저 현재 '실행 중인 프로세스 탐색' 행위를 수행한다. 그 이후에는 '프로세스에 접근 시도' 행위를 수행한다. 이후 '브라우저 실행' 액션으로 브라우저를 성공적으로 실행하고, '정보 탈취'나 '브라우저 설정 변경' 행위를 수행한다. 광고(Adware) 기능에 해당하는 행위의 액션 룰은 다음과 같다. - 실행 중인 프로세스 탐색 행위는 Process32FirstW와 Process32NextW API 콜이 존재한다. - 프로세스에 접근 시도 행위는 RegOpenKeyExA나 NtOpenProcess API 콜이 존재한다. - 브라우저 실행 행위는 레지스트리를 읽어서 웹브라우저(예컨대, FireFox, Chrome 웹 브라우저)를 실행했는지 확인한다. - 브라우저 설정 변경 행위는 NtWriteFile로 브라우저의 프로필 관련 파일을 직접 새로 생성하거나 브라우저 내 통신관련 설정 (예컨대, SPDY 통신, HTTP 2.0 통신과 관련된 설정)을 작성한다. 도 7의 (b)에 도시된 백신 탐지(Anti-av) 기능을 참조하면, 백신 탐지(Anti-av) 기능의 기본 행위 시퀀스는 '백 신 프로그램 탐지' 행위를 수행한 후 크게 두 가지의 행위를 수행하는데, 백신 프로그램을 우회하는 시도를 하 거나 백신 프로그램을 종료시키는 '탐지 우회 시도' 행위와 '백신 프로그램 종료' 행위를 수행한다. 백신 탐지(Anti-av) 기능에 해당하는 행위의 액션 룰은 다음과 같다. - 백신 프로그램 탐지 행위는 레지스트리키나 설치된 파일 정보를 이용하여 알려진 AV의 설치 여부를 확인한다. - 탐지 우회 시도 행위는 CreateProcessInternalW나 ShellExecuteExW를 이용해 자신의 프로세스 명을 널리 알 려진 프로세스명으로 변경한다. - 백신 프로그램 종료 행위는 ControlService나 NtTerminateProcess로 AV 프로세스 및 서비스를 종료 시킨다. 도 7의 (c)에 도시된 디버깅

방해(Anti-dbg) 기능을 참조하면, 디버깅 방해(Anti-dbg) 기능의 기본 행위 시퀀스 는 먼저 디버깅 확인을 하기 위해 '디버거 체크' 행위를 수행한다. 만약 실행 중인 디버거가 탐지되면 디버깅을 방해하기 위해 디버거를 종료 시키거나 디버거 관련 파일을 삭제하는 '디버깅 방해' 행위를 수행한다. 디버깅 방해(Anti-dbg) 기능에 해당하는 행위의 액션 룰은 다음과 같다. - 디버거 체크 행위는 CheckRemoteDebuggerPresent와 IsDebuggerPresent, SystemKernelDebuggerInformation API 콜이 존재한다. - 디버깅 방해 행위는 NtTerminateProcess와 DeleteService, DeleteFileW API 콜이 존재한다. 도 8의 (a)는 도시된 샌드박스 확인(Anti-sandbox) 기능을 참조하면, 샌드박스 확인(Anti-sandbox) 기능의 기본 행위 시퀀스는 특정한 순서가 존재하지 않으며, 다음 네 가지의 행위가 있다. (i) 특정 샌드박스 환경에만 존재하는 디렉토리를 검색하는 경우 '샌드박스 관련 파일 검색' 행위를 수행한다. (ii) 사용자가 직접 창을 사용하는지 탐지하기 위해 '포그라운드 창 검사' 행위를 주기적으로 수행한다. (iii) 샌드박스 환경에서 발생할 수 있는 약간의 idle 시간을 탐지하기 위해 'idle 시간 검사' 행위를 주기적으로 수 행한다. (iv) 샌드박스는 정해진 시간 동안 실행된다는 것을 이용하여 분석을 방해하기 위해 '프로세스 지연' 행위를 수행한다. 샌드박스 확인(Anti-sandbox) 기능에 해당하는 행위의 액션 룰은 다음과 같다. - 샌드박스 관련 파일 검색 행위는 샌드박스 환경에만 존재하는 파일의 유무를 검색한다. - 포그라운드 창 검사 행위는 GetForegroundWindow와 NtDelayExecution API 콜이 존재한다. - Idle 시간 검사 행위는 NtQuerySystemInformation API 콜이 존재하며 이 함수의 첫 번째 인자로 SystemProcessofPerformanceInformation을 전달한다. - 프로세스 지연 행위는 NtDelayExecution API 콜이 존재한다. 도 8의 (b)에 도시된 권한 획득(Backdoor) 기능을 참조하면, 권한 획득(Backdoor) 기능의 기본 행위 시퀀스는 먼저 '네트워크 활동' 행위를 수행한다. 네트워크 활동은 네트워크 패킷이 교환되는 것을 통해 알 수 있다. 이 후 '악성 프로세스 작동' 행위를 수행한다. 권한 획득(Backdoor) 기능에 해당하는 행위의 액션 룰은 다음과 같다. - 네트워크 활동 개시 행위는 네트워크 활동이 탐지된다(네트워크 패킷 송수신 행동, ICMP 서버와의 네트워크 연결 등). - 악성 프로세스 작동 행위는 NtAllocateVirtualMemory나 NtProtectVirtualMemory, WriteProcessMemory, NtMapViewOfSection, NtUnmapViewOfSection, VirtualProtectEx, Process32NextW, Process32FirstW API 콜이 존재한다. 도 8의 (c)에 도시된 원격 제어(Botnet) 기능을 참조하면, 원격 제어(Botnet) 기능의 기본 행위 시퀀스는 '시스 템 정보 수집' 행위를 수행한다. 이후 시스템 정보를 CNC(Command & Control) 서버로 전송하기 위해서 CNC 서버 와 네트워크 연결을 수행하는데, 이는 'CNC 네트워크 연결' 행위로 확인할 수 있다. 이후 멀티 프로세스를 이용 하기 위해 새로운 프로세스를 생성하는 '프로세스 생성' 행위를 수행하고, 생성된 프로세스를 통해 ' 파일 드롭' 행위를 수행한다. 이후 자동 실행을 등록하기 위해서 레지스트리에 자동 실행 프로그램 키를 생성하거나 자동 실행 프로세스를 생성하기도 한다. 이 경우에는 '자동 실행'행위가 기록되기도 한다. 원격 제어(Botnet) 기능에 해당하는 행위의 액션 룰은 다음과 같다. - 시스템 정보 수집 행위는 GetSystemInfo나 RegQueryValueEx API 콜이 존재한다. - CNC 네트워크 연결 행위는 IRC(Internet Relay Chat) 서버나 ICMP(Internet Control Message Protocol) 서버 에 네트워크 연결을 확인한다. - 프로세스 생성 행위는 CreateThread API 콜이 존재함 또는 wbemServices_ExecMethod 또는 IWbemServices_ExecMethodSync API 콜이 존재하며 Win32_Process 프로세스에 접근한다. - 파일 드롭 행위는 응용프로그램 관련 폴더에 확장자가 exe인 파일 또는 실행 가능한 바이너리 파일을 드롭한 다. - 자동 실

행 행위는 레지스트리에 자동 실행 프로그램 키를 생성한다. 도 9의 (a)에 도시된 코드 주입(Code-injection) 기능을 참조하면, 코드 주입(Code-injection) 기능의 기본 행 위 시퀀스는 '파일에 악성코드 주입' 행위를 수행하거나 '프로세스에 악성코드 주입' 행위를 수행한다. 실행 중 인 파일이나 기존 파일에 악성코드를 주입하여 공격자가 원하는 행동을 취할 수 있도록 '파일에 악성코드 주입' 행위를 수행한다. 이와 동일하게 실행 중인 프로세스에 악성코드를 주입하여 곧바로 악성 행위를 실행하고 Non- child 프로세스를 생성하여 원격접근 권한을 얻기 위해 '프로세스에 악성코드 주입' 행위를 수행한다. 혹은 프 로세스의 메모리를 수정하거나 변경하는 과정을 통해 악성코드를 주입한다. 이 경우에는 '메모리 변경' 액션이 기록되기도 한다. 코드 주입(Code-injection) 기능에 해당하는 행위의 액션 룰은 다음과 같다. - 파일에 악성코드 주입 행위는 파일의 메모리와 가상메모리 공간을 수정하는 행위가 나타날 경우 액션이 탐지 된다. - 프로세스에 악성코드 주입 행위는 NtSetContextThread API 콜을 이용하여 프로세스에 대해 원격 접근 권한을 얻는다. - 메모리 변경 행위는 가상메모리 공간에 다른 프로세스의 메모리를 덮어쓰는 행위이다. 도 9의 (b)에 도시된 다운로드(Downloader) 기능을 참조하면, 다운로드(Downloader) 기능의 기본 행위 시퀀스는 먼저 '네트워크 활동 개시' 행위로 네트워크 활동을 시작한다. 송수신한 패킷을 확인하는 등으로 네트워크 활동 을 탐지할 수 있다. 다음으로 다른 장치로부터 악성코드를 다운로드하는 '다운로드' 행위를 수행한다. 다른 장치로부터 네트워크로 패킷을 수신하는 행위이다. 마지막으로 악성코드 실행 API를 이용하여 '악성코드 실행' 행 위를 수행한다. 다운로드(Downloader) 기능에 해당하는 행위의 액션 룰은 다음과 같다. - 네트워크 활동 개시 행위는 네트워크 활동이 탐지된다(네트워크 패킷 송수신 행동, ICMP 서버와의 네트워크 연결 등). - 다운로드 행위는 Recv, RecvFrom, recv, recvFrom 등 패킷을 수신하는 API 콜이 존재한다. - 악성코드 실행 행위는 ShellExecuteExW API 콜이 존재한다. 도 9의 (c)에 도시된 파일 설치(Dropper) 기능을 참조하면, 파일 설치(Dropper) 기능의 기본 행위 시퀀스는 '바 이너리 파일 생성' 행위를 수행하여 실행될 수 있는 바이너리 파일을 생성한다. 이 후 '실행 파일 설치' 행위를 수행하여 exe 파일을 설치 및 실행한다. 파일 설치(Dropper) 기능에 해당하는 행위의 액션 룰은 다음과 같다. - 바이너리 파일 생성 행위는 CreateProcessInternalW와 ShellExecuteExW를 이용해 바이너리 파일을 생성한 후 실행한다. - 실행 파일 설치 행위는 응용프로그램 관련 폴더(예컨대, MyApp)에 확장자가 exe인 파일 또는 실행 가능한 바 이너리 파일을 드롭한다. 도 10의 (a)에 도시된 정보 탈취(Infostealer) 기능을 참조하면, 정보 탈취(Infostealer) 기능의 기본 행위 시 퀀스는 정보 탈취 이전에 '인터넷 브라우저 저장 경로 검색' 행위를 수행하거나 '권한 체크' 액션, '브라우저 보안 설정 변경' 행위를 수행한다. 정보 탈취(Infostealer) 기능에 해당하는 행위의 액션 룰은 다음과 같다. - 인터넷 브라우저 저장 경로 검색 행위는 인터넷 브라우저(예: Chrome, FireFox 웹브라우저)의 레지스트리와 디렉토리를 검색하여 브라우저의 위치를 파악한다. - 권한 체크 행위는 LookupPrivilegeValue API 콜이 존재한다. - 브라우저 보안 설정 변경 행위는 브라우저 관련 레지스트리 값을 변경한다. - 개인정보 탈취 행위는 인터넷 브라우저(예컨대, Chrome, FireFox), FTP 클라이언트(예컨대, FTP Explorer), 메일 클라이언트(예컨대, Outlook Express)가 사용하는 레지스트리와 파일에 접근한다. 도 10의 (b)에 키 후킹(Keylogger) 기능을 참조하면, 광고(Adware) 기능의 기본 행위 시퀀스는 '키보드/마우스 후킹' 행위를 수행한다. 키보드 입력값뿐만 아니라 마우스 이벤트 모니터링을 위한 후킹으로 순서 상관 없이 두 가지 행위를 진행한다. 이후 후킹한 정보를 바로 C&C 서버에 송신하기도 하는데 이 경우에는 '정보 송신' 행위 를 기록하기도 한다. 키 후킹(Keylogger) 기능에

해당하는 행위의 액션 룰은 다음과 같다. - 키보드 후킹 행위는 SetWindowsHookExA, SetWindowsHookExW 콜에서 WH_KEYBOARD_LL 파라미터를 사용하여 키 보드 입력값을 후킹한다. - 마우스 후킹 행위는 SetWindowsHookExA, SetWindowsHookExW 콜에서 WH_MOUSE_LL 파라미터를 사용하여 마우스 입력값을 후킹한다. - 정보 송신 행위는 C&C 서버로 의심되는 IP 주소나 불특정 다수의 IP 주소와 네트워크 트래픽을 발생시킨다. 도 10의 (c)에 도시된 마스터 부트 영역 변경(MBR-destroyer) 기능을 참조하면, 마스터 부트 영역 변경(MBR- destroyer) 기능의 기본 행위 시퀀스는 먼저 MBR 영역을 여는 'MBR 영역 열기' 행위를 수행한 후 'MBR 영역 변 경' 행위를 수행하여 MBR 영역을 파괴한다. 마스터 부트 영역 변경(MBR-destroyer) 기능에 해당하는 행위의 액션 룰은 다음과 같다. - MBR 영역 열기 행위는 CreateFile API 콜이 존재하며, CreateFile의 매개변수로 전달하는 파일명이 '\\.\PhysicalDrive0'에서 '\\.\PhysicalDrive25'까지에 해당한다. - MBR 영역 변경 행위는 WriteFile API 콜이 존재하며, CreateFile의 파일 핸들값을 매개변수로 가진다. 도 11의 (a)에 도시된 암호화(Ransomware) 기능을 참조하면, 암호화(Ransomware) 기능의 기본 행위 시퀀스는 ' 목표 파일 탐색' 행위를 수행하여 해당 디렉토리에 존재하는 타겟 파일을 모두 탐색한다. 이후 파일을 한 디렉 토리에 이동하는 '파일 이동' 행위를 수행한다. 디렉토리에 모인 파일을 일괄적으로 암호화하기 위해서 '파일 암호화' 행위를 수행하며 기존의 파일은 삭제하고 암호화가 된 새로운 파일을 생성하는 '파일 생성' 행위를 수 행한다. 암호화(Ransomware) 기능에 해당하는 행위의 액션 룰은 다음과 같다. - 목표 파일 탐색 행위는 FindNextFile, FindFirstFileEx API 콜이 존재한다. - 파일 이동 행위는 MoveFileWithProgress API 콜을 통해 목표 파일을 이동시킨다. - 파일 암호화 행위는 SetFileAttributes API 콜을 통해 파일 속성을 ENCRYPTED로 변경한다. - 파일 생성 행위는 CreateFile, CopyFile API 콜을 통해 새로운 확장자명을 가진 파일을 생성한다. 도 11의 (b)에 도시된 은폐(Stealth) 기능을 참조하면, 은폐(Stealth) 기능의 기본 행위 시퀀스는 '백그라운드 프로세스 실행' 행위를 수행하거나 '프로세스 위장' 행위를 수행한다. 정상 사용자의 눈에 보이지 않는 프로세 스를 생성하기 위해서 '백그라운드 프로세스 실행' 행위를 수행하여 악성 행위 실행을 숨긴다. 악성 행위를 수행하는 프로세스를 정상 프로세스로 위장하여 정상 사용자나 안티바이러스 프로그램으로부터 악성 행위가 실행 됨을 인지하지 못하게 '프로세스 위장' 행위를 수행한다. 은폐(Stealth) 기능에 해당하는 행위의 액션 룰은 다음과 같다. - 백그라운드 프로세스 행위 ShellExecuteExW API 콜의 인자 값을 hidden으로 바꾸는 경우와 CreateProcessInternalW API 콜의 인자 값 CREATE_NO_WINDOW로 바꾸는 경우가 존재한다. - 프로세스 위장 행위는 CreateProcessInternalW API 콜과 ShellExecuteExW API 콜을 통하여 흔한 프로세스의 이름으로 자신의 프로세스 이름을 변경한다. 도 11의 (c)에 도시된 복제(Worm) 기능을 참조하면, 복제(Worm) 기능의 기본 행위 시퀀스는 먼저 윈도우가 부팅 되면 '자동 시작' 행위를 수행한다. 이후 악성 행위를 담은 파일을 다른 장치로 전송하기 전에 알려진 패커를 사용해 ' 패킹' 행위를 수행한다. 이후 일부 악성코드에서는 패킷 전송 이전에 추가적인 감염 대상을 찾기 위해 ICMP 서버에 연결하여 '주변 장치 탐색' 행위를 수행한다. 마지막으로 포트를 열어 네트워크 패킷을 전송하는 '패킷 전송' 행위를 수행한다. 이 때 주로 445번 포트가 사용된다. 복제(Worm) 기능에 해당하는 행위의 액션 룰은 다음과 같다. - 자동 시작 행위는 CreateServiceA, CreateServiceW가 존재하고 레지스트리에 자동 실행 프로그램 키를 생성한 다. - 패킹 행위는 알려진 패커를 수행한다. - 주변 장치 탐색 행위는 ICMP 서버에 네트워크 연결을 확인한다. - 패킷 전송 행위는 네트워크 패킷을 전송한다(445번 포트가 자주 사

용됨). 도 12는 발명의 일 실시예에 따른 악성코드의 기능 분석 장치가 제1 시퀀스를 이용하여 행위의 유사도를 측정하 는 것을 예시한 도면이다. 악성코드의 윈도우 API 호출 정보는 악성코드의 행위 정보를 담고 있으며, 유사한 행위를 하는 악성코드는 유사 한 API 호출 시퀀스를 생성한다. API 호출 시퀀스 유사도를 측정하기 위해서는 비슷한 시퀀스 사이의 유사도가 높게 나와야 한다는 기본 조건 외 에 두 가지 조건을 만족하는 알고리즘을 사용해야 한다. 첫 번째 조건은 서로 길이가 다른 시퀀스의 유사도도 길이의 차이와 관계없이 측정할 수 있어야 한다. 두 번째 조건은 길이가 긴 시퀀스에 대한 유사도 계산이 빠르게 진행되어야 한다. 시퀀스 유사도 알고리즘은 다양하지만 대부분 비교 대상 시퀀스 사이의 길이 차이가 유사도 측정 결과에 많은 영향을 끼친다. 두 가지 조건을 만족하 는 시퀀스 유사도 측정 알고리즘으로 지역적 센시티브 해시(Locality Sensitive Hashing, LSH) 알고리즘을 적용 할 수 있다. LSH 알고리즘은 일반적인 암호학적 해시와 달리 충돌 확률을 최대화하여 유사한 입력 값에 대해 유사한 해시를 생성하도록 설계되었다. 서로 길이가 다른 시퀀스를 일정한 길이의 해시값으로 만들어주기 때문에 생성된 해시 값 사이의 유사도를 측정하면 입력 시퀀스의 길이에 영향 받지 않고 시퀀스 사이의 유사도를 측정할 수 있다. LSH 알고리즘 중 Nilsimsa 알고리즘이 있으나Nilsimsa 알고리즘을 API 호출 시퀀스 유사도 측정에 곧바로 적용 할 수 없다. 첫째, API 호출 시퀀스는 character 단위가 아닌 문자열(API) 단위로 의미를 가지는데, 기존의 알고리즘은 문자 열을 character 단위로 나누어 연산을 수행한다. 즉, CreateFileA라는 API는 C, r, e, a, t, e, F, i, l, e, A 라는 11개의 서로 다른 문자로 나누어져 처리된다. 따라서 문자열에서 character 단위로 처리하지 않고 문자열 리스트에서 문자열 단위로(ex - ["CreateFileA", "WriteFileA", ...]) 처리하도록 알고리즘을 수정할 필요가 있다. 둘째, 기존에는 ASCII 문자에 대해서 utf-8 인코딩을 적용하였기 때문에 utf-8에서 표현할 수 있는 256가지의 서로 다른 입력값만을 처리할 수 있었다. 본 실시예는 utf-8 인코딩 방식 대신 새로운 API 매핑 테이블을 적용 하여 512가지의 서로 다른 입력값을 처리한다. 프로세서는 액션 룰에 등록된 제1 행위 정보와 추출된 제1 행위 정보 간의 유사도를 측정하고, 측정한 유사도를 기준으로 상기 제2 행위 정보를 추출한다. 프로세서는 추출된 제1 행위 정보로부터 애플리케이션 프로그래밍 인터페이스 호출 명령에 관한 문자열을 추출 하고, 추출한 문자열을 배열한 문자열 리스트를 생성하고, 애플리케이션 프로그래밍 인터페이스 호출 명령과 숫 자가 매칭된 매핑 테이블을 이용하여 문자열 리스트에 속한 각 문자열에 대응하는 숫자로 변환한다. 프로세서는 변환된 숫자에 지역적 센시티브 해시 연산을 수행하여 유사도를 측정한다. 프로세서는 API 호출 시퀀스 (문자열 리스트)를 LSH 함수를 이용하여 고정 길이의 해시 값으로 변환하고, 생성 된 해시 값의 차이를 이용하여 유사도를 계산한다. 유사도를 측정하는 방식은 비교하고자 하는 두 개의 해시 값 을 비트 단위(bit-wise) 비교를 수행한다. 예컨대, [동일한 비트의 수 / 전체 비트의 수 (해시 길이)] 로 계산 하여 0부터 1사이의 값을 획득할 수 있다. 프로세서는 리스트를 입력값으로 하여 유사도를 연산한다. 이때 리스트에 포함된 각 문자열을 순서대로 맵핑 테 이블에 넣어 숫자로 변환하여 사용한다. 맵핑 테이블은 새롭게 정의된 테이블로, 512개의 서로 다른 문자열을 각각 특정 숫자로 변환하는 테이블이며, 예컨대, 출력값은 0~511 사이의 정수가 될 수 있다. 실시예들에 의하면, API 호출에 사용되는 매개변수를 고려하여 정의된 액션 룰을 기준으로 하위 레벨의 제1 행 위 정보를 상위 레벨의 제2 행위 정보로 추상화하고, 시퀀스에 따른 기능 룰을 기준으로 상위 레벨의 제2 행위 정보로부터 기능 정보를 추정함으로써, 악성 코드의 주요 목적 또는 핵심 기능을 예측할 수 있고, 다른 API를 호출한 변종 악성 코드에 대하여 동일 기능을 수행하는지 여부를 판단할 수 있다. 악성코드의

기능 분석 장치에 포함된 복수의 구성요소들은 상호 결합되어 적어도 하나의 모듈로 구현될 수 있다. 구성요소들은 장치 내부의 소프트웨어적인 모듈 또는 하드웨어적인 모듈을 연결하는 통신 경로에 연결되 어 상호 간에 유기적으로 동작한다. 이러한 구성요소들은 하나 이상의 통신 버스 또는 신호선을 이용하여 통신 한다. 악성코드의 기능 분석 장치는 하드웨어, 펌웨어, 소프트웨어 또는 이들의 조합에 의해 로직회로 내에서 구현될 수 있고, 범용 또는 특정 목적 컴퓨터를 이용하여 구현될 수도 있다. 장치는 고정배선형(Hardwired) 기기, 필드 프로그램 가능한 게이트 어레이(Field Programmable Gate Array, FPGA), 주문형 반도체(Application Specific Integrated Circuit, ASIC) 등을 이용하여 구현될 수 있다. 또한, 장치는 하나 이상의 프로세서 및 컨트롤러를 포함한 시스템온칩(System on Chip, SoC)으로 구현될 수 있다. 악성코드의 기능 분석 장치는 하드웨어적 요소가 마련된 컴퓨팅 디바이스에 소프트웨어, 하드웨어, 또는 이들의 조합하는 형태로 탑재될 수 있다. 컴퓨팅 디바이스는 각종 기기 또는 유무선 통신망과 통신을 수행하기 위한 통 신 모뎀 등의 통신장치, 프로그램을 실행하기 위한 데이터를 저장하는 메모리, 프로그램을 실행하여 연산 및 명 령하기 위한 마이크로프로세서 등을 전부 또는 일부 포함한 다양한 장치를 의미할 수 있다. 도 1 및 도 2에서는 각각의 과정을 순차적으로 실행하는 것으로 기재하고 있으나 이는 예시적으로 설명한 것에 불과하고, 이 분야의 기술자라면 본 발명의 실시예의 본질적인 특성에서 벗어나지 않는 범위에서 도면에 기재된 순서를 변경하여 실행하거나 또는 하나 이상의 과정을 병렬적으로 실행하거나 다른 과정을 추가하는 것으로 다 양하게 수정 및 변형하여 적용 가능할 것이다. 도 13은 예시적인 실시예들에서 사용되기에 적합한 컴퓨팅 디바이스를 포함하는 컴퓨팅 환경을 예시하여 설명하 기 위한 블록도이다. 도시된 실시예에서, 각 컴포넌트들은 이하에 기술된 것 이외에 상이한 기능 및 능력을 가 질 수 있고, 이하에 기술되지 것 이외에도 추가적인 컴포넌트를 포함할 수 있다. 도시된 컴퓨팅 환경은 컴퓨팅 디바이스(12)를 포함한다. 일 실시예에서, 컴퓨팅 디바이스(12)는 타 단말과 신호 를 송수신하는 모든 형태의 컴퓨팅 디바이스일 수 있다. 컴퓨팅 디바이스(12)는 적어도 하나의 프로세서(14), 컴퓨터 판독 가능한 저장매체(16) 및 통신 버스(18)를 포 함한다. 프로세서(14)는 컴퓨팅 디바이스(12)로 하여금 앞서 언급된 예시적인 실시예에 따라 동작하도록 할 수 있다. 예컨대, 프로세서(14)는 컴퓨터 판독 가능한 저장 매체(16)에 저장된 하나 이상의 프로그램들을 실행할 수 있다. 상기 하나 이상의 프로그램들은 하나 이상의 컴퓨터 실행 가능 명령어를 포함할 수 있으며, 상기 컴퓨 터 실행 가능 명령어는 프로세서(14)에 의해 실행되는 경우 컴퓨팅 장치(12)로 하여금 예시적인 실시예에 따른 동작들을 수행하도록 구성될 수 있다. 컴퓨터 판독 가능한 저장 매체(16)는 컴퓨터 실행 가능 명령어 내지 프로그램 코드, 프로그램 데이터 및/또는 다른 적합한 형태의 정보를 저장하도록 구성된다. 컴퓨터 판독 가능한 저장 매체(16)에 저장된 프로그램(20)은 프로세서(14)에 의해 실행 가능한 명령어의 집합을 포함한다. 일 실시예에서, 컴퓨터 판독한 가능 저장 매체 (16)는 메모리(랜덤 액세스 메모리와 같은 휘발성 메모리, 비휘발성 메모리, 또는 이들의 적절한 조합), 하나 이상의 자기 디스크 저장 디바이스들, 광학 디스크 저장 디바이스들, 플래시 메모리 디바이스들, 그 밖에 컴퓨 팅 디바이스(12)에 의해 액세스되고 원하는 정보를 저장할 수 있는 다른 형태의 저장 매체, 또는 이들의 적합한 조합일 수 있다. 통신 버스(18)는 프로세서(14), 컴퓨터 판독 가능한 저장 매체(16)를 포함하여 컴퓨팅 장치(12)의 다른 다양한 컴포넌트들을 상호 연결한다. 컴퓨팅 디바이스(12)는 또한 하나 이상의 입출력 장치(24)를 위한 인터페이스를 제공하는 하나 이상의 입출력 인터페이스(22) 및 하나 이상의 통신 인터페이스(26)를 포함할 수 있다. 입출력 인터페이스(22) 및 통신 인터페 이스(26)는 통신 버스(18)에 연결된다. 입출력 장치(미

도시)는 입출력 인터페이스(22)를 통해 컴퓨팅 디바이스 (12)의 다른 컴포넌트들에 연결될 수 있다. 예시적인 입출력 장치는 포인팅 장치(마우스 또는 트랙패드 등), 키 보드, 터치 입력 장치(터치패드 또는 터치스크린 등), 음성 또는 소리 입력 장치, 다양한 종류의 센서 장치 및/또는 촬영 장치와 같은 입력 장치, 및/또는 디스플레이 장치, 프린터, 스피커 및/또는 네트워크 카드와 같은 출 력 장치를 포함할 수 있다. 예시적인 입출력 장치(24)는 컴퓨팅 디바이스(12)를 구성하는 일 컴포넌트로서 컴퓨 팅 디바이스(12)의 내부에 포함될 수도 있고, 컴퓨팅 디바이스(12)와는 구별되는 별개의 장치로 컴퓨팅 디바이 스와 연결될 수도 있다. 본 실시예들에 따른 동작은 다양한 컴퓨터 수단을 통하여 수행될 수 있는 프로그램 명령 형태로 구현되어 컴퓨 터 판독 가능한 매체에 기록될 수 있다. 컴퓨터 판독 가능한 매체는 실행을 위해 프로세서에 명령어를 제공하는 데 참여한 임의의 매체를 나타낸다. 컴퓨터 판독 가능한 매체는 프로그램 명령, 데이터 파일, 데이터 구조 또는 이들의 조합을 포함할 수 있다. 예를 들면, 자기 매체, 광기록 매체, 메모리 등이 있을 수 있다. 컴퓨터 프로그 램은 네트워크로 연결된 컴퓨터 시스템 상에 분산되어 분산 방식으로 컴퓨터가 읽을 수 있는 코드가 저장되고 실행될 수도 있다. 본 실시예를 구현하기 위한 기능적인(Functional) 프로그램, 코드, 및 코드 세그먼트들은 본 실시예가 속하는 기술분야의 프로그래머들에 의해 용이하게 추론될 수 있을 것이다. 본 실시예들은 본 실시예의 기술 사상을 설명하기 위한 것이고, 이러한 실시예에 의하여 본 실시예의 기술 사상 의 범위가 한정되는 것은 아니다. 본 실시예의 보호 범위는 아래의 청구범위에 의하여 해석되어야 하며, 그와 동등한 범위 내에 있는 모든 기술 사상은 본 실시예의 권리범위에 포함되는 것으로 해석되어야 할 것이다.