(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2019년02월15일
(11) 등록번호 10-1948711
(24) 등록일자 2019년02월11일

(51) 국제특허분류(Int. Cl.)
*G06F 21/56* (2013.01) *G06F 9/54* (2018.01)
*H04L 29/06* (2006.01)
(52) CPC특허분류
*G06F 21/566* (2013.01)
*G06F 9/54* (2013.01)
(21) 출원번호 10-2016-7008888
(22) 출원일자(국제) 2014년09월25일
심사청구일자 2018년09월21일
(85) 번역문제출일자 2016년04월04일
(65) 공개번호 10-2016-0065852
(43) 공개일자 2016년06월09일
(86) 국제출원번호 PCT/RO2014/000027
(87) 국제공개번호 WO 2015/050469
국제공개일자 2015년04월09일
(30) 우선권주장
14/046,728 2013년10월04일 미국(US)
(56) 선행기술조사문헌
US20130145463 A1*
US20100058473 A1*
KR101057432 B1
KR1020130076266 A
*는 심사관에 의하여 인용된 문헌

(73) 특허권자
**비트데펜더 아이피알 매니지먼트 엘티디**
사이프러스 니코시아 1076 12 피시 크레온토스
(72) 발명자
**루칵스, 산도르**
루마니아 주데출 클루지, 사트 플로레슈티, 이티. 3, 불레바르둘 체타테아 페테이 비엘. 비
**토샤, 라울-바실레**
루마니아 주데출 클루지, 클루지-나포카, 에이피. 30,이티. 4, 스트라다 에드가르 키네 엔알. 32
*(뒷면에 계속)*
(74) 대리인
**권영준**

전체 청구항 수 : 총 22 항 심사관 : 윤혜숙

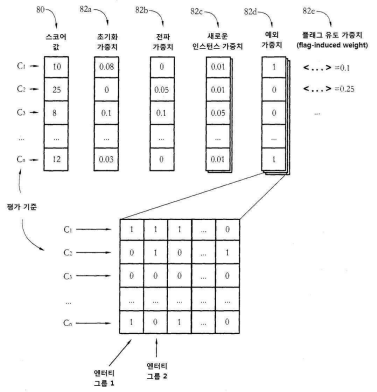
(54) 발명의 명칭 **멀웨어 탐지를 위한 복합 스코어링**

## (57) 요 약

바이러스, 트로이 목마 바이러스(Trojan) 및 스파이웨어와 같은 멀웨어로부터 컴퓨터 시스템을 보호하게 할 수 있는 시스템과 방법이 설명된다. (컴퓨터 시스템에서 실행되는 프로세스와 쓰레드와 같은) 복수의 실행 엔터티들 각각에 대하여,스코어링 엔진이 복수의 평가 스코어들을 기록하고, 각 스코어는 특징적 평가 기준에 따라서 결정

(뒷면에 계속)

*대 표 도* - 도9



된다. 엔터티가 평가 기준을 만족(예를 들어서, 활동을 수행)할 때마다, 상기 엔터티의 개별 스코어는 업데이트된다. 엔터티의 스코어를 업데이트하는 것은, 관련 엔터티가 종료되었을 때 조차도, 즉, 더 이상 활성이 아닐때 조차도, 개별 엔터티에 관련된 엔터티들의 스코어 업데이트를 트리거링할 수 있다. 관련 엔터티들에는 특히 개별 엔터티의 부모, 및/또는 개별 엔터티로 코드를 삽입하는 엔터티가 있을 수 있다. 스코어링 엔진은 개별 엔터티의 복수의 평가 스코어에 따라서 엔터티가 악성인지 여부를 결정한다.

(52) CPC특허분류
*H04L 63/14* (2013.01)
(72) 발명자
**보카, 파울-다니엘**
루마니아 주데출 클루지, 클루지-나포카, 에이피. 22, 에스씨. 2, 스트라다 아그리쿨토릴로르 엔알. 20
**하지마샨, 게오르게-플로린**
루마니아 주데출 알바, 코무나 룬카 무레술루이, 사트 룬카무레술루이 엔알. 351
**루차스, 안드레이-블라드**
루마니아 주데출 사투 마레, 사투 마레, 불레바르 둘 클로슈카 엔알. 111

# 명 세 서

## 청구범위

### 청구항 1

엔터티 관리 모듈을 실행하도록 구성된 적어도 하나의 프로세서, 엔터티 평가자, 및 스코어링 엔진을 포함하는 호스트 시스템으로서,

상기 엔터티 관리 모듈은 평가된 소프트웨어 엔터티들의 집합(collection)을 관리하도록 구성되고,

상기 집합을 관리하는 것은, 상기 집합의 제1 엔터티의 파생 엔터티(descendant entity)들의 세트를 식별하는 것; 상기 제1 엔터티가 종료되었는지 결정하는 것; 응답으로 상기 제1 엔터티가 종료되었을 때 파생 엔터티들의 세트의 모든 멤버들이 종료되는지 결정하는 것; 및 응답으로 파생 엔터티들의 세트의 모든 멤버들이 종료 될 때 상기 제1 엔터티를 상기 집합으로부터 제거하는 것을 포함하고,

상기 엔터티 평가자는, 평가 기준에 따라서 상기 제1 엔터티를 평가하고; 응답으로 상기 제1 엔터티가 평가 기준을 만족할 때 평가 표시자를 스코어링 엔진에 전송하도록 구성되고,

상기 스코어링 엔진은, 상기 제1 엔터티에 대하여 결정된 제1 스코어와 상기 집합의 제2 엔터티에 대하여 결정된 제2 스코어를 기록하고(상기 제1 및 제2 스코어들은 상기 평가 기준에 따라서 결정됨); 상기 제1 스코어와 제2 스코어를 기록하는 것에 응답으로 그리고 평가 표시자를 수신하는 것에 응답으로, 상기 평가 표시자에 따라서 상기 제2 스코어를 업데이트하고; 응답으로 상기 제2 엔터티가 상기 업데이트된 제2 스코어에 따라서 악성인지를 결정하도록 구성되며,

상기 스코어는 상기 엔터티들 사이에서 전파되도록 구성되는 것을 특징으로 하는 호스트 시스템.

### 청구항 2

제1항에 있어서,

상기 스코어링 엔진은,

상기 평가 표시자를 수신하는 것에 응답으로, 상기 평가 표시자에 따라서 상기 제1 스코어를 업데이트하며,

응답으로, 상기 제1 엔터티가 상기 업데이트된 제1 스코어에 따라서 악성인지를 결정하도록 추가적으로 구성되는 것을 특징으로 하는 호스트 시스템.

### 청구항 3

제1항에 있어서,

상기 제1 엔터티는 상기 제2 엔터티의 자식인 것을 특징으로 하는 호스트 시스템.

### 청구항 4

제1항에 있어서,

상기 제2 엔터티는 상기 제1 엔터티의 자식인 것을 특징으로 하는 호스트 시스템.

### 청구항 5

제1항에 있어서,

상기 제1 엔터티는 상기 제2 엔터티에 의하여 삽입된 코드의 섹션을 포함하는 것을 특징으로 하는 호스트 시스템.

### 청구항 6

제1항에 있어서,

상기 제2 엔터티는 상기 제1 엔터티에 의하여 삽입된 코드의 섹션을 포함하는 것을 특징으로 하는 호스트 시스템.

**청구항 7**

제1항에 있어서,

상기 제2 스코어를 업데이트하는 것은 $w \cdot S$에 따라서 결정된 양만큼 상기 제2 스코어를 변경하는 것을 포함하고, 이 때 S는 상기 제1 스코어, w는 수치 가중치(numerical weight)인 것을 특징으로 하는 호스트 시스템.

**청구항 8**

제1항에 있어서,

평가된 소프트웨어 엔터티들의 집합을 관리하는 것은,

새로운 소프트웨어 엔터티의 개시를 중간차단하는 것과,

응답으로, 상기 새로운 소프트웨어 엔터티를 상기 집합에 추가하는 것을 추가적으로 포함하는 것을 특징으로 하는 호스트 시스템.

**청구항 9**

실행될 때, 호스트 시스템의 적어도 하나의 프로세서가,

- 평가된 소프트웨어 엔터티들의 집합을 관리하도록 구성하고(이 때, 상기 집합을 관리하는 것은, 상기 집합의 제1 엔터티의 파생 엔터티(descendant entity)들의 세트를 식별하는 것; 제1 엔터티가 종료되는지 결정하는 것; 응답으로 상기 제1 엔터티가 종료될 때 파생 엔터티들의 세트의 모든 멤버들이 종료되는지 결정하는 것; 및 응답으로 파생 엔터티들의 세트의 모든 멤버들이 종료될 때 선택된 엔터티를 상기 집합으로부터 제거하는 것을 포함),

- 상기 제1 엔터티에 대하여 결정된 제1 스코어와 상기 집합의 제2 엔터티에 대하여 결정된 제2 스코어를 기록하도록 구성하고(이 때  상기 제1 스코어와 제2 스코어는 평가 기준에 따라서 결정),

- 상기 제1 및 제2 스코어들을 기록하는 것에 응답으로, 상기 평가 기준에 따라서 제1 엔터티를 평가하도록 구성하고,

- 상기 제1 엔터티를 평가하는 것에 응답으로, 상기 제1 엔터티가 평가 기준을 만족할 때, 상기 제2 스코어를 업데이트하도록 구성하며,

- 상기 제2 스코어를 업데이트하는 것에 응답으로, 상기 업데이트된 제2 스코어에 따라서 상기 제2 엔터티가 악성인지 여부를 결정하도록 구성하는 명령들을 저장하는 비-일시적 컴퓨터 판독가능 매체로서,

상기 스코어는 상기 엔터티들 사이에서 전파되도록 구성되는 비-일시적 컴퓨터 판독가능 매체.

**청구항 10**

제9항에 있어서,

상기 적어도 하나의 프로세서는,

상기 제1 엔터티를 평가하는 것에 응답으로, 상기 제1 엔터티가 상기 평가기준을 만족할 때, 상기 제1 스코어를 업데이트하며,

응답으로, 상기 제1 엔터티가 상기 업데이트된 제1 스코어에 따라서 악성인지를 결정하도록 추가적으로 구성되는 것을 특징으로 하는 비-일시적 컴퓨터 판독가능 매체.

**청구항 11**

제9항에 있어서,

상제 제1 엔터티는 상기 제2 엔터티의 자식인 것을 특징으로 하는 비-일시적 컴퓨터 판독가능 매체.

**청구항 12**

제9항에 있어서,

상제 제2 엔터티는 상기 제1 엔터티의 자식인 것을 특징으로 하는 비-일시적 컴퓨터 판독가능 매체.

**청구항 13**

제9항에 있어서,

상기 제1 엔터티는 상기 제2 엔터티에 의하여 삽입된 코드의 섹션을 포함하는 것을 특징으로 하는 비-일시적 컴퓨터 판독가능 매체.

**청구항 14**

제9항에 있어서,

상기 제2 엔터티는 상기 제1 엔터티에 의하여 삽입된 코드의 섹션을 포함하는 것을 특징으로 하는 비-일시적 컴퓨터 판독가능 매체.

**청구항 15**

제9항에 있어서,

상기 제2 스코어를 업데이트하는 것은 $w \cdot S$에 따라서 결정된 양만큼 상기 제2 스코어를 변경하는 것을 포함하고, 이 때 S는 상기 제1 스코어, w는 수치 가중치(numerical weight)인 것을 특징으로 하는 비-일시적 컴퓨터 판독가능 매체.

**청구항 16**

제9항에 있어서,

평가된 소프트웨어 엔터티들의 집합을 관리하는 것은,

새로운 소프트웨어 엔터티의 개시를 중간차단하는 것과,

응답으로, 상기 새로운 소프트웨어 엔터티를 상기 집합에 추가하는 것을 추가적으로 포함하는 것을 특징으로 하는 비-일시적 컴퓨터 판독가능 매체.

**청구항 17**

엔터티 평가자와 스코어링 엔진을 실행하도록 구성된 적어도 하나의 프로세서를 포함하는 호스트 시스템으로서,

상기 엔터티 평가자는, 평가 기준에 따라서 제1 소프트웨어 엔터티를 평가하도록(상기 제1 소프트웨어 엔터티는 클라이언트 시스템에서 실행됨); 그리고, 응답으로, 상기 제1 소프트웨어 엔터티가 상기 평가 기준을 만족하면, 상기 스코어링 엔진에 평가 표시자를 전송하도록 구성되고,

상기 스코어링 엔진은, 상기 평가 표시자를 수신하는 것에 응답으로, 상기 평가 표시자에 따라서 스코어를 업데이트하도록(상기 스코어는 상기 호스트 시스템에서 이전에 실행되는 제2 소프트웨어 엔터티에 대하여 결정되고, 상기 제2 소프트웨어 엔터티는 상기 스코어를 업데이팅할 때 종료됨); 그리고 응답으로, 상기 제2 소프트웨어 엔터티가 상기 업데이트된 제2 스코어에 따라서 악성인지 여부를 결정하도록 구성되고,

상기 스코어는 상기 엔터티들 사이에서 전파되도록 구성되는 것을 특징으로 하는 호스트 시스템.

**청구항 18**

제17항에 있어서,

상기 제1 소프트웨어 엔터티는 상기 제2 소프트웨어 엔터티의 자식인 것을 특징으로 하는 호스트 시스템.

**청구항 19**

제17항에 있어서,

상기 제1 소프트웨어 엔터티는 상기 제2 소프트웨어 엔터티에 의해서 삽입된 코드 섹션을 포함하는 것을 특징으로 하는 호스트 시스템.

**청구항 20**

호스트 시스템에서 실행되는 제1 소프트웨어 엔터티가 평가 기준을 만족하는지 여부를 결정하기 위하여 상기 호스트 시스템의 적어도 하나의 프로세서를 채용하는 단계,

응답으로, 상기 제1 소프트웨어 엔터티가 상기 평가 기준을 만족할 때 상기 호스트 시스템에서 이전에 실행된 제2 소프트웨어 엔터티에 대하여 결정된 스코어를 업데이트 하기 위하여 적어도 하나의 프로세서를 채용하는 단계(상기 제2 소프트웨어 엔터티는 상기 스코어의 업데이트 시점에 종료되고, 상기 스코어는 상기 평가 기준에 따라서 결정됨),

상기 스코어를 업데이트 하는 것에 응답으로, 상기 제2 소프트웨어 엔터티가 상기 업데이트된 스코어에 따라서 악성인지 여부를 결정하기 위하여 상기 적어도 하나의 프로세서를 채용하는 단계를 포함하고,

상기 스코어는 상기 엔터티들 사이에서 전파되도록 구성되는 것을 특징으로 하는 방법.

**청구항 21**

제20항에 있어서,

상기 제1 소프트웨어 엔터티는 상기 제2 소프트웨어 엔터티의 자식인 것을 특징으로 하는 방법.

**청구항 22**

제20항에 있어서,

상기 제1 소프트웨어 엔터티는 상기 제2 소프트웨어 엔터티에 의해서 삽입된 코드 섹션을 포함하는 것을 특징으로 하는 방법.

## *발명의 설명*

### *기 술 분 야*

[0001] 본 발명은 멀웨어로부터 컴퓨터 시스템을 보호하기 위한 시스템 및 방법에 대한 것이다.

### *배 경 기 술*

[0002] 멀웨어로도 알려진 악성 소프트웨어는 세계적으로 상당수의 컴퓨터 시스템에 영향을 주고 있다. 멀웨어는 컴퓨터 바이러스, 웜, 루트킷(rootkit) 및 스파이웨어와 같은 많은 형태로, 수백만의 컴퓨터 사용자에게 심각한 위협이 되고 있으며, 무엇보다도 데이터 및 민감한 정보의 손실, 신원 도용, 및 생산성 손실에 있어 이들을 취약하게 하고 있다.

[0003] 보안 소프트웨어가 사용자의 컴퓨터 시스템을 감염시킨 멀웨어를 탐지하는 데 그리고 추가적으로 그러한 멀웨어를 제거하거나 실행을 중지시키는 데 사용될 수 있다. 여러 멀웨어 탐지 기술들이 본 기술분야에 알려져 있다. 일부는 멀웨어 에이전트의 코드 단편을 멀웨어를 가리키는 서명들의 라이브러리에 매칭시키는 것에 의한다. 다른 종래의 방법은 멀웨어 에이전트의 멀웨어를 나타내는 행동들의 세트를 탐지한다.

[0004] 보안 소프트웨어의 작동을 손상하거나 그리고/또는 탐지를 피하기 위하여, 일부 멀웨어 에이전트는 이들의 코드를 암호화하는 것과 같은 난독화 기술(obfuscation technique)을 채용하거나 또는 각각의 감염된 컴퓨터 시스템에 조금씩 다른 코드 버전을 사용한다(폴리모피즘). 다른 예시적인 탐지 회피 방법들은 악성 활동들을 여러 활동으로 분할하고, 각 활동이 별도의 에이전트에 의해서, 가능하다면 시차(time delay)를 두고서 수행되게 한다. 다른 예에서는, 멀웨어가, 예를 들어서 권한 상승을 이용해서 그리고/또는 보안 소프트웨어의 코드를 덮어씀으로써 보안 소프트웨어를 적극적으로 공격하거나 불능화하려고 시도할 수 있다.

### *발명의 내용*

### 해결하려는 과제

[0005] 일련의 신속하게 변화하는 멀웨어 위협들에 대처하기 위하여 강력하며 확장가능한(scalable) 안티-멀웨어 솔루션의 개발의 강력한 요청이 있다.

### 과제의 해결 수단

[0006] 본 발명의 일 태양에 따르면, 호스트 시스템은 엔터티 관리 모듈을 실행하도록 구성된 적어도 하나의 프로세서, 엔터티 평가자, 및 스코어링 엔진을 포함한다. 엔터티 관리 모듈은 평가된 소프트웨어 엔터티들의 집합(collection)을 관리하도록 구성되고, 상기 집합을 관리하는 것은 상기 집합의 제1 엔터티의 파생 엔터티들(descendant entity)의 세트를 식별하는 것, 제1 엔터티가 종료되는지 결정하는 것, 응답으로 제1 엔터티가 종료되었을 때 파생 엔터티들의 세트의 모든 멤버들이 종료되는지 결정하는 것, 및 응답으로 파생 엔터티들의 세트의 모든 멤버들이 종료될 때 제1 엔터티를 상기 집합으로부터 제거하는 것을 포함한다. 상기 엔터티 평가자는, 평가 기준에 따라서 제1 엔터티를 평가하고, 응답으로 제1 엔터티가 평가 기준을 만족할 때 평가 표시자를 스코어링 엔진에 전송하도록 구성된다. 스코어링 엔진은 제1 엔터티에 대하여 결정된 제1 스코어와 상기 집합의 제2 엔터티에 대하여 결정된 제2 스코어를 기록하도록 구성되고(상기 제1 및 제2 스코어들은 상기 평가 기준에 따라서 결정됨), 제1 스코어와 제2 스코어를 기록하는 것에 응답으로 그리고 평가 표시자를 수신하는 것에 응답으로, 평가 표시자에 따라서 제2 스코어를 업데이트하도록 구성되고, 그리고 응답으로 제2 엔터티가 상기 업데이트된 제2 스코어에 따라서 악성인지를 결정하도록 구성된다.

[0007] 본 발명의 다른 태양에 따르면, 비-일시적 컴퓨터 판독가능 매체는, 실행될 때, 평가된 소프트웨어 엔터티들의 집합을 관리하기 위하여 호스트 시스템의 적어도 하나의 프로세서를 설정하는 명령들을 저장하고, 상기 집합을 관리하는 것은, 상기 집합의 제1 엔터티의 파생 엔터티들(descendant entity)의 세트를 식별하는 것, 제1 엔터티가 종료되는지 결정하는 것, 응답으로 제1 엔터티가 종료될 때 파생 엔터티들의 세트의 모든 멤버들이 종료되는지 결정하는 것, 및 응답으로 파생 엔터티들의 세트의 모든 멤버들이 종료되었을 때 선택된 엔터티를 상기 집합으로부터 제거하는 것을 포함한다. 상기 명령들은 상기 적어도 하나의 프로세서가 제1 엔터티에 대하여 결정된 제1 스코어와 상기 집합의 제2 엔터티에 대하여 결정된 제2 스코어를 기록하도록 추가적으로 설정하고, 제1 스코어와 제2 스코어는 평가 기준에 따라서 결정된다. 상기 명령들은 상기 적어도 하나의 프로세서가 상기 제1 및 제2 스코어들을 기록하는 것에 응답으로 상기 평가 기준에 따라서 제1 엔터티를 평가하도록 추가적으로 구성한다. 상기 명령들은 상기 적어도 하나의 프로세서가, 상기 제1 엔터티를 평가하는 것에 응답으로, 상기 제1 엔터티가 평가 기준을 만족할 때, 상기 제2 스코어를 업데이트하고, 상기 제2 스코어를 업데이트하는 것에 응답으로, 상기 업데이트된 제2 스코어에 따라서 상기 제2 엔터티가 악성인지 여부를 결정하도록 추가적으로 구성된다.

[0008] 본 발명의 다른 태양에 따르면, 호스트 시스템은 엔터티 평가자와 스코어링 엔진을 실행하도록 구성된 적어도 하나의 프로세서를 포함한다. 상기 엔터티 평가자는, 평가 기준에 따라서 제1 소프트웨어 엔터티를 평가하도록 구성되고(상기 제1 소프트웨어 엔터티는 클라이언트 시스템에서 실행됨), 그리고, 응답으로, 제1 소프트웨어 엔터티가 평가 기준을 만족하면, 상기 스코어링 엔진에 평가 표시자를 전송하도록 구성된다. 상기 스코어링 엔진은, 상기 평가 표시자를 수신하는 것에 응답으로, 평가 표시자에 따라서 스코어를 업데이트하도록 구성되는데, 상기 스코어는 호스트 시스템에서 이전에 실행되는 제2 소프트웨어 엔터티에 대하여 결정되고, 제2 소프트웨어 엔터티는 상기 스코어를 업데이팅할 때 만료된다. 상기 스코어링 엔진은, 상기 제2 스코어를 업데이트하는 것에 응답으로, 제2 소프트웨어 엔터티가 상기 업데이트된 제2 스코어에 따라서 악성인지 여부를 결정하도록 추가적으로 구성된다.

[0009] 본 발명의 다른 태양에 따르면, 본 발명의 방법은 호스트 시스템에서 실행되는 제1 소프트웨어 엔터티가 평가 기준을 만족하는지 여부를 결정하기 위하여 호스트 시스템의 적어도 하나의 프로세서를 채용하는 것을 포함한다. 상기 방법은 상기 제1 소프트웨어 엔터티가 상기 평가 기준을 만족할 때 상기 호스트 시스템에서 이전에 실행된 제2 소프트웨어 엔터티에 대하여 결정된 스코어를 업데이트 하기 위하여 적어도 하나의 프로세서를 채용하는 것을 추가적으로 포함하고, 상기 제2 소프트웨어 엔터티는 상기 스코어의 업데이트 시점에 종료되고, 상기 스코어는 상기 평가 기준에 따라서 결정된다. 상기 방법은, 상기 제2 스코어를 업데이트 하는 것에 응답으로, 제2 소프트웨어 엔터티가 상기 업데이트된 제2 스코어에 따라서 악성인지 여부를 결정하기 위하여 적어도 하나의 프로세서를 채용하는 것을 추가적으로 포함한다.

## 도면의 간단한 설명

[0010] 본 발명의 전술한 태양들 및 장점은 후술하는 상세한 설명 및 도면을 참조로 이해하면 더욱 잘 이해될 것이다.

도 1은 본 발명의 일부 실시예에 따른 멀웨어로부터 보호된 호스트 컴퓨터 시스템의 예시적 하드웨어 구성을 보여준다.

도 2a는 본 발명의 일부 실시예에 따른 호스트 시스템에서 실행되는 보안 어플리케이션을 포함하는 소프트웨어 객체의 예시적 세트를 보여준다.

도 2b는 가상화를 지원하도록 구성된 호스트 시스템에서, 가상 머신 내에서 실행되는 보안 어플리케이션을 포함하는 소프트웨어 객체들의 예시적인 세트를 보여준다.

도 3은 본 발명의 일부 실시예에 따른 안티 멀웨어 객체들의 세트를 포함하여, 다양한 프로세서 권한 레벨들에서 호스트 시스템에서 실행되는 소프트웨어 객체들의 예시적 계층을 보여준다.

도 4는 본 발명의 일부 실시예에 따른 도 3의 엔터티 관리 모듈에 의해서 수행되는 단계들의 예시적인 시퀀스를 보여준다.

도 5는 본 발명의 일부 실시예에 따른 복수의 엔터티 평가자 모듈들에 의하여 소프트웨어 엔터티에 대하여 결정되는 복수의 엔터티 평가 표시자들을 수신하는 예시적인 스코어링 엔진을 보여준다.

도 6은 Windows® 환경에서 일련의 프로세스들의 예시적 실행 흐름을 보여준다. 실선 화살표는 안티-멀웨어 시스템이 없을 때의 예시적 실행 흐름을 나타낸다. 파선 화살표는 본 발명의 일부 실시예에 따라 작동하는 복수의 엔터티 평가자에 의하여 도입되는 실행 흐름의 변경을 보여준다.

도 7은 본 발명의 일부 실시예에 따른 엔터티 평가자 모듈에 의해서 수행되는 단계들의 예시적인 시퀀스를 보여준다.

도 8은 복수의 예시적인 엔터티 스코어링 객체들(ESO, entity scoring object)을 보여주고, 각 ESO는 본 발명의 일부 실시예에 따른 개별 소프트웨어 엔터티에 대하여 결정된다. ESO의 예시적인 데이터 필드들은 특히, 엔터티 아이덴터티 표시자(entity identity indicator, EID), 복수의 스코어들($S_i$), 및 개별 엔터티에 대하여 결정된 총합 스코어 A를 포함한다.

도 9는 본 발명의 일부 실시예에 따라서 소프트웨어 엔터티들을 스코어링 하기 위하여 스코어링 엔진에 의하여 사용되는 스코어 값들의 예시적인 세트와, 다양한 예시적인 가중치의 세트를 보여준다.

도 10은 본 발명의 일부 실시예에 따라서 스코어링 엔진(도 3 내지 도 4)에 의해서 수행되는 단계들의 예시적인 시퀀스를 보여준다.

도 11은 컴퓨터 네트워크를 통하여 보안 서버에 연결되는 복수의 호스트 시스템을 포함하는 예시적인 구성을 보여준다.

도 12는 본 발명의 일부 실시예에 따른 호스트 시스템과 보안 서버 사이의 예시적인 안티-멀웨어 트랜잭션을 보여준다.

## 발명을 실시하기 위한 구체적인 내용

[0011] 이하의 설명에서, 구조들 사이에서 언급된 모든 연결들은 직접적인 동작 연결들 또는 매개 구조들을 통한 간접적인 동작 연결들일 수 있는 것으로 이해된다. 구성 요소들의 세트는 하나 이상의 구성 요소를 포함한다. 구성 요소의 임의의 열거는 적어도 하나의 구성 요소를 언급하는 것으로 이해된다. 복수의 구성 요소는 적어도 2개의 구성 요소를 포함한다. 달리 요구되지 않는다면, 기술된 어떠한 방법 단계들도 설명된 특정 순서로 반드시 실행될 필요는 없다. 제2 구성 요소로부터 유도되는 제1 구성 요소(예컨대, 데이터)는 제2 구성 요소와 동일한 제1 구성 요소는 물론, 제2 구성 요소 그리고 선택적으로는 다른 데이터를 처리하는 것에 의해 생성된 제1 구성 요소를 포함한다.  파라미터에 따라 결정 또는 판정하는 것은 파라미터에 따라 그리고 선택적으로는 다른 데이터에 따라 결정 또는 판정하는 것을 포함한다. 달리 구체화되지 않는다면, 일부 수량/데이터의 표시자는 수량/데이터 그 자체, 또는 수량/데이터 그 자체와 상이한 표시자일 수 있다. 달리 구체화되지 않는다면, 프로세스는 컴퓨터 프로그램의 인스턴스(instance)를 나타내고, 컴퓨터 프로그램은 컴퓨터 시스템이 특정 과업을 수행하도록 결정하는 명령들의 시퀀스이다. 컴퓨터 판독 가능 매체는 자성, 광, 및 반도체 저장 매체(예컨대, 하드 드라

이브, 광 디스크, 플래시 메모리, DRAM)와 같은 비-일시적 매체(non-transitory medium)는 물론, 전도성 케이블 및 파이버 옵틱 링크와 같은 통신 링크들을 포함한다. 일부 실시예들에 따르면, 본 발명은, 그 중에서도, 본원에 설명된 방법들을 수행하기 위해 프로그래밍된 하드웨어(예컨대, 하나 이상의 프로세서)는 물론, 본원에서 설명된 방법들을 수행하기 위한 명령들을 인코딩하는 컴퓨터-판독 가능 매체를 포함하는 컴퓨터 시스템을 제공한다.

[0012] 후술하는 설명은 본 발명의 실시예들을 예시적으로 설명하는 것이며, 반드시 제한적인 것은 아니다.

[0013] 도 1은 본 발명의 일부 실시예에 따라서 안티-멀웨어 작동을 수행하는 호스트 시스템(10)의 예시적 하드웨어 구성을 보여준다. 호스트 시스템(10)은 특히, 엔터프라이즈 서버와 같은 기업 컴퓨팅 장치, 또는 퍼스널 컴퓨터나 스마트폰과 같은 최종 사용자 장치(end-user device)를 나타낼 수 있다. 다른 호스트 시스템들은 TV 및 게임 컨솔과 같은 오락 장치, 또는 메모리와 프로세서를 구비하고 가상화를 지원하며 멀웨어 보호가 필요한 모든 다른 장치를 포함한다. 도 1은 설명적 목적으로 컴퓨터 시스템을 나타낸 것이다. 모바일 전화기나 태블릿과 같은 다른 클라이언트 장치들은 다른 구성을 가질 수 있다. 일부 실시예에서, 시스템(10)은 프로세서(12), 메모리 유닛(14), 입력 장치(16)들 세트, 출력 장치(18)들 세트, 저장 장치(20)들 세트, 및 네트워크 어댑터(22)들 세트(이들은 모두 버스(24)들 세트에 의하여 연결됨)를 포함하는 물리적 장치들의 세트를 포함한다.

[0014] 일부 실시예에서, 프로세서(12)는 신호 및/또는 데이터의 세트로 산술 및/또는 논리 연산을 실행하도록 구성된 물리적 장치(예컨대, 멀티-코어 집적 회로)를 포함한다. 일부 실시예들에서, 이러한 논리 연산들은 프로세서 명령(예를 들어, 기계 코드 또는 다른 유형의 소프트웨어)의 시퀀스 형태로 프로세서(12)에 전달된다. 메모리 유닛(14)은 명령들을 수행하는 도중에 프로세서(12)에 의해 액세스되거나 생성되는 데이터/신호들을 저장하는 휘발성 컴퓨터-판독 가능 매체(예컨대, RAM)를 포함할 수 있다. 입력 장치(16)는 사용자가 시스템(10)으로 데이터 및/또는 명령들을 도입할 수 있게 하는 개별 하드웨어 인터페이스 및/또는 어댑터를 포함하는, 특히 컴퓨터 키보드, 마우스, 및 마이크를 포함할 수 있다. 출력 장치(18)는 특히 모니터와 같은 디스플레이 장치 및 스피커는 물론, 시스템(10)이 사용자에게 데이터를 통신하게 할 수 있는 그래픽 카드와 같은 하드웨어 인터페이스/어댑터를 포함할 수 있다. 일부 실시예들에서, 입력 장치(16)와 출력 장치(18)는 터치-스크린 장치들의 경우와 같이, 하드웨어의 공통적인 부품을 공유할 수 있다. 저장 장치(20)는 소프트웨어 명령들 및/또는 데이터의 비휘발성 저장, 판독, 및 기록을 가능하게 하는 컴퓨터-판독 가능 매체를 포함한다. 예시적인 저장 장치(20)는 자기 디스크 및 광 디스크 및 플래시 메모리 장치들은 물론, CD 및/또는 DVD 디스크들 및 드라이브들과 같은 소거 가능 매체를 포함한다. 네트워크 어댑터(22) 세트는 시스템(10)이 컴퓨터 네트워크 및/또는 다른 장치들/컴퓨터 시스템들에 연결될 수 있게 한다. 버스(24)들은 호스트 시스템(10)의 장치(12-22)들의 사이의 통신을 가능하게 하는 복수의 시스템, 주변 장치, 및/또는 칩셋 버스들, 및/또는 다른 모든 회로망을 나타낸다. 예를 들어서, 버스(24)들은 특히 프로세서(12)를 메모리(14)에 연결시키는 노스브리지, 및/또는 프로세서(12)를 장치들(16-22)에 연결시키는 사우스브리지를 포함할 수 있다.

[0015] 도 2a는 하드웨어 가상화를 채용하지 않는 구성에서 호스트 시스템(10)에서 실행되는 소프트웨어 객체들의 예시적 세트를 보여준다. 일부 실시예에서, 게스트 운영 시스템(OS)(34)은 호스트 시스템(10)의 하드웨어에 인터페이스를 제공하고 소프트웨어 어플리케이션(42a-c 및 44)의 세트를 위한 호스트로서 역할하는 소프프웨어를 포함한다. OS(34)는 특히, Windows®, MacOS®, Linux®, iOS®, 또는 Android™과 같은 임의의 널리 이용가능한 운영 시스템을 포함할 수 있다. 어플리케이션(42a-c)은 특히, 워드 프로세싱, 이미지 프로세싱, 데이터베이스, 브라우저 및 전자 통신 어플리케이션을 포함할 수 있다.

[0016] 도 2b는 하드웨어 가상화를 이용하는 실시예에서 호스트 시스템(10)에서 실행되는 소프트웨어 객체들의 예시적인 세트를 보여준다. 게스트 가상 머신(32a-b)들의 세트는 하이퍼바이저(30)에 의하여 노출된다. 가상 머신(Virtual machine, VM)들은 본 기술분야에서 통상적으로 다른 VM들과 독립적으로 고유의 운영 시스템 및 소프트웨어를 각각 구동할 수 있는 실제 물리적 머신/컴퓨터 시스템의 소프트웨어 에뮬레이션(emulation)으로 알려져 있다. 하이퍼바이저(30)는 프로세서 동작, 메모리, 저장소, 입력부/출력부, 및 네트워킹 장치들과 같은 호스트 시스템(10)의 하드웨어 자원의 복수의 가상 머신에 의해 멀티플렉싱(공유)을 허용하는 소프트웨어를 포함한다. 일부 실시예들에서, 하이퍼바이저(30)는 다수의 가상 머신들 및/또는 운영 시스템(OS)들이 다양한 고립도(degree of isolation)로, 호스트 시스템(10) 상에서 동시에 실행되게 한다. 이러한 구성을 가능하게 하기 위해, 하이퍼바이저(30)의 일부를 구성하는 소프트웨어는 복수의 가상화된, 즉 소프트웨어-에뮬레이팅된 장치(software-emulated device)들을 생성할 수 있으며, 각각의 가상화된 장치는 특히 프로세서(12) 및 메모리(14)와 같은 시스템(10)의 물리적 하드웨어 장치를 에뮬레이팅한다. 하이퍼바이저(30)는 호스트 시스템(10) 상에서 작동하는 각각의 VM에 대해 가상 장치들의 세트를 또한 할당할 수 있다. 따라서, 각각의 VM(32a-b)은 고유의

물리적 장치 세트를 구비하는 것처럼, 즉 거의 완벽한 컴퓨터 시스템인 것처럼 작동한다. 가상 머신으로의 가상 장치의 할당과 생성은 개별 VM을 노출하는 것으로서 본 기술분야에서 통상적으로 알려져 있다. 유명한 하이퍼바이저의 예로는, 특히 VMware Inc.의 VMware vSphere™ 및 오픈 소스 Xen 하이퍼바이저가 있다.

[0017] 일부 실시예들에서, 하이퍼바이저(30)는 이하에서 상술하는 바와 같은 안티-멀웨어 작업을 수행하도록 구성된 메모리 인트로스펙션 엔진(40)을 포함한다. 엔진(40)은 하이퍼바이저(30)에 통합되거나, 또는 하이퍼바이저(30)와는 구별되고 독립적인 소프트웨어 구성요소로서 전달될 수 있지만, 하이퍼바이저(30)와 실질적으로 유사한 프로세서 권한 레벨로 실행된다. 단일 엔진(40)은 호스트 시스템(10) 상에서 실행되는 다수의 가상머신에 대해서 멀웨어를 보호하도록 구성될 수 있다.

[0018] 도 2b가 단순화를 위해 단지 2개의 VM(32a-b)을 도시하고 있더라도, 호스트 시스템(10)은 많은 개수의, 예컨대 수백 개의 VM을 동시에 작동시킬 수 있으며, 이러한 VM의 개수는 호스트 시스템(10)이 작동하는 동안 변경될 수 있다. 일부 실시예들에서, 각각의 VM(32a-b)은 각각 호스트 시스템(10) 상에서 다른 VM과 독립적으로 그리고 동시에 실행되는 게스트 운영 시스템(34a-b) 및/또는 소프트웨어 어플리케이션(42d-e, 및 42f) 세트를 실행한다. 각각의 OS(34a-b)는 개별 VM(32a-b)의 (가상화된) 하드웨어에 대한 인터페이스를 제공하고, 개별 OS 상에서 실행되는 컴퓨팅 어플리케이션을 위한 호스트로서의 역할을 하는 소프트웨어를 포함한다.

[0019] 일부 실시예에서, 보안 어플리케이션(44)은 호스트 시스템(10)을 멀웨어로부터 보호하기 위하여, 후술하는 바와 같이 안티 멀웨어 작업을 수행하도록 구성된다. 도 2b의 예에서, 어플리케이션(44)의 인스턴스는 개별 VM(32a-b)에서 실행될 수 있고, 그러한 인스턴스 각각은 각각의 가상 머신을 보호하도록 구성된다. 보안 어플리케이션(44)은  자립형 프로그램(standalone program)일 수 있고, 또는 특히, 안티-멀웨어, 안티-스팸, 및 안티-스파이웨어 요소를 포함하는 소프트웨어 군(software suite)의 일부를 형성할 수 있다.

[0020] 도 3은 본 발명의 일부 실시예에 따른 호스트 시스템(10) 상에서 실행되는 소프트웨어 객체들의 계층도를 보여준다. 도 3은 가상화 환경에서 실행되도록 구성된 예시적 실시예를 보여준다. 도시된 실시예는 VM(32) 내 대신에 호스트 시스템(10)에서 직접 실행되도록 변경이 가능하다는 것은 본 기술분야의 통상의 기술자에게 명확할 수 있다. 도 3은 본 기술분야에서 레이어(layer) 또는 보호 링(protection ring)으로 또한 알려진 프로세서 권한 레벨의 관점으로부터 도시되었다. 일부 실시예에서, 각각의 그러한 레이어 또는 보호 링은, 각각의 프로세서 권한 레벨에서 실행되는 소프트웨어 객체가 실행할 수 있는 명령 세트로 특징지어진다. 소프트웨어 객체가 각각의 권한 레벨 내에서 허용되지 않는 명령을 실행하고자 할 때, 해당 시도는 예외(exception), 오류(falut) 또는 가상 머신 종료 이벤트(exit event)와 같은 프로세서 이벤트를 촉발할 수 있다. 일부 실시예에서, 권한 레벨들 사이에서의 전환(switching)은 전용 명령 세트를 통해서 이뤄질 수 있다. 그러한 명령들의 예로는 사용자 레벨에서 커널 레벨(kernel level)로 전환하는 SYSCALL/SYSENTER, 커널 레벨로부터 사용자 레벨로 전환하는 SYSRET/SYSEXIT, 사용자 또는 커널 레벨로부터 루트 레벨로 전환하는 VMCALL, 및 루트 레벨로부터 커널 또는 사용자 레벨로 전환하는 VMRESUME 등이 있다.

[0021] 운영 시스템(34)의 대부분의 요소들은 본 기술 분야에서 커널 레벨 또는 커널 모드(예를 들어서, 인텔 플랫폼에서 링 0)로 알려진 프로세서 권한 레벨에서 실행된다. 어플리케이션(42g)는 OS(34) 보다 낮은 프로레서 권한에서 실행된다(예를 들어서, 링 3 또는 사용자 모드). 가상화를 지원하는 실시예에서, 하이퍼바이저(30)는 루트 레벨 또는 루트 모드(예를 들어서, Intel® 플랫폼에서 링 -1 또는 VMXroot)로 또한 알려진 최상 권한 레벨에서 프로세서(12)를 컨트롤하고, 가상 머신(32)을 OS(34)와 어플리케이션(42g)과 같은 다른 소프트웨어 객체들에 노출시킨다.

[0022] 일부 실시예에서, 보안 어플리케이션(44)의 부분들이 사용자 레벨 프로세서 권한에서, 즉 어플리케이션(42d-e)과 동일한 레벨에서 실행될 수 있다. 예를 들어서, 이러한 부분들은 사용자에게 각각의 VM에서 탐지된 모든 멀웨어 또는 보안 위협들을 통지하고, 사용자가 가리키는 것으로부터의 입력, 예를 들어서 어플리케이션(44)을 위한 바람직한 구성 옵션(configuration option)을 수신하는 그래픽 유저 인터페이스를 포함할 수 있다. 사용자 레벨에서 실행되는 구성요소의 다른 예는 이하에서 상술하는 것처럼 작동하는 사용자 레벨 엔터티 평가자(50a)이다. 일부 실시예에서, 사용자 레벨 엔터티 평가자(50a) 부분은 보안 어플리케이션(44) 내에서 작동할 수 있고, 반면에 (후킹 모듈과 같은) 다른 부분은 어플리케이션(42g)과 같은 평가된 어플리케이션 내에서 작동할 수 있다. 어플리케이션(44)의 다른 부분들은 커널 권한 레벨에서 실행될 수 있다. 예를 들어서, 어플리케이션(44)은 안티-멀웨어 드라이버(36)와, 엔터티 관리 모듈(37)과, 스코어링 엔진(38)을 설치할 수 있고, 이들은 모두 커널 모드에서 작동한다. 드라이버(36)는, 예를 들어서 멀웨어 서명을 위하여 메모리를 스캔하기 위한 그리고/또는 OS(34) 상에서 실행되는 프로세스 및/또는 다른 소프트웨어 객체들의 멀웨어를 나타내는 행동을 탐지하

기 위한 기능(functionality)을 안티 멀웨어 어플리케이션(44)에 제공한다. 일부 실시예에서, 안티 멀웨어 드라이버(36)는 이하에서 설명하는 바와 같이 작동하는 커널 레벨 엔터티 평가자(50b)를 포함한다.

[0023] 일부 실시예에서, 엔터티 관리 모듈(37)은 호스트 시스템(10)(또는 VM (32)) 내에서 실행되는 소프트웨어 엔터티들의 집합을 관리한다. 일부 실시예에서, 상기 집합은 도면부호 55a-b와 같은 엔터티 평가 모듈에 의하여 멀웨어에 대하여 평가되는 모든 엔터티를 포함한다. 상기 집합을 관리하기 위하여, 모듈(37)은 이하의 모드 상세에서 보여지는 바와 같이, 엔터티 개시 및/또는 종료 이벤트들과 같은 라이프 사이클 이벤트들의 발생을 탐지하는 것에 응답으로 상기 집합으로부터 엔터티를 제거하거나 그리고/또는 추가할 수 있다. 모듈(37)은 부모 엔터티의 자식 엔터티(예를 들어서, 자식 프로세스)를 결정하는 것, 그리고/또는 선택된 엔터티가 라이브러리와 같은 소프트웨어 객체를 다른 엔터티에 삽입했는지 여부 또는 상기 선택된 엔터티가 다른 소프트웨어 엔터티에 의한 삽입 타겟인지 여부를 결정하는 것과 같은 엔터티간 관계들을 추가적으로 결정할 수 있다. 자식 엔터티는 부도 엔터티로 호칭되는 다른 실행 엔터티(executable entity)에 의하여 생성되는 실행 엔터티이고, 상기 자식 엔터티는 부모 엔터티와는 독립적으로 실행된다. 예시적인 자식 엔터티들은 예를 들어서 Windows® OS의 CreateProcess 함수를 통해서 생성되거나 또는 Linux®에서 포크 매커니즘(fork mechanism)을 통해서 생성된 자식 프로세스이다. 코드 삽입은 개별 프로세스의 원래 기능성(original functionality)을 변경하기 위하여, 기존 프로세스의 메모리 공간으로 DLL(dynamic-link library)과 같은 코드의 시퀀스를 도입하는 방법군을 가리키기 위하여 본 기술분야에서 사용되는 관용적 용어이다. 프로세스 개시 탐지 및/또는 코드 삽입 탐지와 같은 과업을 수행하기 위하여 모듈(37)은 특정 OS 함수를 호출하거나 후킹(hooking)하는 것과 같은 본 기술분야에서 알려진 임의의 방법을 채용할 수 있다. 예를 들어서, Windows® OS를 운영하는 시스템에서, 모듈(37)은 새로운 프로세스의 개시를 탐지하기 위하여 PsSetCreateProcessNotifyRoutine 콜백을 등록하거나, 그리고/또는 삽입된 코드의 실행을 탐지하기 위하여 CreateRemoteThread 함수를 후킹할 수 있다.

[0024] 도 4는 본 발명의 일부 실시예에 따른 엔터티 관리 모듈(37)에 의해서 수행되는 단계들의 예시적인 시퀀스를 보여준다. 단계(250-252)들의 시퀀스에서, 모듈(37)은 예를 들어서 상술한 방법을 이용하여 엔터티 라이프 사이클 이벤트를 중간차단한다. 그와 같은 이벤트가 발생할 때, 단계(254)는 개별 이벤트를 트리거링하는 엔터티를 식별한다. 단계(258)는 개별 엔터티의 고유 엔터티 식별 표시자(entity identification indicator, EID)를 결정하는 것을 포함할 수 있다. 그러한 표시자는 이하에서 추가적으로 보여지는 바와 같이 개별 엔터티를 스코어링하는 데 사용될 수 있다. 단계(256)는 이벤트가 새로운 엔터티(예를 들어서, 새로운 프로세스)의 개시를 포함하는지를 결정하고, 아니라면, 모듈(37)은 단계(260)로 나아간다. 상기 이벤트가 개시를 포함할 때, 단계(258)에서, 모듈(37)은 트리거링 엔터티를 평가된 엔터티들의 집합에 추가할 수 있다. 단계(260)는 이벤트가 자식 엔터티를 생성(spawning)하는 부모 엔터티를 포함하는지를 결정하는 것을 포함하고, 아니라면, 모듈(37)은 단계(264)로 나아갈 수 있다. 맞다면, 단계(262)에서, 모듈(37)은 개별 자식 엔터티를 평가된 엔터티들의 집합에 추가할 수 있다. 단계(262)는 자식 엔터티의 EID를 결정하는 것과, 트리거링 엔터티와 자식 엔터티 사이의 관계를 계통(filiation) (부모 -자식) 관계로 기록하는 것을 추가적으로 포함할 수 있다.

[0025] 일부 실시예에서, 단계(264)는 상기 이벤트가 코드의 삽입을 포함하는 지를 결정하고, 아니라면, 모듈(37)은 단계(268)로 나아갈 수 있다. 맞다면, 모듈(37)은 코드 삽입의 타겟 엔터티와 소스 엔터티를 식별할 수 있고, 상기 소스 엔터티는 타겟 엔터티로 코드를 삽입한다. 단계(266)에서, 모듈(37)은 소스 엔터티와 타겟 엔터티 사이의 코드 삽입 형태의 관계를 등록할 수 있다.

[0026] 단계(268)에서, 엔터티 관리 모듈(37)은 상기 이벤트가 트리거링 엔터티의 종료를 포함하는지 결정하고, 아니라면 모듈(37)은 단계(250)로 회귀한다. 일부 실시예에서, 엔터티는 개별 엔터티의 모든 구성요소가 실행을 종료할 때 종료된 것으로 간주된다. 예를 들어서, 개별 프로세스의 모든 쓰레드가 실행을 종료할 때 프로세스는 종료된다. 상기 이벤트가 트리거링 엔터티의 종료를 포함했을 때, 단계(270)에서, 모듈(37)은 트리거링 엔터티의 후손 엔터티(descendant entity, 파생 엔터티)의 세트를 결정할 수 있다. 일부 실시예에서, 트리거링 엔터티의 후손 엔터티들은 복수의 세대에 걸쳐서, 자식 엔터티의 자식 엔터티뿐만 아니라 개별 엔터티의 자식 엔터티를 포함한다. 일부 실시예에서, 후손 엔터티들은 회귀적으로 타겟팅된 엔터티들에 의하여 타게팅된 엔터티뿐만 아니라 트리거링 엔터티에 의하여 삽입된 코드를 포함하는 타겟 엔터티를 포함할 수 있다. 단계(272)에서, 모듈(37)은 후손 엔터티들의 세트의 모든 엔터티가 종료되었는지를 결정할 수 있고, 아니라면, 단계(250)로 회귀하여 실행된다. 모든 후손들이 종료되었을 때, 단계(274)에서 엔터티 관리 모듈(37)은 평가된 엔터티들의 집합으로부터 트리거링 엔터티를 제거할 수 있다.

[0027] 일부 실시예에서, 스코어링 엔진(38)은 평가자(50a-b)와 같은 복수의 엔터티 평가자 모듈들로부터 평가된 소프트웨어 엔터티에 대하여 결정된 데이터를 수신하도록 구성되고, 또한 개별 엔터티가 개별 데이터에 따라서 악성

인지 여부를 결정하도록 구성된다. 일부 실시예에서, 스코어링 엔진(38)에 의하여 분석된 소프트웨어 엔터티들은 특히, 프로세스와 실행 쓰레드와 같은 실행 객체를 포함한다. 프로세스는 운영 시스템의 일부 또는 어플리케이션과 같은 컴퓨터 프로그램의 인스턴스이고, 적어도 실행 쓰레드와 상기 운영시스템에 의하여 실행 쓰레드에 할당된 가상 메모리의 섹션을 가지는 것을 특징으로 하고, 상기 개별 섹션은 실행 코드를 포함한다. 일부 실시예에서, 평가된 소프트웨어 엔터티들은 예를 들어서, 개별 쓰레드로부터, 개별 어플리케이션으로, 운영 시스템 및/또는 가상 머신들의 전체 인스턴스들로, 범위와 복잡성이 실질적으로 변화될 수 있다.

[0028] 도 5는 복수의 평가 표시자(52a-d)들을 수신하기 위한 예시적인 스코어링 엔진(38)을 도시하고, 여기서 개별 표시자(52a-d)는 엔터티 평가자에 의하여 결정된다. 도 5에서, 그러한 평가자들은 특히 사용자 레벨 엔터티 평가자(50a), 커널 레벨 엔터티 평가자(50b), 및 시스템 호출 평가자(50c)를 포함한다. 그러한 각각의 평가자 모듈은 다른 평가자들과 독립적으로 실행될 수 있고, 각각은 평가된 소프트웨어 엔터티의 복수의 특징적 엔터티 평가 표시자들를 결정할 수 있다. 하드웨어 가상화를 구현하는 시스템들에서, 도 5의 표시자(52a-c)와 같은 일부 평가 표시자들은 VM(32) 내에서 실행되는 요소들에 의하여 결정되고, 반면에 52d와 같은 다른 평가 표시자들은 VM(32) 밖에서(예를 들어서, 메모리 인트로스펙션 엔진(40)에 의하여) 실행되는 구성요소에 의하여 결정된다. 일부 실시예에서, 각 평가 표시자(52a-d)는 엔터티 식별 표시자를 포함하고, 엔진(38)이 개별 평가 표시자를 소프트웨어 엔터티에 특징적으로 연계시킬 수 있게 하며, 상기 평가 표시자는 상기 소프트웨어 엔터티에 대하여 결정된다.

[0029] 일부 평가 표시자들은 멀웨어를 나타낼 수 있고, 다시 말해서, 평가된 엔터티가 악성이라는 것을 나타낼 수 있다. 일부 평가 표시자들은 그 자체로는 멀웨어를 나타내는 것이 아닐 수 있으나, 다른 평가 표시자들과 결합할 때 악성을 나타낼 수 있다. 개별 평가 표시자(52a-d)는 특징적인 방법 및/또는 기준에 따라서 결정될 수 있다. 예시적인 평가 기준은, 평가된 엔터티가 특정 활동, 이를테면 디스크 파일에 쓰기, VM(32)의 시스템 레지스터 키를 편집하기 또는 피보호 소프트웨어 객체에 속하는 메모리 페이지에 쓰기를 수행하는지를 결정하는 것과 같은 행동적 기준을 포함할 수 있다. 다른 예시적인 기준은 평가된 엔터티에 속하는 메모리의 섹션이 멀웨어를 나타내는 서명을 포함하는지 여부를 결정하는 것을 포함할 수 있다.

[0030] 엔터티 평가자(50a-c)들의 작동을 설명하기 위하여, 도 6은 본 발명의 일부 실시예에 따른 소프트웨어 엔터티(70a-b)들의 세트의 예시적 실행 플로우를 보여준다. 단순화를 위하여, 선택된 엔터티(70a-b)들은 Windows® OS의 인스턴스에서 실행되는 프로세스들이다. 유사한 다이어그램들이 예를 들어서, Linux와 같은 다른 운영 시스템들을 위하여 제공될 수 있다. 실선 화살표들은 엔터티 평가자들이 없을 때(예를 들어서, 보안 어플리케이션(44)이 없을 때) 실행 흐름을 보여준다. 파선 화살표는 본 발명의 일부 실시예에 따라서 실행되는 엔터티 평가자(50a-c)들이 존재할 때의 흐름이 수정되는 을 보여줄 수 있다.

[0031] 프로세스(70a)는 다수의 DLL(dynamic linked library)(72a-c)을 로딩한다. 도 6의 예에서, DLL(72c)은 (악성 가능성이 있는) 프로세스(70b)에 의해서 프로세스(70a)내로 삽입된다. 프로세스(70a)(또는 프로세스의 로딩된 DLL 중 하나)가, 예를 들어서 디스크 파일에 무언가를 기록하거나, 레지스트리 키를 편집하는 것과 같은 몇 가지 시스템 기능을 요청하는 명령을 실행할 때, 각각의 명령은 KERNEL32.DLL 또는 NTDLL.DLL과 같은 사용자 모드 API를 호출한다. 도 6의 예에서, 각 사용자 모드 API 호출은 사용자 레벨 행동 필터(user-level behavioral filter,50a)에 의해서 중간차단되고 분석된다. 그와 같은 중간차단(interception)은 특히 DDL 삽입 또는 후킹(hooking)과 같은 방법에 의해서 행해질 수 있다. 후킹은 소프트웨어 요소들간에 전달되는 기능 호출, 메시지 또는 이벤트를 가로채는 방법을 위하여 본 기술분야에서 통용되는 용어이다. 후킹 방법의 일예는 명령 우회 실행(instruction redirecting execution)을 제2 함수에 삽입하여 타겟 함수(target function)의 엔트리 포인트를 변경하는 것을 포함한다. 그러한 후킹 다음에, 제2 함수가 타겟 함수 대신에, 또는 그 이전에 실행될 수 있다. 도 6의 예에서, 안티-멀웨어 드라이버(36)는 KERNEL32.DLL 또는 NTDLL.DLL의 특정 함수들로 연결(hook)되서 각 함수들이 필터(50a)로 실행되도록 우회 명령할 수 있다. 따라서, 필터(50a)는 프로세스(70a)가 후킹된 함수에 따라서 식별된 특정의 행동을 수행하는 것을 시도하는 것을 탐지할 수 있다. 필터(50a)가 그러한 행동을 탐지할 때, 필터(50)는 평가 표시자(52a)를 구축하고 표시자(52a)를 스코어링 엔진(38)에 전달할 수 있다(예를 들어서 도 5 참조).

[0032] 통상적인 실행 흐름도에서는, 엔터티(70a)에 의해서 호출되는 사용자 모드 API 함수는 운영 시스템의 커널로부터 서비스를 요청할 수 있다. 일부 실시예에서, 그러한 작업들은 x86 플랫폼에서 SYSCALL 및 SYSENTER와 같은 시스템 호출을 발급함으로써 수행된다. 도 6의 예에서, 그러한 시스템 호출들은 시스템 호출 평가자(50c)에 의해서 중간차단된다. 일부 실시예에서, 그러한 중간차단은 예를 들어서 프로세서(12)의 모델 특정 레지스터(model-specific register, MSR)에 저장된 값을 변경시킴으로써 시스템 호출 핸들러 루틴(system call handler

routine)을 수정하는 것을 포함할 수 있고, 이것은 필터(50c)로의 실행을 효과적으로 우회시킨다. 이와 같은 기술들은 MSR 후킹으로서 본 기술분야에서 알려져 있고 시스템 호출 평가자(50c)가 피평가 프로세스가 특정 시스템 호출을 수행하고자 시도하는 것을 탐지할 수 있게 할 수 있다. 그러한 시스템 호출이 중간차단되면 시스템 호출 필터(50c)는 엔터티 평가 표시자(52c)를 구축하고 표시자(52c)를 스코어링 엔진(38)에 전달할 수 있다.

[0033] 상기 시스템 호출에 이어서, 프로세스의 제어가 일반적으로 OS(34)의 커널에게 넘어간다. 일부 실시예에서, 커널 레벨 엔터티 평가자(50b)는 OS 커널의 특정 작업을 중간차단하고, 이에 따라서 피평가 프로세스가 악성일 수 있는 특정 작업을 수행시도하는지를 결정하도록 구성된다. 그러한 작동을 중간차단하기 위하여, 일부 실시예에서는 OS(34) 내에 구축되고 OS(34)에 의해서 노출되는 필터링 메커니즘 세트를 채용할 수 있다. 예를 들어서, 윈도우즈 OS에서는, FltRegisterFilter가 파일 생성/열기/쓰기/삭제와 같은 작업을 중간차단하기 위하여 사용될 수 있다. 다른 예에서, 평가자(50b)가 ObRegisterCallback을 이용해서 객체-핸들 작업(object-handle operation)을 생성 또는 복사(duplicate)하는 것을 중간차단할 수 있고, 또는 PsSetCreateProcessNotifyRoutine을 사용해서 새로운 프로세스의 생성을 차단할 수 있다. 또 다른 예에서, 레지스트리 키/값을 생성하고 설정하는 것과 같은 윈도우즈 레지스트리 작업들은 CmRegisterCallbackEx을 사용하여 중간차단될 수 있다. Linux®과 같은 다른 운영 시스템들을 위한 유사한 필터링 메커니즘이 본 기술분야에서 알려져 있다. 커널-모드 엔터티 평가자(50b)가 그러한 작업을 중간차단할 때, 평가자(50b)는 엔터티 평가 표시자(52b)를 구축하고 표시자(52b)를 스코어링 엔진(38)에 전달할 수 있다.

[0034] 엔터티 평가 표시자(52a-c)와 같은 데이터를 평가자(50a-c)로부터 스코어링 엔진(38)으로 전달하기 위하여, 통상의 기술자라면 임의의 프로세스간 통신 방법(inter-process communication method)을 채용할 수 있다. 예를 들어서, 사용자 모드 요소와 커널 모드 요소 사이의 통신을 위하여 평가자(50a-c)와 엔진(38)은 공유된 메모리 섹션을 사용하도록 구성될 수 있다. VM(32) 내에서 실행되는 구성요소와 개별 VM 밖에서 실행되는 구성요소 사이의 데이터 교환이 필요할 때, 그러한 통신은 가상화 기술에서 알려진 임의의 방법을 사용하여 실행될 수 있다. 예를 들어서 평가 표시자(52d)를 메모리 인트로스펙션 엔진(40)으로부터 스코어링 엔진(38)으로 전송하기 위하여, 일부 실시예들은 데이터가 개별 VM 밖으로부터 전송되는 엔진(38)으로의 신호로의 차단 삽입 메커니즘(interrupt injection mechanism to signal to engine)을 사용한다. 실제 데이터는 예를 들어서 상술한 공유된 메모리 섹션을 통해서 전달될 수 있다.

[0035] 도 7은 본 발명의 일부 실시예에 따라서 도 4-5의 메모리 인트로스펙션 엔진(40) 및/또는 평가자(50a-c)와 같은 엔터티 평가자에 의해서 수행된 단계들의 예시적 시퀀스를 보여준다. 단계(302-304)들의 시퀀스에서, 엔터티 평가자는 호스트 시스템(10) 및/또는 가상 머신(32) 내에서의 트리거 이벤트의 발생을 기다린다. 예시적인 트리거 이벤트는 특히, 특정한 프로세서 명령을 발행하거나, 저장 장치(20) 또는 네트워크 어댑터(22)와 같은 특정의 하드웨어 부분을 사용하려고 시도하거나, 보호된 메모리 페이지에 쓰기 시도를 하는 것과 같은 특정의 행동을 수행하는 소프트웨어 엔터티를 포함한다. 예를 들어서, 평가자(50c)를 위한 트리거 이벤트는 시스템 호출(예를 들어서, SYSENTER)을 발행하는 소트프웨어 엔터티를 포함할 수 있다. 평가자(50d)를 위한 트리거 이벤트의 다른 예는 UrlDownloadToFile API 함수를 호출하는 어플리케이션을 포함할 수 있다. 트리거 이벤트의 발생을 탐지하기 위하여, 개별 엔터티 평가자는 특히 코드 삽입과 MSR 후팅과 같은 본 기술분야에서 알려진 임의의 방법을 사용할 수 있다. 트리거 이벤트 중단차단의 일부 예들은 도 6과 관련하여 앞서 설명하였다.

[0036] 트리거 이벤트가 탐지될 때, 단계(306)에서 엔터티 평가자는 개별 트리거 이벤트를 유발하는 소프트웨어 엔터티(예를 들어서, 프로세스)를 식별할 수 있다. 일부 실시예에서, 엔터티 평가자는 현재 실행 중인 개별 프로세스 및/또는 쓰레드를 나타내기 위하여 OS(34)에 의하여 사용되는 데이터 구조로부터 소프트웨어 아이덴터티를 결정할 수 있다. 예를 들어서, 윈도우스에서, 개별 프로세서는, 특히 각 프로세스의 쓰레드(thread)들 각각에 대한 핸들(handle)과 OS(34)가 다수의 실행 프로세스로부터 각 프로세스를 식별할 수 있도록 하는 고유 프로세스 ID를 포함하는, 실행 프로세스 블록(executive process block)(EPROCESS)으로서 표현한다. 유사한 프로스세/쓰레드의 표현법(representation)이 Linux와 같은 다른 OS들을 위하여 이용가능하다.

[0037] 단계(308)에서, 엔터티 평가자는 개별 소프트웨어 엔터티에 의하여 수행되고 단계(302-304)들에서 중간차단되는 활동/이벤트의 종류의 표시자와 개별 소프트웨어 엔터티의 식별자(예를 들어서, 프로세스 ID)를 포함하는 평가 표시자를 형성할 수 있다. 일부 실시예에서, 엔터티 평가자는 중간차단된 트리거 이벤트의 파라미터로부터 개별 소프트웨어 엔터티의 행동 및/또는 활동의 유형을 결정할 수 있다. 작동의 예에서, 프로세스가 인터넷으로부터 파일을 다운받으려고 시도할 때, 사용자 레벨 엔터티 평가자(50a)는 그러한 시도를 중간차단할 수 있다. 어떤 프로세스가 그러한 활동을 수행하는지 식별하는 것 이외에 특히, 평가자(50a)는 또한 활동의 유형(파일 다운로드), 파일이 다운로드 되는 출처 IP 주소, 및 다운로드 되는 파일의 디스크 위치를 결정할 수 있다. 그러한 데

이터는 평가 표시자로 선택적으로 병합될 수 있고, 스코어링 엔진(38)이 엔터티 X가 활동 Y를 수행했다는 것을 파라미터 Z로 결정할 수 있도록 한다. 단계(310)에서, 엔터티 평가자는 평가 표시자를 스코어링 엔진(38)으로 전송한다.

[0038] 일부 실시예에서, 스코어링 엔진(38) 및/또는 엔터티 관리 모듈(37)은 호스트 시스템(10)(또는 VM(32))에서 실행되는 프로세스와 쓰레드와 같은 평가된 소프트웨어 엔터티들의 중앙화된 지식베이스(centralized knowledgebase)를 관리한다. 도 8은 평가된 엔터티(70c-e)의 세트를 보여주고, 이들 각각은 예시적인 엔터티 스코어링 객체(exemplary scoring object, ESO)(74a-c)로 각각 표현된다. 각각의 ESO는 복수의 데이터 필드를 포함하고, 이들중 일부는 도 8에서 나타내어져 있다. 그러한 필드들은 고유 엔터티 식별자(unique entity identifier, EID)(76a), 복수의 평가 스코어(76b)들, 및 총합 스코어(76d)를 포함할 수 있다. 일부 실시예에서, 평가 스코어(76b)들은 개별 엔터티 평가자들로부터 수신된 평가 표시자(52a-d)에 따라서 엔진(38)에 의하여 결정된다. 그러한 스코어들의 각각은 표시자(76c)에 의하여 식별된 평가 기준에 따라서 결정될 수 있다. 일부 실시예에서, 평가 스코어(76b)는 평가 기준(76c)과 일대일 대응관계를 가지고, 그래서 각 스코어는 개별 기준에 따라서 부여된다. 예를 들어서, 특정 기준 $C_k$는 평가된 엔터티가 인터넷과 같은 컴퓨터 네트워크로부터 파일을 다운로드하는지 여부를 결정하는 것을 포함할 수 있다. 그러한 일예에서, 개별 스코어 $S_k$는 평가된 엔터티가 다운로드를 실행하는 경우에만 부여될 수 있다.

[0039] 일부 실시예에서, ESO(74a)는 플래그(76e) 세트를 추가적으로 포함할 수 있다. 일부 플래그(76e)들은 이진수 표시자(예를 들어서, 0/1, yes/no)일 수 있다. 그러한 일예에서, 플래그는 개별 피평가 엔터티 $E_1$이 특정 평가 기준을 만족하는지를 나타낸다(예를 들어서, $E_1$이 인터넷으로부터 실행가능한 파일을 다운로드 했는지 여부, $E_1$이 커맨드 라인 모드에서 작동했는지 등). 다른 예시적인 플래그는 엔터티 $E_1$의 분류를 나타내는데, 예를 들어서, $E_1$이 트로잔 멀웨어, 브라우저 객체, PDF 읽기 어플리케이션 등과 같은 특정 카테고리의 객체에 속하는지를 나타내는 표시자이다. 플래그의 예시적인 사용은 엔터티 $E_1$의 평가 스코어 $S_i$의 업데이트가 $E_1$의 다른 평가 스코어 $S_j$의 업데이트를 트리거링 하는 상황이다(이하 참조). 플래그들은 그러한 상호 업데이트 메커니즘(co-update mechanism)을 켜고 끄는 데 사용될 수 있다. 예를 들어서, $E_1$이 평가 기준 $C_i$(예를 들어서, 엔터티가 특정 활동을 수행하는지 여부)를 만족할 때, 엔터티 $E_1$이 또한 기준 $C_j$를 만족할 것이라는 것이 알려져 있을 수 있다. 따라서, 관계($C_i$, $C_j$>를 나타내는 플래그 $F_1$은 엔터티 $E_1$을 위하여 설정될 수 있고, 스코어 $S_i$가 업데이트될 때 스코어 $S_j$의 업데이트를 트리거링한다.

[0040] ESO(74a)는 개별 엔터티가 현재 활성 또는 종료되었는지를 가리키는 종료 표시자(76f)를 또한 포함할 수 있다. 그러한 종료 표시자는 스코어링 엔진(38)으로 하여금 종료된 엔터티들의 스코어들의 기록을 보관 및/또는 업데이트하도록 할 수 있다. ESO(74a)는 개별 엔터티 $E_1$과 관련된 소프트웨어 엔터티들의 식별자 세트를 추가적으로 포함할 수 있다. 그러한 관련된 소프트웨어 엔터티들의 예들은 식별자(76g)에 의하여 표시되는 $E_1$의 부모 엔터티와, 식별자(76h)에 의하여 표시되는 $E_1$의 자식 엔터티의 세트를 포함할 수 있다. ESO(74a)는 어떤 $E_1$으로 삽입된 코드를 가지는지를 소프트웨어 엔터티들을 식별하는 삽입 타겟 엔터티의 표시자(아이템 76j)들 세트와, $E_1$으로 삽입된 코드를 가지는 소프트웨어 엔터티를 식별하는 삽입 소스 엔터티들의 표시자(아이템 76k)들의 세트를 추가적으로 포함할 수 있다.

[0041] 일부 실시예에서, 평가된 소프트웨어 엔터티들의 스코어링은 스코어 값의 세트와 또한 추가적인 파라미터들에 따라서 수행된다. 도 9는 스코어 값 세트가 아이템(80)으로 표시되는 그러한 데이터를 보여준다. 스코어 값들은 이들의 대응하는 평가 기준 $C_1...C_n$에 의하여 인덱싱된다. 그러한 값 각각은 예를 들어서 피평가 엔터티가 개별 평가 기준(예를 들어서, 피평가 엔터티가 인터넷으로 파일을 다운로드 했는지, MS Word® 에 쓰기를 했는지 등)을 만족할 때 받는 소정 숫자의 포인트를 나타낼 수 있다.

[0042] 스코어링을 컨트롤하는 예시적인 파라미터들은 초기화 가중치(initialization weight)(82a) 세트, 전파 가중치(82b)(propagation weight) 세트, 새로운 인스턴스 가중치(new instance weight)(82c) 세트, 예외 가중치(82d)(exception weight) 세트, 및 플래그 유도 가중치(82e)(flag-induced weight) 세트를 포함한다. 가중치(82a-e)들은 평가 기준 $C_1...C_n$에 의하여 인덱싱된다. 일부 유형의 가중치는 평가 기준과 일대일 대응관계에 있어서, 각 $C_i$에 대하여 하나의 가중치 값 $w_i$가 있다. 다른 유형의 가중치들은 평가 기준과 1:다 관계에 있다. 그

러한 일예는 도 9의 예외 가중치(82d)이고, 특정 평가 기준 $C_i$에 대응하는 복수의 가중치($w_{ij}$)가 있을 수 있다. 가중치들은 도 9의 예에서 설명하는 바와 같이 엔터티들의 클라스 또는 카테고리에 의하여 그룹핑될 수 있다. 예를 들어서, 워드 프로세싱 어플리케이션(예를 들어서, MS Word®)에 적용가능한 제1 가중치 값, 웹 브라우저(예를 들어서, Firefox® 및 MS Internet Expolrer®)에 적용가능한 제2 가중치 값(제1 가중치 값과 다를 수 있음), 파일 관리 어플리케이션(예를 들어서, Windows Explorer®)에 적용 가능한 제3 가중치 값이 있을 수 있다. 다른 카테고리들의 엔터티들 사이에서의 차별화가 유용할 수 있는데, 왜냐하면 일부 평가 기준이 어느 일 카테고리의 엔터티들에 대해서 다른 것들에 대해서보다 더욱 멀웨어를 나타낼 수 있기 때문이다. 보다 일반적으로, 각 스코어링 가중치는 투플(tuple) <$C_i$,$E_k$...>에 의하여 인덱싱될 수 있는데, $C_i$는 특정 평가 기준을 표시하고, $E_k$는 특정 피평가 엔터티를 표시한다. 스코어링 가중치(82a-e)를 저장하고 이에 접근하기 위한 실제 데이터 포맷은 실시예들 사이에서 변화될 수 있다. 가중치(82a-e)는 특히 행렬, 리스트, 관계 데이터베이스(relational databases, RDB) 또는 XML(extensible markup language ) 구조로서 저장될 수 있다. 스코어링을 위한 예시적인 가중치 사용은 이하에서 설명된다.

[0043] 스코어 값(80) 및/또는 가중치는 예를 들어서 인간 운영자에 의하여 사전에 결정될 수 있다. 일부 실시예에서, 그러한 값들은 시간에 따라서 변화될 수 있고, 멀웨어 탐지를 최적화하기 위하여 조정될 수 있다. 업데이트된 스코어 값들 및/또는 가중치 값들은 보안 서버로부터 주기적 및/또는 요청시(on-demand) 소프트웨어 업데이트로서 호스트 시스템(10)에 전달될 수 있다(도 10-11과 관련하여 이하 참조).

[0044] 도 10은 본 발명의 일부 실시예에 따른 스코어링 엔진(38)에 의해서 실행되는 단계들의 예시적인 시퀀스를 보여준다. 단계(302)에서, 엔진(38)은 엔터티 평가자, 예를 들어서 도 5의 평가자(50a-c)들 중에서 하나로부터 엔터티 평가 표시자를 수신한다. 하드웨어 가상화를 구현하는 일부 실시예에서, 엔진(38)은 개별 가상 머신(예를 들어서 도 5의 메모리 인트로스펙션 엔진(40))의 밖에서 실행되는 구성요소로부터 개별 엔터티 평가 표시자를 수신할 수 있다. 단계(304)에서, 엔진(38)은 예를 들어서, 개별 평가 표시자에 매입된 엔터티 ID에 따라서, 개별 엔터티 평가 표시자가 어떤 소프트웨어 엔터티에 대하여 결정되었는지 식별할 수 있다(도 7과 관련하여 상술 내용 참조).

[0045] 다음으로, 스코어링 엔진(38)은 단계(304)에서 식별된 엔터티 E에 대해서 뿐만 아니라, E와 관련된 다른 엔터티들에 대해서도, 단계(318-332)들의 블록을 수행한다. 그러한 관련된 엔터티들은 특히, E의 부모 및 자식 엔터티들, E가 삽입 코드를 가지는 삽입 타겟 엔터들, 및 E로 삽입된 코드를 가지는 삽입 소스 엔터티들을 포함할 수 있다. 이와 같은 방식으로, 엔진(38)이 (예를 들어서 엔터티 E가 특정 활동을 수행한다는 것을 표시하는) 평가 표시자를 수신하는 매번, 블록(318-332)은 여러 차례 실행될 수 있고, 엔터티 E의 평가 스코어들뿐만 아니라 E와 관련된 엔터티들의 평가 스코어들로 업데이트한다. 일부 실시예에서, 블록(318-332)은 E를 위하여 한번 실행되고 E와 관련된 각각의 엔터티 $E^*$를 위하여 한번 실행된다. 선택적 실시예에서, 블록(318-332)은 반복적으로, 일부 수렴 기준(convergence criterion)이 만족될 때까지 실행된다. 예시적 수렴 기준은 E 및/또는 $E^*$의 평가 스코어들이 블록(318-332)의 연속적 실행 사이에서 변경되는지를 검증하는 것과, 그러한 변경이 없을 때 나가기(exiting)를 하는 것을 포함한다. 도 10의 예시적인 알고리즘에서, 변수 X는 스코어 업데이트를 현재 진행중인 엔터티를 가리키는 데 사용된다. 단계(316)에서, X는 단계(304)에서 식별된 엔터티 X로 설정된다.

[0046] 단계(318)에서, 엔진(38)은 엔터티 X(예를 들어서, 도 8의 엔터티들 76b)의 평가 스코어를 업데이트 한다. 일부 실시예에서, 평가 스코어를 업데이트하는 것은 개별 평가 스코어의 기록된 값을 새로운 값으로 대체하는 것을 포함한다:

*수학식 1*

[0047] $$S_k^{(x)} \rightarrow S_k^{(x)} + \triangle S_k$$

[0048] 여기에서, $S_k^{(x)}$는 평가 기준 $C_k$에 따라서 엔터티 X에 대하여 결정된 평가 스코어를 나타내고, $\triangle S_k$는 + 또는 - 일 수 있는 증분을 나타낸다(일부 실시예에서 평가 스코어는 업데이트 시 감소할 수 있다).

[0049] 일부 실시예에서, 스코어 증분 $\triangle S_k$은 단계(312)에서 수신된 평가 표시자에 따라서 스코어링 엔진(38)에 의하

여 결정된다. 개별 표시자는 스코어를 포함할 수 있고 그리고/또는 개별 표시자를 결정하는 데 사용된 평가 기준을 표시할 수 있다. 일부 실시예에서, 스코어링 엔진(38)은 예를 들어서, 개별 평가 기준 $C_k$(도 9 참조)에 대응하는 스코어 값(80)에 따라서 스코어 증분 $\triangle S_k$을 결정한다.

### 수학식 2

[0050] $$\triangle S_k = V_k$$

[0051] 여기서 $V_k$는 기준 $C_k$에 할당된 스코어 값(80)을 나타낸다. 그러한 일예에서, 기준 $C_k$는 엔터티 X가 네트워크로부터 객체를 다운로드했는지를 결정하는 것을 포함하고, $V_k$=20이며, 평가 스코어 $S_k^{(x)}$는 엔터티 X가 다운로드를 수행할 때마다 20 포인트씩 증가될 것이다. 일부 실시예에서, $\triangle S_k = \varepsilon V_k$인데, ε는 이진수 예외 가중치(예를 들어서, 도 9의 아이템 82d 참조)이고 스코어 $S_k$가 평가된 엔터티의 서브세트(subset)에 대해서만 업데이트되도록 한다. 그러한 예외 가중치는 예를 들어서 여러 유형의 평가된 엔터티들 사이를 구별하는 데 유용하다. 예를 들어서, 브라우저는 멀웨어에 대한 드높은 의심 없이 무제한 개수의 IP 주소에 접근하도록 허용되어야 한다. 인터넷 접근을 탐지하는 것을 포함한 평가 기준은 브라우저 유형의 엔터티들에 대하여는 예외 가중치를 0으로 설정함으로써, 다른 한편으로는 다른 유형의 엔터티들에 대해서는 이것을 활성( ε=1)으로 유지함으로써 브라우저 객체에 대하여 효과적으로 스위치 오프를 할 수 있다.

[0052] 일부 실시예에서, 엔터티 X의 평가 스코어를 업데이트 하는 데 사용되는 스코어 증분 $\triangle S_k$는 X와 관련된 엔터티 $X^*$에 대하여 결정된 평가 스코어에 따라서 결정되는데, 즉 스코어들은 자식으로부터 부모로, 또는 삽입 타겟으로부터 삽입의 소스로와 같이 하나의 엔터티로부터 관련 엔터티로 전파될 수 있다. 그러한 일예에서, 자식 프로세스에 의해서 수행된 활동은 상기 활동(자식 프로세스)을 수행한 엔터티의 스코어뿐만 아니라 개별 자식 프로세스의 부모 프로세스의 스코어의 업데이트도 트리거링할 수 있다. 그러한 스코어 업데이트는 아래 식에 따라서 스코어 증분을 연산하는 것을 포함할 수 있다.

### 수학식 3

[0053] $$\triangle S_k = w_k S_k^{(X')}$$

[0054] 여기에서 $w_k$는 엔터티 $X^*$의 스코어가 엔터티 X의 스코어에 영향을 미치는 강도를 나타내는 수치적인, 기준 특정 가중치를 말한다. 가중치 $w_k$는 전파 가중치(82b)를 포함할 수 있다. 일부 실시예들은 다양한 그러한 전파 가중치들 사이를 구별하고, 예를 들어서, 자식 엔터티로부터 부모 엔터티로 스코어를 전파하는 데 사용되는 가중치는 부모 엔터티로부터 자식 엔터티로 스코어를 전파하는데 사용되는 가중치와 값이 다를 수 있다. 유사하게, 자식 엔터티로부터 부모 엔터티로 스코어를 전파하는 데 사용되는 가중치는 코드 삽입을 위하여 타게팅된 엔터티로부터 코드 삽입을 수행하는 엔터티로 스코어를 전파하는데 사용되는 가중치와 값이 다를 수 있다. 일부 실시예에서, 스코어들은 활성 엔터티로부터 종료된 엔터티로 전파할 수 있다. 예를 들어서, 자식 프로세스의 활동은 부모 프로세스가 종료될때에도 부모 프로세스의 스코어를 증가시킬 수 있다.

[0055] 일부 실시예에서, 수학식 3의 엔터티 $X^*$는 엔터티 X의 다른 인스턴스이다. 예를 들어서, X와 $X^*$는 동시에 실행되는 동일한 프로세스 또는 쓰레드의 복사물일 수 있다. 그러한 경우에서, 가중치 $w_k$는 새로운 인스턴스 가중치(예를 들어서, 도 9의 아이템 82c) 또는 초기화 가중치(예를 들어서, 아이템 82a)일 수 있다. 일부 실시예에서, 엔터티 X의 새로운 인스턴스 X'가 개시될 때, 엔진(38)은 X로부터 X'로 스코어를 전파하기 위하여 새로운 인스턴스 가중치 $w_k$를 사용하여 기존 엔터티 X의 일부 또는 모든 평가 스코어들을 업데이트 할 수 있다. 유사하게, X'가 개시될 때, 엔진(38)은 이미 실행되고 있는 엔터티 X로부터 새로운 엔터티 X'로 스코어를 전파하기 위하여

초기화 가중치 $w_k$를 사용하여 X'의 일부 또는 모든 평가 스코어들을 업데이트할 수 있다.

[0056] 일부 실시예에서, 평가 스코어 $S_k$를 업데이트 하는 것은 개별 엔터티의 특징적 평가 스코어 $S_m$의 업데이트를 트리거링할 수 있다. 예를 들어서, 다음과 같다.

[0057] [수학식 4]

[0058] $S_k^{(X)} \rightarrow S_k^{(X)} + V_k$가 $S_m^{(X)} \rightarrow S_m^{(X)} + F^{(X)} f_{km} V_m$, 트리거링한다.

[0059] 여기서 $F^{(X)}$는 엔터티 X(예를 들어서, 도 8의 아이템 76e 참조)에 대한 플래그 세트이고, 상기 플래그는 평가 기준 $C_k$와 $C_m$ 사이의 연결을 나타내고, $f_{km}$은 플래그 유도 가중치(예를 들어서, 도 9의 아이템 82e 참조)이고 $S_k$의 업데이트가 엔터티 X의 $S_m$의 업데이트에 영향을 미치는 강도를 나타낸다.

[0060] 단계(320)에서, 스코어링 엔진(38)은 단계(312)에서 수신된 평가 표시자에 따라서 엔터티 X의 플래그(도 8과 관련하여 플래그에 대한 상기 기술 참조)를 업데이트할 수 있다. 플래그는 수학식 4와 관련하여 앞서 설명한 바와 같이 스코어 상호 업데이팅 메커니즘을 활성화 및/또는 비활성화하도록 설정될 수 있다. 그러한 일예에서, 평가된 엔터티는 평가 표시자(단계 312)에 따라서 웹 브라우저 어플리케이션인 것으로 식별될 수 있다. 그와 같은 식별은 인터넷으로부터 장차 다운로드하는 것에 대해 개별 엔터티를 스코어링 하지 않도록 스코어링 엔진(38)에게 표시해야한다. 이것은 개별 엔터티에 대하여 특정 플래그 F의 값을 0으로 설정함으로써 달성될 수 있는데, 플래그 F는 엔터티가 인터넷으로부터 객체를 다운로드할 때 개별 엔터티의 평가 스코어를 업데이트하도록 스코어링 엔진(38)에게 표시한다.

[0061] 단계(322)에서, 스코어링 엔진(38)은 개별 프로세스에 대하여 결정된 개별 평가 스코어를 예를 들어서 다음의 합계로 합산함으로써 엔터티(X)의 총합 스코어를 결정할 수 있다.

*수학식 5*

[0062] $$A^{(X)} = \bigsqcup_k S_k^{(X)}$$

[0063] 단계(324)에서, 엔진(38)은 총합 스코어와 소정의 임계값을 비교할 수 있다. 총합 스코어가 입계값을 초과하지 않을 때, 스코어링 엔진(38)은 후술하는 단계(326)로 진행할 수 있다. 일부 실시예에서, 임계값은 사용자 입력에 따라서 결정된 값으로 설정될 수 있다. 그러한 임계값의 수치들은 개별 사용자의 보안 요청사항(security preference)을 반영할 수 있다. 예를 들어서, 사용자가 엄격한 보안을 선택할 때, 임계값은 상대적으로 낮은 값으로 설정될 수 있다. 사용자가 보다 완화된 보안 설정을 선호할 때, 임계값은 상대적으로 높은 값으로 설정될 수 있다. 일부 실시예에서, 임계값 수치는 도 10-11과 관련하여 후술하는 바와 같이 원격 보안 서버로부터 수신될 수 있다.

[0064] 일부 실시예에서, 단계(322-324)에서, 스코어링 엔진(38)은 복수의 총합 스코어들을 결정할 수 있고, 각 총합 스코어를 (가능한 특징적인) 임계값과 비교할 수 있다. 그러한 각각의 총합 스코어는 평가 스코어들의 특징적인 서브세트에 따라서 결정될 수 있다. 예시적인 실시예에서, 그러한 스코어들의 각각의 서브세트와 이들의 대응하는 평가 기준의 서브세트는 멀웨어(예를 들어서, 트로잔, 루트킷 등)의 특정 군과 유형을 나타낼 수 있다. 이것은 엔진(38)으로 하여금 탐지된 멀웨어의 분류를 수행하도록 할 수 있다. 다른 실시예에서, 스코어링 엔진(38)은 다양한 악성의 정도에 따라서(예를 들어서, 깨끗함, 의심됨, 위험함 및 심각함) 실행 엔터티들을 분류하기 위하여 복수의 임계값 수치를 채용한다.

[0065] 총합 스코어가 임계값을 초과할 때, 단계(326)에서 엔진(38)은 평가된 프로세스가 악성이라고 판단할 수 있고, 안티 멀웨어 작동을 수행할 수 있다. 일부 실시예에서, 그러한 안티 멀웨어 작동은 특히 평가된 프로세스를 종료하는 것, 평가된 프로세스를 격리하는 것, 평가된 프로세스의 (파일 또는 메모리 섹션과 같은) 리소스를 제거하거나 불능화하는 것을 포함할 수 있다. 일부 실시예에서, 안티 멀웨어 작동은 예를 들어서, 네트워크 어댑터(22)를 통해서 호스트 시스템(10)에 연결된 컴퓨터 네트워크 상에서 시스템 관리자에게 메시지를 보냄으로써, 호스트 시스템(10)의 사용자에게 경고하는 것, 그리고/또는 시스템 관리자에게 경고하는 것을 추가적으로 포함

할 수 있다. 일부 실시예에서, 안티 멀웨어 작동은 도 10-11과 관련하여 후술하는 바와 같이 원격 보안 서버에 보안 리포트를 전송하는 것을 또한 포함할 수 있다.

[0066] 단계(328-330)의 시퀀스에서, 엔진(38)은 X와 관련된 엔터티 $X^*$를 식별할 수 있고, 이 때 $X^*$의 스코어들은 X의 현재 스코어 업데이트들 다음에 업데이팅이 필요하다. 예를 들어서, $X^*$는 X의 부모 또는 자식 엔터티일 수 있다. 일부 실시예에서, 엔터티 $X^*$는 엔터티 X(예를 들어서, 도 8 참조)의 ESO의 필드(76g-k)에 따라서 식별될 수 있다. 그러한 엔터티 $X^*$가 존재하지 않을 때, 또는 모든 그러한 엔터티 $X^*$가 이미 스코어 업데이트를 위하여 고려되었을 때, 엔진(38)은 단계(312)로 회귀한다. 적어도 엔터티 $X^*$가 있을 때, 단계(332)에서 스코어링 엔진은 $X^*$를 현재의 엔터티로 만들고 단계(318)로 회귀한다.

[0067] 도 3-4에 도시된 예시적인 스코어링 엔진(38)은 OS 프로세서 권한 레벨(예를 들어서, 커널 모드)에서 VM(32) 내에서 작동한다. 선택적 실시예에서, 스코어링 엔진(38)은 하이퍼바이저(30)의 프로세서 권한 레벨에서, 사용자 모드에서 VM(32) 내에서, 또는 심지어 VM(32) 밖에서 실행될 수 있다.

[0068] 일부 실시예에서, 인트로스펙션 엔진(40)은 하이퍼바이저(30)와 실질적으로 동일한 권한 레벨에서 실행되고, VM(32)(도 3)과 같은 가상 머신의 인트로스펙션을 수행하도록 구성된다. VM 또는 개별 VM에서 실행되는 소프트웨어 엔터티의 인트로스펙션은, 특히, 소프트웨어 엔터티의 행동을 분석하는 것, 그러한 엔터티들의 메모리 주소를 결정 및/또는 이에 접근하는 것, 그러한 주소에 위치된 메모리의 콘텐트로 특정 프로세스가 액세스하는 것을 제한하는 것, 그러한 콘텐트를 분석하는 것, 및 개별 엔터티의 평가 표시자(예를 들어서, 도 5의 표시자 52d)를 결정하는 것을 포함할 수 있다.

[0069] 일부 실시예에서, 호스트 시스템(10)은 원격 보안 서버와, 멀웨어 탐지 이벤트에 대한 상세와 같은 보안 정보를 교환하도록 구성될 수 있다. 도 11은 그러한 구성의 예를 보여주는데, 상술한 시스템(10)과 같은 복수의 호스트 시스템(10a-c)이 컴퓨터 네트워크(26)를 통하여 보안 서버(110)에 연결된다. 예시적 실시예에서, 호스트 시스템(10a-c)은 기업의 근로자에 의하여 사용되는 개별 컴퓨터들이고, 한편 보안 서버(110)는 시스템(10a-c)에서 발생하는 보안 이벤트들 또는 멀웨어 위협을 모니터링 하기 위하여 개별 기업의 네트워크 관리자에 의하셔 설정된 컴퓨터 시스템을 포함할 수 있다. 다른 실시예에서, 예를 들어서, 각 호스트 시스템(10a-c)이 수십 또는 수백의 가상 머신들을 호스팅하는 서버인 IAAS(Infrastructure-as-a-service) 시스템에서, 보안 서버(110)는 중앙 위치로부터 모든 그러한 VM들에 대한 안티 멀웨어 작동을 관리하도록 구성된 컴퓨터 시스템을 포함할 수 있다. 또 다른 실시예에서, 보안 서버(110)는 안티 멀웨어 소프트웨어의 제공자(예를 들어서, 특히, 보안 어플리케이션(44)의 제공자)에 의해서 네트워크(26) 주변의 여러 시스템에서 탐지된 멀웨어에 대한 통계적 및/또는 행동적 정보를 수신하도록 구성된 컴퓨터 시스템을 포함할 수 있다. 네트워크(26)는 인터넷과 같은 광역 네트워크를 포함할 수 있고, 반면에 네트워크(26)의 일부들은 LAN(local area network)을 포함할 수 있다.

[0070] 도 12는 도 11에 도시된 바와 같은 실시예에서 호스트 시스템(10)과 보안 서버(110) 사이의 예시적 데이터 교환을 보여준다. 호스트 시스템(10)은 서버(110)에 보안 리포트(80)를 보내도록 구성될 수 있고, 서버(110)로부터 보안 설정(82) 세트를 수신하도록 구성될 수 있다. 일부 실시예에서, 보안 리포트(80)는 특히 엔터티 평가 표시자(52a-d) 및/또는 호스트 시스템(10)에서 실행되는 엔터티 평가자(50a-c 및/또는 40)에 의해서 결정된 스코어, 및/또는 스코어링 엔진(38)에 의해서 결정된 총합 스코어를 포함한다. 보안 리포트(80)는 또한 각 시스템(10)과 피평가 엔터티를 식별하는 데이터(예를 들어서, 엔터티 ID, 이름, 경로, 해쉬, 다른 종류의 엔터티 식별자)뿐만 아니라 엔터티 평가 표시자/스코어를 이것과 관련되어 결정되는 호스트 시스템과 엔터티와 연결하는 표시자를 포함할 수 있다. 일부 실시예에서, 리포트(80)는 호스트 시스템(10)에서 실행되는 엔터티들에 대한 통계 데이터 및/또는 행동 데이터를 추가적으로 포함할 수 있다. 시스템(10)은 멀웨어의 탐지 시에, 그리고/또는 스케줄에 따라서(예를 들어서 매 수분, 매 시간 등) 리포트(80)를 보내도록 구성될 수 있다.

[0071] 일부 실시예에서, 보안 설정(82)은 엔터티 평가자의 작업 파라미터(operational parameter)(예를 들어서, 도 5의 필터(50a-c)들의 파라미터들), 및/또는 스코어링 엔진(38)의 파라미터들을 포함할 수 있다. 엔진(38)의 예시적 파라미터는 특히, 피평가 프로세스가 악성인지 여부를 결정하기 위한 임계값 뿐만 아니라, 스코어 값(80)과 가중치(82a-e)를 포함할 수 있다.

[0072] 일부 실시예에서, 서버(110)는 멀웨어 탐지 성능을 최대화하기 위하여, 예를 들어서 긍정 오류(false positive)를 감소시키면서 탐지율을 증가시키기 위하여 위와 같은 파라미터들을 동적으로 조정하기 위한 최적화 알고리즘을 작동시킨다. 최적화 알고리즘은 다양한 엔터티 평가자들에 의하여 스코어링 엔진(38)에 보고된 엔터티 평

가 표시자/스코어들을 포함하여, 다수의 호스트 시스템(10a-c)들에서 실행되는 여러 엔터티에 대한 통계적 데이터 및/또는 행동 데이터를 수신할 수 있고, 파라미터들을 위한 최적값을 결정할 수 있다. 다음으로 이들 값들은 네트워크(26)를 통해서 각 호스트 시스템으로 전송된다.

[0073] 최적화의 그러한 일예에서, 스코어 값(80)을 변경하는 것이 서로에 대하여 각 평가 기준의 관련성을 효과적으로 변경할 수 있다. 멀웨어 위협은 통상적으로 연달아 발생하고, 굉장히 많은 전세계의 컴퓨터 시스템이 짧은 시간 간격에서 동일한 멀웨어 에이전트에 영향 받는다. 다수의 호스트 시스템으로부터 실시간으로 보안 리포트(80)를 수신함으로써, 보안 서버(110)는 현재의 멀웨어 위협에 대하여 최신의 상태가 될 수 있고, 최적의 보안 설정(82), 예를 들어서 현재의 멀웨어 위협을 탐지하는 데 최적화된 스코어 값(80)들의 세트를 포함하는 설정(82)을 각 호스트 시스템에 즉각적으로 전달할 수 있다.

[0074] 상술한 예시적 시스템과 방법은 바이러스, 트로잔, 및 스파이웨어와 같은 멀웨어로부터 컴퓨터 시스템과 같은 호스트 시스템을 보호할 수 있게 한다. 호스트 시스템에서 현재 실행되는 프로세스와 쓰레드와 같은 복수의 실행가능한 엔터티들 각각에 대하여, 스코어링 엔진은 복수의 평가 스코어들을 기록하고, 각 스코어는 특징적인 평가 기준에 따라서 결정된다. 일부 실시예에서, 평가된 소프트웨어 엔터티들은 범위와 복잡성에서, 예를 들어서, 개별 실행 쓰레드로부터, 개별 어플리케이션으로, 운영 시스템 및/또는 가상 머신의 전체 인스턴스로, 실질적으로 변화될 수 있다.

[0075] 모니터링된 엔터티가 평가 기준(예를 들어서, 활동을 수행)을 만족할 때마다, 엔터티의 개별 스코어는 업데이트된다. 타겟 엔터티의 스코어를 업데이트하는 것은 타겟 엔터티와 관련된 다른 엔터티들의 스코어 업데이트를 트리거링할 수 있다. 그러한 관련 엔터티들은, 특히, 타겟 엔터티의 자식, 타겟 엔터티의 부모, 타겟 엔터티가 주입 코드를 가지게 되는 엔터티(entities into which the target entity has injected code), 및 타겟 엔터티로 주입된 코드를 가지는 엔터티(entities which have injected code into the target entity)를 포함한다.

[0076] 종래의 안티-멀웨어 시스템들은 통상적으로 다른 엔터티들과 별개로 개별 엔터티를 스코어링한다. 일부 멀웨어는 악성 프로세서의 자식 프로세스와 같은 여러 특징적인 에이전트들 사이에서 악성적 활동을 분할함으로써 탐지를 회피하려고 시도할 수 있어서, 개별 에이전트 어떠한 것도 멀웨어를 나타내는 탐지될 활동을 충분히 수행할 수 없다. 반대로, 본 발명의 일부 실시예는 하나의 엔터티로부터 다른 관련 엔터티로 스코어들을 전파해서, 따라서, 관련 엔터티들에 걸쳐서 멀웨어를 나타내는 데이터를 제공한다. 스코어 전파는 악성 활동에 관련된 에이전트들 중 적어도 하나가 탐지되는 것을 보장할 수 있다.

[0077] 하나의 예시적인 회피 전략으로, 멀웨어 에이전트는 다수의 자식 프로세스를 스포닝(spanwing, 생성)하고 종료될 수 있다. 악성 활동들은 자식 프로세스들 사이에서 분할될 수 있고, 그래서 어떠한 개별 자식들의 활동도 그 자체로는 멀웨어 알람을 트리거링 할 수 없다. 본 발명의 일부 실시예에서, 스코어들은 심지어 다른 엔터티가 종료되었을 때도 하나의 엔터티로부터 다른 엔터티로 전파될 수 있다. 그러한 구성은 자식 프로세스의 악성을 탐지하는 것을 비록 실패할 수 있다고 하여도 부모 프로세스를 악성으로 탐지할 수 있다. 일부 실시예들은 현재 평가중인 엔터티들의 리스트를 관리할 수 있다. 상기 리스트는 활성 및 종료 엔터티들을 모두 포함할 수 있다. 엔터티는 개별 엔터티의 모든 후손들이 종료되었을 때에만 상기 리스트로부터 제거될 수 있다.

[0078] 종래의 안티 멀웨어 시스템들에서는 오직 하나의 스코어가 통상적으로 각 엔터티에 대하여 기록된다. 복수의 엔터티별 스코어를 유지하고 이들 각각이 이들의 특징적 기준에 따라서 연산됨으로써, 본 발명의 일부 실시예에서는 스코어들이 기준별로 관련 엔터티들 사이에서 전파될 수 있게 한다. 그러한 스코어들은 전파 시에 증가 또는 감소될 수 있고, 각 엔터티의 라이프 사이클을 통해서 긍정 오류 탐지를 더 줄여서 악성을 보다 정확하게 평가하도록 할 수 있다. 일부 실시예에서, 하나의 엔터터의 스코어가 관련 엔터티의 스코어에 영향을 미치는 정도는 수치적 전파 가중치를 통해서 조정가능하다. 그러한 가중치는 하나의 엔터티로부터 다른 엔터티에서, 그리고/또는 하나의 평가 기준으로부터 다른 평가 기준에서 다를 수 있고, 스코어 전파의 유연하고 정확한 조정(tuning)을 가능하게 한다. 가중치 값들은 인간 운영자에 의하여 결정될 수 있고 그리고/또는 멀웨어 탐지 성능 향상을 목표로 한 자동화된 최적화에 따라 결정될 수도 있다.

[0079] 일부 종래의 안티 멀웨어 시스템들은 개별 엔터티가 멀웨어를 나타내는 행동을 수행하는지 결정하거나, 그리고/또는 엔터티가 멀웨어를 나타내는 코드의 시퀀스와 같은 멀웨어를 나타내는 특징들을 가지는지를 결정하여 피평가 엔터티가 악성인지 여부를 결정한다. 반대로, 본 발명의 일부 실시예에서는, 엔터티 평가 기준이 그 자체로 반드시 멀웨어를 나타내는 것일 필요가 없다. 예를 들어서, 일부 기준은 엔터티가 파일을 열거나 IP 주소에 접근하는 것과 같은 행동을 수행하는지 결정하는 것을 포함한다. 그럼에도 불구하고, 그러한 활동은, 그 자체들은 스스로 멀웨어를 나타낼 수 없는 다른 행동들과 결합할 때 악성일 수 있다. 다양한 엔터티의 행동 및/또는 특징

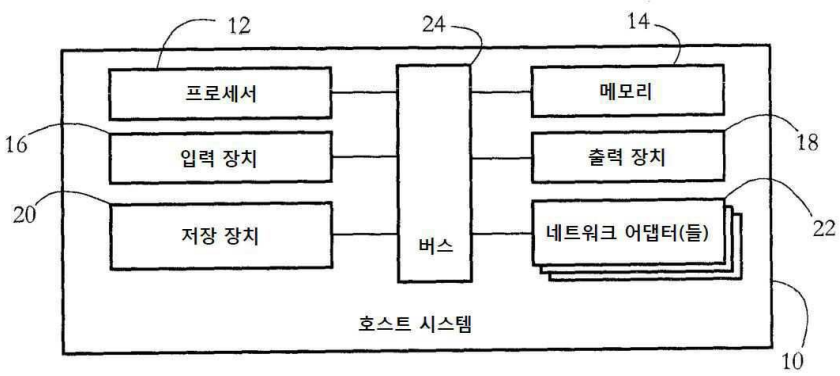
들을 모니터링하고, 다음으로 많은(가능한 수백) 평가 스코어들을 기록하고, 엔터티별 유형으로 그러한 스코어들을 합산함으로써, 본 발명의 일부 실시예 긍정 오류는 최소화하면서 탐지율을 증가시킬 수 있다.

[0080] 본 발명의 일부 실시예들은 가상화된 환경을 보호할 수 있다. 가상화를 지원하도록 구성된 환경에서, 본 발명의 일부 구성요소들은 가상 머신 내에서 실행될 수 있고, 반면에 다른 것들은, 예를 들어서 개별 가상 머신을 노출하는 하이퍼바이저의 레벨에서, 개별 가상 머신 밖에서 실행될 수 있다. 하이퍼바이저 레벨에서 실행되는 그러한 요소들은 개별 호스트 시스템에서 동시에 실행되는 복수의 가상 머신에 대하여 안티-멀웨어 작동을 수행하도록 구성될 수 있다.

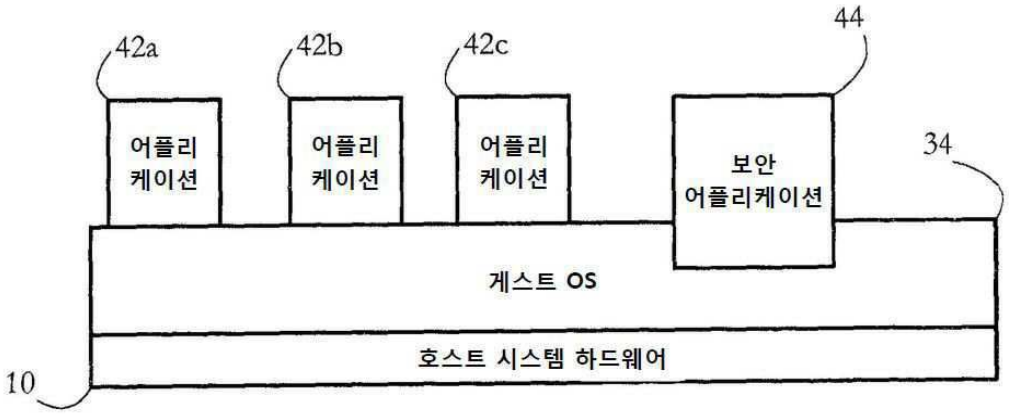
[0081] 상기의 실시예들이 본 발명의 범위를 벗어나지 않는다면 다양한 방법으로 변경될 수 있음은 통상의 기술자에게 당연한 것이다. 따라서 본 발명의 범위는 이하의 청구항과 그들의 법적 균등물에 의해서 결정되어야 한다.

## 도면

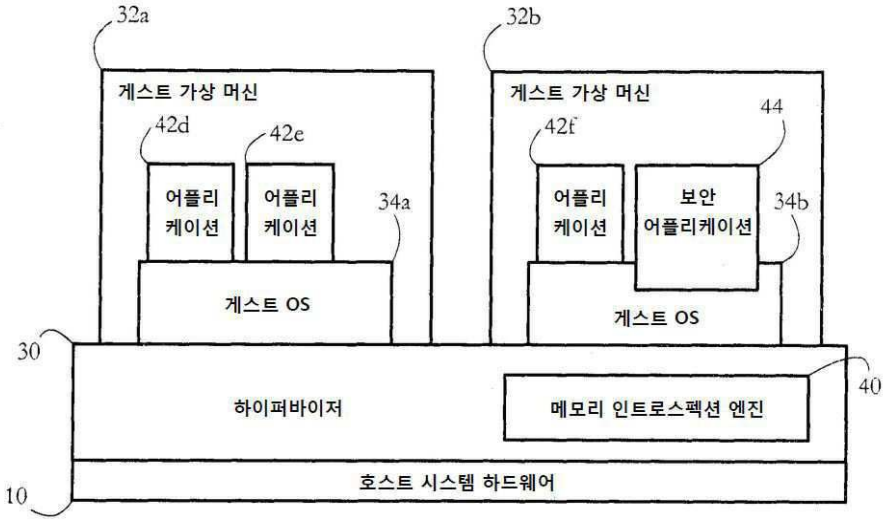
### 도면1



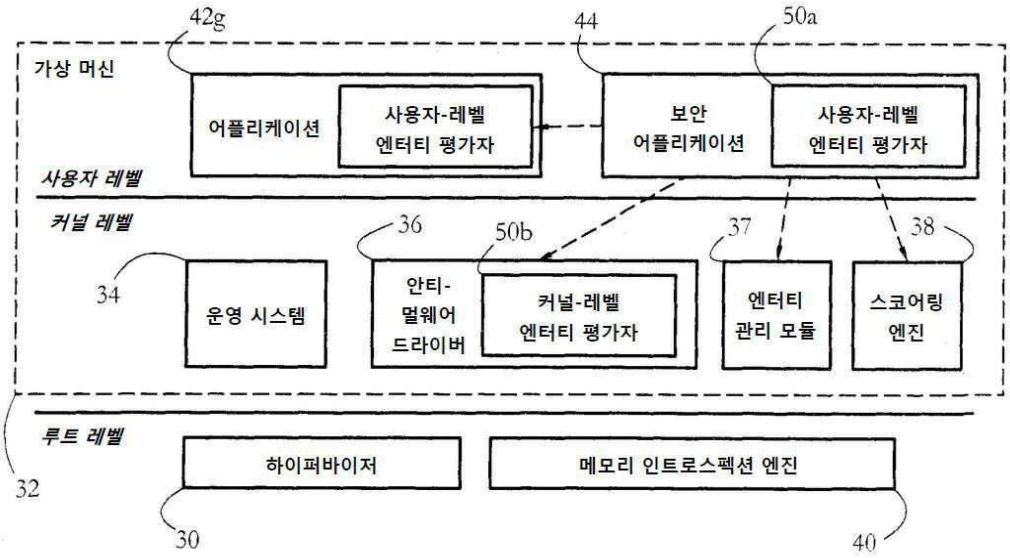

### 도면2a



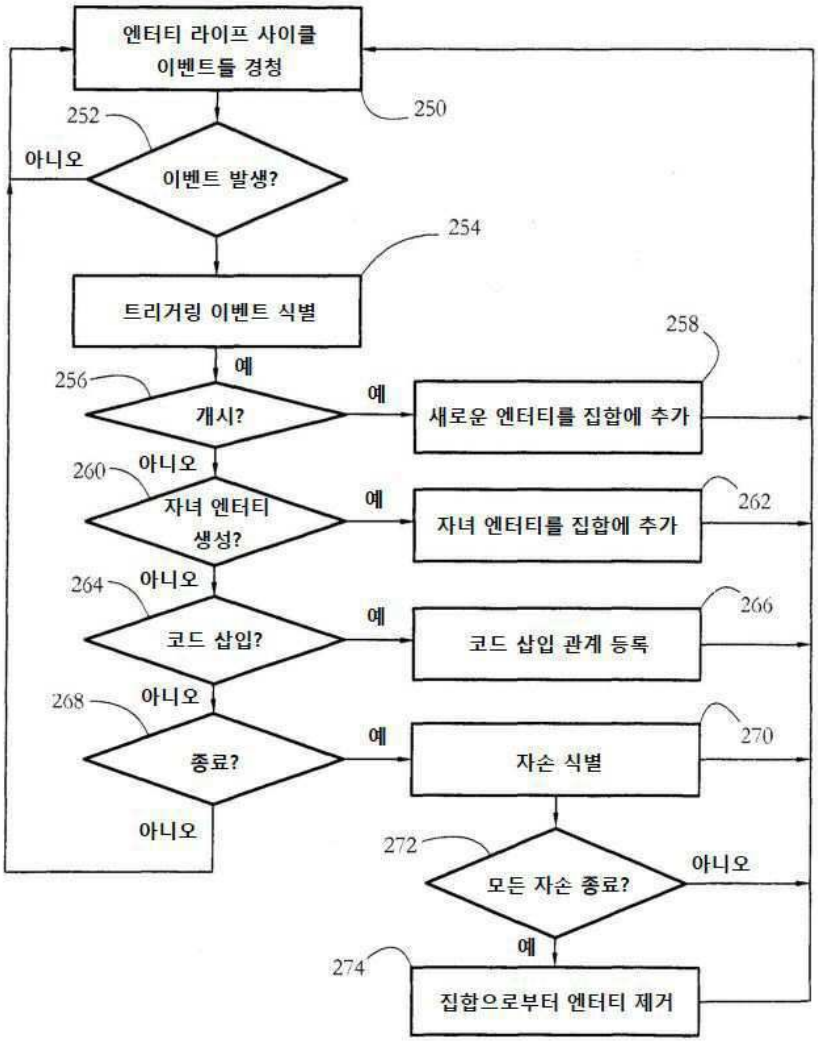
도면2b


32a
32b
게스트 가상 머신
게스트 가상 머신
42d
42e
42f
44
어플리케이션
어플리케이션
어플리케이션
보안 어플리케이션
34a
34b
게스트 OS
게스트 OS
30
하이퍼바이저
메모리 인트로스펙션 엔진
40
호스트 시스템 하드웨어
10

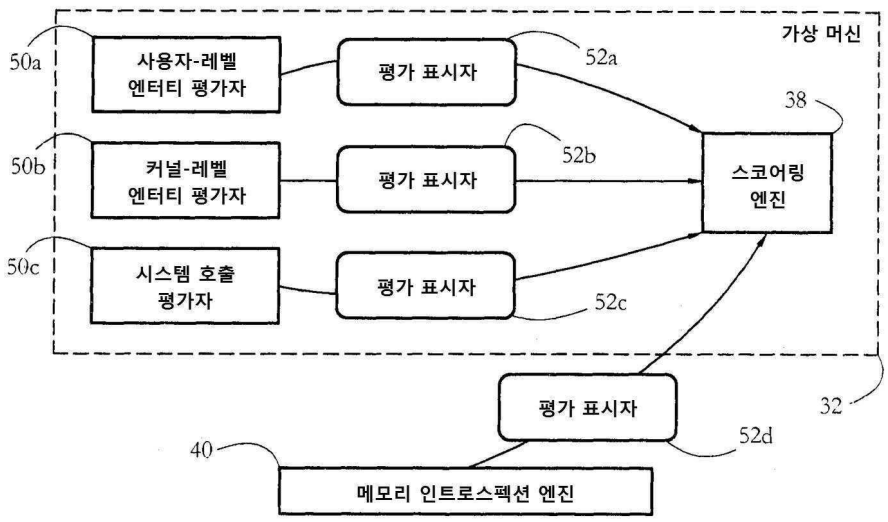
도면3


42g
44
50a
가상 머신
어플리케이션
사용자-레벨 엔터티 평가자
보안 어플리케이션
사용자-레벨 엔터티 평가자
사용자 레벨
커널 레벨
36
50b
37
38
34
운영 시스템
안티-멀웨어 드라이버
커널-레벨 엔터티 평가자
엔터티 관리 모듈
스코어링 엔진
루트 레벨
32
하이퍼바이저
메모리 인트로스펙션 엔진
30
40

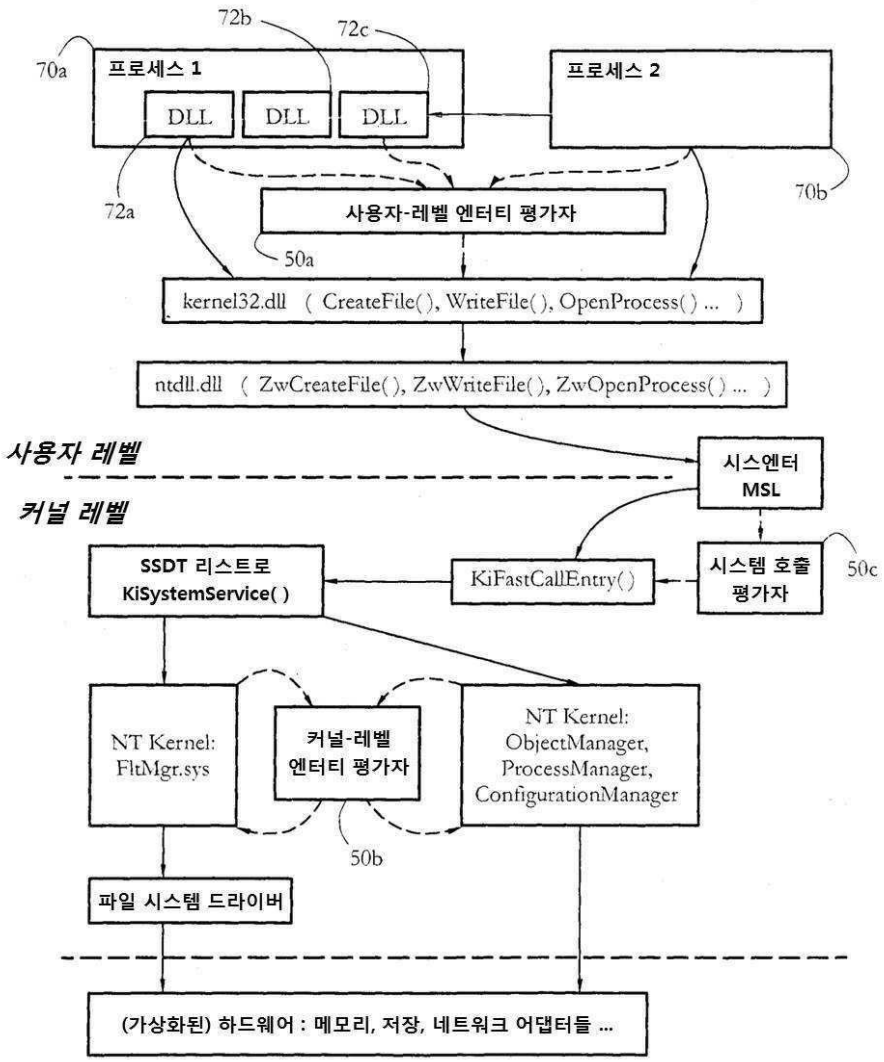
도면4


엔터티 라이프 사이클 이벤트들 경청
250
252
이벤트 발생?
아니오
254
트리거링 이벤트 식별
예
256
개시?
예
258
새로운 엔터티를 집합에 추가
아니오
260
자녀 엔터티 생성?
예
262
자녀 엔터티를 집합에 추가
아니오
264
코드 삽입?
예
266
코드 삽입 관계 등록
아니오
268
종료?
예
270
자손 식별
아니오
272
모든 자손 종료?
아니오
예
274
집합으로부터 엔터티 제거

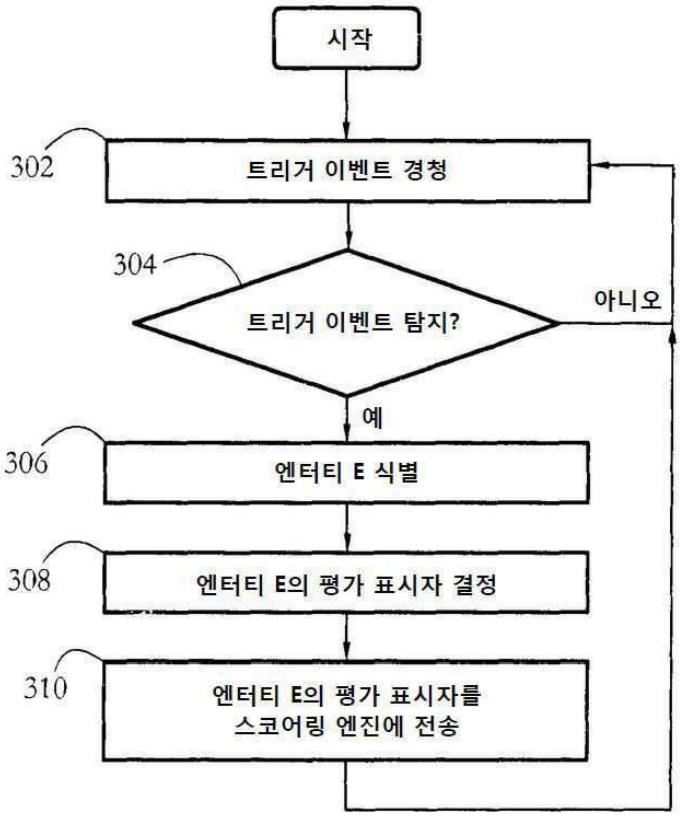
도면5


가상 머신
50a
사용자-레벨 엔터티 평가자
평가 표시자
52a
38
50b
커널-레벨 엔터티 평가자
평가 표시자
52b
스코어링 엔진
50c
시스템 호출 평가자
평가 표시자
52c
32
평가 표시자
52d
40
메모리 인트로스펙션 엔진

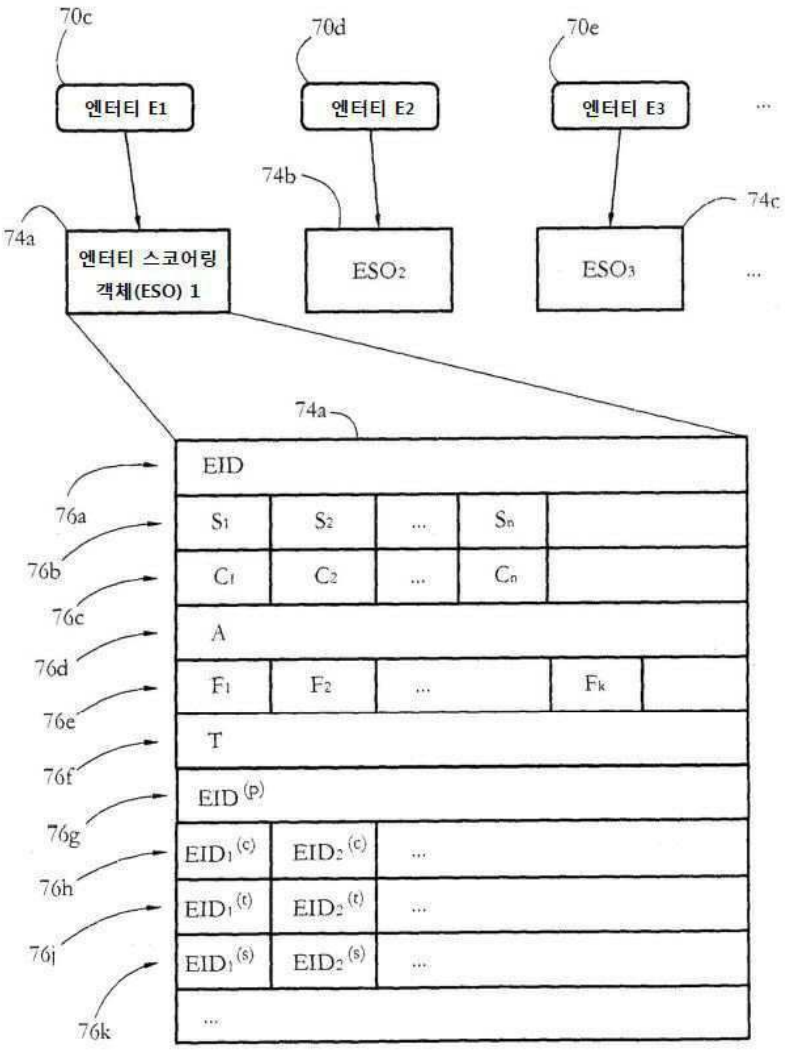
도면6


72b
72c
70a
프로세스 1
DLL
DLL
DLL
프로세스 2
72a
70b
사용자-레벨 엔터티 평가자
50a
kernel32.dll ( CreateFile( ), WriteFile( ), OpenProcess( ) ... )
ntdll.dll ( ZwCreateFile( ), ZwWriteFile( ), ZwOpenProcess( ) ... )
사용자 레벨
시스엔터
MSL
커널 레벨
SSDT 리스트로
KiSystemService( )
KiFastCallEntry( )
시스템 호출
평가자
50c
NT Kernel:
FltMgr.sys
커널-레벨
엔터티 평가자
NT Kernel:
ObjectManager,
ProcessManager,
ConfigurationManager
50b
파일 시스템 드라이버
(가상화된) 하드웨어 : 메모리, 저장, 네트워크 어댑터들 ...

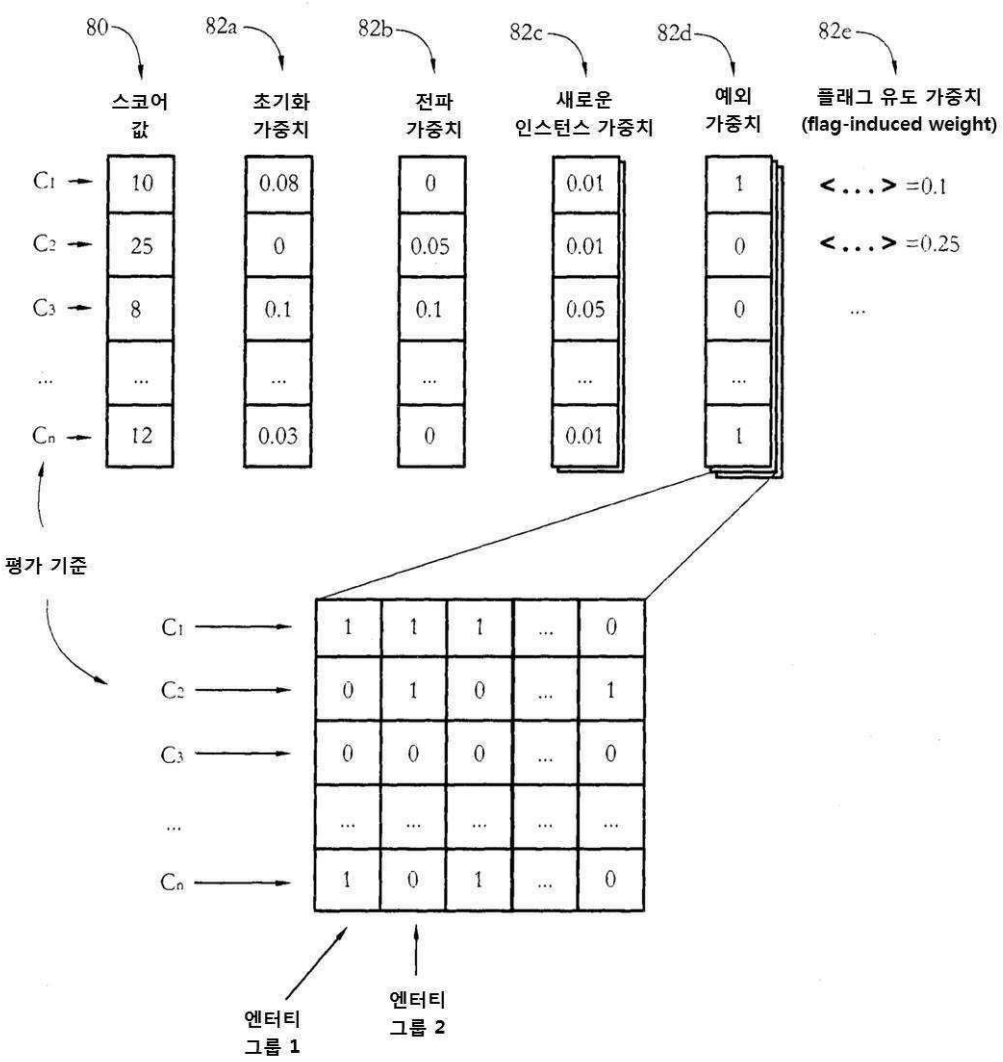
도면7


시작
302
트리거 이벤트 경청
304
트리거 이벤트 탐지?
아니오
예
306
엔터티 E 식별
308
엔터티 E의 평가 표시자 결정
310
엔터티 E의 평가 표시자를
스코어링 엔진에 전송

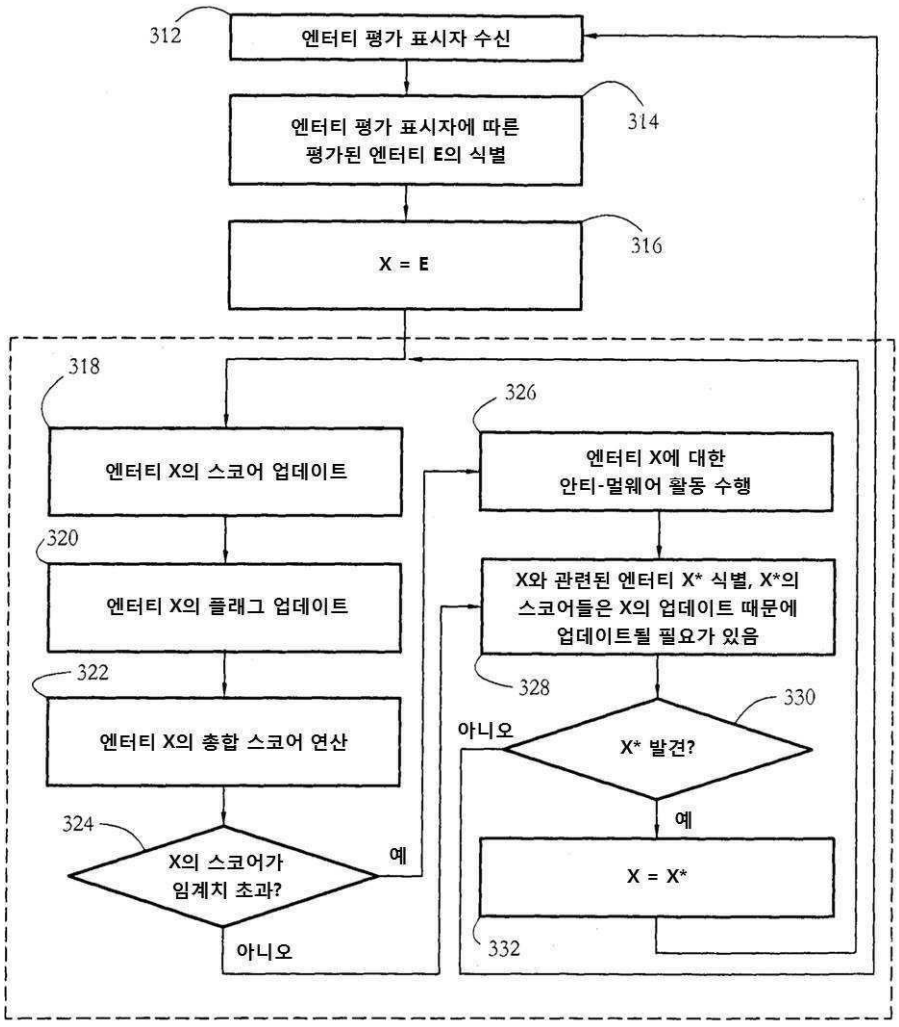
도면8


70c
70d
70e
엔터티 E1
엔터티 E2
엔터티 E3
...
74b
74c
74a
엔터티 스코어링 객체(ESO) 1
ESO2
ESO3
...
74a
EID
76a
S1 S2 ... Sn
76b
C1 C2 ... Cn
76c
A
76d
F1 F2 ... Fk
76e
T
76f
EID (P)
76g
EID1 (c) EID2 (c) ...
76h
EID1 (t) EID2 (t) ...
76j
EID1 (s) EID2 (s) ...
76k
...

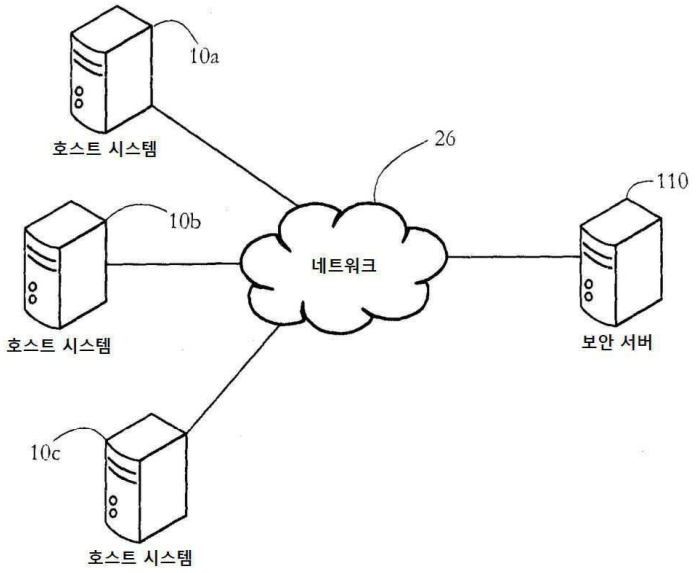
도면9


80
82a
82b
82c
82d
82e
스코어 값
초기화 가중치
전파 가중치
새로운 인스턴스 가중치
예외 가중치
플래그 유도 가중치 (flag-induced weight)
C1
C2
C3
Cn
10
25
8
12
0.08
0
0.1
0.03
0
0.05
0.1
0
0.01
0.01
0.05
0.01
1
0
0
1
<...> =0.1
<...> =0.25
평가 기준
1 1 1 ... 0
0 1 0 ... 1
0 0 0 ... 0
1 0 1 ... 0
엔터티 그룹 1
엔터티 그룹 2

도면10


312
엔터티 평가 표시자 수신
314
엔터티 평가 표시자에 따른
평가된 엔터티 E의 식별
316
X = E
318
엔터티 X의 스코어 업데이트
320
엔터티 X의 플래그 업데이트
322
엔터티 X의 총합 스코어 연산
324
X의 스코어가
임계치 초과?
예
아니오
326
엔터티 X에 대한
안티-멀웨어 활동 수행
X와 관련된 엔터티 X* 식별, X*의
스코어들은 X의 업데이트 때문에
업데이트될 필요가 있음
328
330
X* 발견?
아니오
예
X = X*
332

도면11


10a
호스트 시스템
26
10b
네트워크
110
호스트 시스템
보안 서버
10c
호스트 시스템

도면12