(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2013년06월27일
(11) 등록번호 10-1279683
(24) 등록일자 2013년06월21일

(51) 국제특허분류(Int. Cl.)
G06F 9/06 (2006.01) *G06F 11/30* (2006.01)
G06F 15/00 (2006.01)
(21) 출원번호 10-2009-7015094
(22) 출원일자(국제) 2007년12월19일
심사청구일자 2012년11월30일
(85) 번역문제출일자 2009년07월17일
(65) 공개번호 10-2009-0107035
(43) 공개일자 2009년10월12일
(86) 국제출원번호 PCT/US2007/088219
(87) 국제공개번호 WO 2008/091462
국제공개일자 2008년07월31일
(30) 우선권주장
11/627,314 2007년01월25일 미국(US)
(56) 선행기술조사문헌
US4091447 B1
US4271464 B1
전체 청구항 수 : 총 10 항

(73) 특허권자
마이크로소프트 코포레이션
미국 워싱턴주 (우편번호 : 98052) 레드몬드 원 마이크로소프트 웨이
(72) 발명자
베이커, 브랜든
미국 98052-6399 워싱턴주 레드몬드 원 마이크로소프트 웨이 마이크로소프트 코포레이션 내
필드, 스코트, 에이.
미국 98052-6399 워싱턴주 레드몬드 원 마이크로소프트 웨이 마이크로소프트 코포레이션 내
(뒷면에 계속)
(74) 대리인
제일특허법인

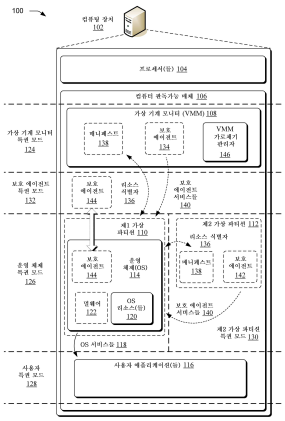
심사관 : 복진요

(54) 발명의 명칭 **운영 체제 리소스 보호**

(57) 요 약

본 문서는 보호 에이전트가, 운영 체제 특권 모드로부터 액세스가능하지 않은 메모리로부터, 운영 체제의 하나 이상의 리소스가 수정되었는지 여부를 판정할 수 있게 해주는 도구들에 대해 기술하고 있다. 어떤 경우에, 이들 도구는 보호 에이전트가 가상 기계 모니터 내에 존재할 수 있게 해준다. 다른 경우에, 이들 도구는 보호 에이전트가 가상 기계 모니터에 의해 제공되는 별도의 가상 파티션 내에 존재할 수 있게 해줄 수 있다. 보호 에이전트는, 운영 체제 특권 모드 밖에서 동작함으로써, 운영 체제 특권 모드 내에서 동작하는 개체들에 의한 공격에 덜 취약할 수 있다.

대 표 도 - 도1



(72) 발명자

트라우트, 에릭

미국 98052-6399 워싱턴주 레드몬드 원 마이크로소프트 웨이 마이크로소프트 코포레이션 내

신하, 수야시

미국 98052-6399 워싱턴주 레드몬드 원 마이크로소프트 웨이 마이크로소프트 코포레이션 내

간굴리, 조이

미국 98052-6399 워싱턴주 레드몬드 원 마이크로소프트 웨이 마이크로소프트 코포레이션 내

폴츠, 포레스트

미국 98052-6399 워싱턴주 레드몬드 원 마이크로소프트 웨이 마이크로소프트 코포레이션 내

커틀러, 데이비드

미국 98052-6399 워싱턴주 레드몬드 원 마이크로소프트 웨이 마이크로소프트 코포레이션 내

특허청구의 범위

청구항 1

컴퓨팅 장치에 의해 실행될 때, 상기 컴퓨팅 장치로 하여금 동작들을 수행하게 하는 컴퓨터 판독가능 명령어들을 갖는 하나 이상의 컴퓨터 판독가능 매체로서,

상기 동작들은,

가상 기계 모니터에 의해 적어도 제1 파티션 및 제2 파티션을 제공하는 동작;

운영 체제 특권 모드 내에서 동작하는 개체에 의해 액세스될 수 없는 메모리에서, 상기 가상 기계 모니터에 의해 제공된 상기 제1 파티션에서 동작하는 운영 체제의 일련의 하나 이상의 리소스를 식별하는 시행 정책(enforcement policy)을 수신하는 동작;

상기 시행 정책을 사용하여, 상기 운영 체제 특권 모드 내에서 동작하는 개체에 의해 액세스될 수 없는 상기 메모리로부터, 상기 가상 기계 모니터에 의해 제공된 상기 제1 파티션에서 동작하는 상기 운영 체제의 상기 일련의 하나 이상의 리소스를 식별하는 동작;

상기 메모리로부터, 상기 일련의 하나 이상의 리소스 중 하나 이상의 리소스가 변경되었는지 여부를 판정하는 동작; 및

상기 일련의 하나 이상의 리소스 중 하나 이상의 리소스가 변경되었다는 판정에 응답하여, (i) 상기 운영 체제를 종료하는 동작, 또는 (ii) 재부팅 시에 상기 운영 체제에게 불법적인 동작을 통지하는 동작을 포함하고,

상기 시행 정책을 수신하는 동작, 상기 일련의 하나 이상의 리소스를 식별하는 동작 및 상기 일련의 하나 이상의 리소스 중 하나 이상의 리소스가 변경되었는지 여부를 판정하는 동작은 (i) 상기 가상 기계 모니터 또는 (ii) 상기 가상 기계 모니터에 의해 제공된 상기 제2 파티션 내에서 이루어지는

컴퓨터 판독가능 매체.

청구항 2

제1항에 있어서,

상기 동작들은, API(application program interface)를 상기 운영 체제에 노출시키고, 상기 API를 통해, 상기 일련의 하나 이상의 리소스 중 하나 이상 리소스의 식별을 수신하는 동작을 더 포함하는

컴퓨터 판독가능 매체.

청구항 3

제1항에 있어서,

상기 일련의 하나 이상의 리소스는 SSDT(system service dispatch table), IDT(interrupt dispatch table), 또는 GDT(global descriptor table)를 포함하는

컴퓨터 판독가능 매체.

청구항 4

운영 체제 리소스와 연관된 메모리 페이지 또는 레지스터가 변경되었다는 표시의 수신을 가능하게 하도록 가상 기계 모니터의 가로채기(intercept) 관리자를 변경하는 단계- 상기 운영 체제 리소스는 상기 가상 기계 모니터에 의해 제공된 파티션 내에 위치함 -;,

상기 가상 기계 모니터에서, 상기 운영 체제 리소스 및 하나 이상의 추가 운영 체제 리소스를 식별하는 시행 정책을 수신하는 단계;

상기 가상 기계 모니터의 상기 가로채기 관리자에 의해, 상기 운영 체제 리소스와 연관된 상기 메모리 페이지 또는 레지스터가 변경되었다는 표시를 수신하는 단계; 및

상기 표시의 수신에 응답하여 상기 운영 체제 리소스와 연관된 운영 체제를 종료시키도록 운영 체제 특권 모드

를 종료시키는 단계를 포함하는 방법.

청구항 5

제4항에 있어서,

상기 운영 체제 리소스는 상기 메모리 페이지에 있는 IDT(interrupt dispatch table)이고, 운영 체제 리소스와 연관된 상기 레지스터는 IDT 레지스터인, 방법.

청구항 6

제4항에 있어서,

상기 운영 체제 리소스는 SSDT(system service dispatch table) 또는 GDT(global descriptor table)인, 방법.

청구항 7

제4항에 있어서,

상기 운영 체제 특권 모드를 종료시키는 단계는 상기 가상 기계 모니터에 의해 수행되는, 방법.

청구항 8

제4항에 있어서,

상기 시행 정책은 또한 상기 식별된 운영 체제 리소스들 각각과 연관된 보호 속성을 기술하며, 상기 보호 속성들 중 적어도 하나는 대응하는 리소스를 읽기 전용인 것으로 기술하는, 방법.

청구항 9

제8항에 있어서,

상기 읽기 전용 보호 속성과 연관된 상기 리소스에 대해 불변성을 시행하는 단계를 더 포함하는 방법.

청구항 10

컴퓨팅 장치에 의해 실행될 때, 상기 컴퓨팅 장치로 하여금 동작들을 수행하게 하는 컴퓨터 판독가능 명령어들을 갖는 하나 이상의 컴퓨터 판독가능 매체로서,

상기 동작들은,

상기 컴퓨팅 장치를 가상 기계 모니터를 통해 적어도 제1 및 제2 가상 기계 파티션으로 가상화하는 동작- 운영 체제 특권 모드와 연관된 운영 체제가 상기 제1 가상 기계 파티션에 존재하고 보호 에이전트가 상기 제2 가상 기계 파티션 또는 상기 가상 기계 모니터에 존재함 -;

상기 보호 에이전트에 의해, 상기 제1 가상 기계 파티션에 존재하고 상기 운영 체제 특권 모드 내에서 동작하도록 설계된 하나 이상의 운영 체제 리소스를 식별하는 동작;

상기 보호 에이전트에 의해, 상기 제1 가상 기계 파티션에 존재하는 상기 하나 이상의 운영 체제 리소스 중 하나 이상의 리소스가 변경되었는지 여부를 판정하는 동작- 상기 제2 가상 기계 파티션 또는 상기 가상 기계 모니터에 존재하는 상기 보호 에이전트는 상기 운영 체제 특권 모드 내로부터의 공격에 영향을 받지 않음 -; 및

상기 하나 이상의 운영 체제 리소스들 중 하나 이상의 리소스가 변경된 것으로 판정한 것에 응답하여 상기 제1 가상 기계 파티션 내의 상기 운영 체제 특권 모드와 연관된 상기 운영 체제를 종료시키는 동작을 포함하는

컴퓨터 판독가능 매체.

청구항 11

삭제

청구항 12

삭제

청구항 13

삭제

청구항 14

삭제

청구항 15

삭제

청구항 16

삭제

청구항 17

삭제

청구항 18

삭제

청구항 19

삭제

청구항 20

삭제

명 세 서

배 경 기 술

[0001] 컴퓨팅 장치 내의 프로세서들은 종종 특권 모드(privileged mode) 및 비특권 모드(unprivileged mode)를 포함한다. 특권 모드에서 실행되는 소프트웨어는 일반적으로 프로세서에 의해 지원되는 모든 명령어를 실행할 수 있다. 통상적으로, 운영 체제 커널은, 때때로 "링 0(Ring 0)", "수퍼바이저 모드(Supervisor Mode)" 또는 "커널 모드(Kernel Mode)"라고 하는 특권 모드 내에서 실행된다.

[0002] 이와 달리, 컴퓨팅 장치 상에서 실행되는 어떤 소프트웨어는 비특권 모드로만 실행되도록 제약되어 있을 수 있다. 이 모드에서는 일반적으로 소프트웨어가 프로세서의 명령어들의 일부분을 실행할 수 있다. 따라서, 운영 체제는 비특권 모드를 사용하여 이 모드에서 실행되는 소프트웨어의 활동을 제한할 수 있다. 예를 들어, 소프트웨어는 컴퓨팅 장치의 메모리 중 특정의 일부분으로 제한될 수 있다. 이러한 비특권 모드는 때때로 "링 3(Ring 3)" 또는 "사용자 모드(User Mode)"라고 한다. 일반적으로, 컴퓨팅 장치의 사용자 애플리케이션은 이 비특권 모드에서 동작한다.

[0003] 소프트웨어 애플리케이션이 이 비특권 모드에서 동작하는 경우, 이 애플리케이션은 비특권 모드로부터 직접 액세스될 수 없는 메모리 일부분에의 액세스를 요청할 수 있다. 이 애플리케이션이, 예를 들어, 이 메모리 일부분에서 "새 파일을 생성" 등의 동작을 수행하고자 할 수 있다. 이 요청은 통상적으로 이러한 비특권 모드 코드를 특권 모드 코드로 전환시키는 콜 게이트(call gate) 또는 기타 시스템 콜(system call) 명령어를 통해 라우팅된다. 이러한 전환은 비특권 모드가 특권 모드로부터만 액세스될 수 있는 것으로 지정되어 있는 메모리에 직접 액세스하지 못하도록 한다.

[0004] 이들 모드에 따르면, 악의적 코드의 저작자는 특권 모드에 액세스하여 컴퓨팅 장치의 거동을 변경시키는 멀웨어(malware)를 설치할 수 있다. 이러한 멀웨어는, 예를 들어, 파일의 위치를 변경시키거나, 파일을 숨기거나, 파일을 수정하거나, 키스트로크를 변경하거나, 기타 등등을 할 수 있다. 이러한 멀웨어의 일부는 컴퓨팅 장치의 거동을 변경시킬 뿐만 아니라 그 자신을 특권 모드의 메모리 내에 감추는 "루트킷(rootkit)"을 포함할 수 있다. 컴퓨팅 장치 상에서 실행되는 안티바이러스 애플리케이션은 그에 따라 이러한 숨겨진 루트킷을 발견하지 못할

수 있으며, 따라서 멀웨어가 악의적인 행동을 계속할 수 있게 된다. 게다가, 이러한 멀웨어는, 이하에 설명하는 바와 같이, 운영 체제의 내장된 보호 시스템에 패치될 수 있다.

[0005] 멀웨어 저작자는, 컴퓨팅 장치 사용자를 속여 사용자의 컴퓨팅 장치에 부지 중에 멀웨어를 설치하게 하는 것을 비롯하여, 다양한 방식으로 특권 모드에 액세스하여 컴퓨팅 장치에 멀웨어를 로드할 수 있다. 그 결과, 현재의 운영 체제는 이러한 멀웨어를 탐지하기 위해 종종 하나 이상의 보호 시스템을 이용한다. 이러한 보호 시스템은 일반적으로 어떤 중요한 운영 체제 리소스들에 대한 어떤 변경들을 탐지하기 위해 이들 리소스를 모니터링한다. 이러한 보호 시스템이 이러한 변경을 탐지하는 경우, 보호 시스템은 그 특정의 리소스가 멀웨어에 의해 감염된 것으로 결정할 수 있다. 이들 보호 시스템은 또한 현재 비특권 모드의 메모리에 존재하는 애플리케이션들의 리스트를 사용자의 안티바이러스 애플리케이션에 제공할 수 있다. 물론, 멀웨어가 성공적으로 숨어 있는 경우, 멀웨어가 제공된 리스트에 나타나지 않게 된다. 게다가, 멀웨어가 성공적으로 보호 시스템에 패치된 경우, 보호 시스템이 실행되지 못하거나 중요한 운영 체제 리소스들에 대한 어떤 변경도 탐지하지 못할 수 있다.

[0006] 이들 보호 시스템은, 효과적일 수 있는 반면, 몇가지 약점도 가질 수 있다. 첫째, 이들 시스템은 종종 불명료함에 의존하며, 따라서 멀웨어에 의해 식별되는 경우 이용당할 수 있다. 즉, 멀웨어가 보호 시스템의 정체를 해독하고 보호 시스템을 찾아내는 경우, 멀웨어는 보호 시스템 자체를 무력화시킬 수 있다. 멀웨어 저작자는 또한 다른 사람들에게 똑같이 하는 방법에 관해 알려줄 수 있다. 게다가, 첫번째 것과 관련하여, 이들 보호 시스템은 일반적으로 운영 체제와 동일한 보호 도메인(protection domain)에서(예를 들어, 특권 모드 자체 내에서) 동작한다. 따라서, 멀웨어가 특권 모드에 액세스하여 은닉된 보호 시스템의 정체를 알아낼 수 있는 경우, 보호 시스템 자체가 공격을 받는다. 마지막으로, 이들 보호 시스템은 운영 시스템 또는 특권 모드와 동시에 초기화한다. 따라서, 멀웨어 또는 멀웨어 저작자는, 이 초기화 전에 컴퓨팅 장치를 장악하는 경우, 보호 시스템이 초기화하지 못하도록 할 수 있다.

발명의 상세한 설명

[0007] 본 문서는 보호 에이전트가, 운영 체제 특권 모드로부터 액세스가능하지 않은 메모리로부터, 운영 체제의 하나 이상의 리소스가 수정되었는지 여부를 판정할 수 있게 해주는 도구들에 대해 기술하고 있다. 어떤 실시예들에서, 이들 도구는 보호 에이전트가 가상 기계 모니터 내에 존재할 수 있게 해준다. 다른 실시예들에서, 이들 도구는 보호 에이전트가 가상 기계 모니터에 의해 제공되는 별도의 가상 파티션 내에 존재할 수 있게 해줄 수 있다. 보호 에이전트는, 운영 체제 특권 모드 밖에서 동작함으로써, 운영 체제 특권 모드 내에서 동작하는 개체들에 의한 공격에 덜 취약할 수 있다.

[0008] 이 요약은 이하에서 상세한 설명에 더 기술되는 일련의 개념들을 간략화된 형태로 소개하기 위해 제공된 것이다. 이 요약은 청구된 발명 대상의 주요 특징들 또는 필수적인 특징들을 확인하기 위한 것이 아니며, 청구된 발명 대상의 범위를 정하는 데 보조 수단으로 사용되기 위한 것도 아니다. 예를 들어, 용어 "도구"는 이상의 내용에 의해 또한 본 문서 전체를 통해 가능하게 되는 시스템(들), 방법(들), 컴퓨터 판독가능 명령어들, 및/또는 기법(들)을 말하는 것일 수 있다.

실 시 예

[0020] 명세서 및 도면에 걸쳐 동일한 구성요소들 및 특징들을 참조하는 데 동일한 번호가 사용되고 있다.

[0021] 이하의 문서는 보호 에이전트를 운영 체제 특권 모드로부터 변경하거나 액세스하지 못하게 하는 방식으로 보호 에이전트를 동작시킬 수 있는 도구들에 대해 기술하고 있다. 따라서, 이들 도구가 보호 에이전트 자체를 보호할 수 있음으로써, 보호 에이전트가 중요한 운영 체제 리소스들에 대한 변경을 탐지할 수 있도록 해준다. 그에 부가하여, 이들 도구는 리소스 변경을 탐지한 것에 응답하여 또는 보호 에이전트 자체를 수정하려는 시도에 응답하여 운영 체제 또는 운영 체제 특권 모드를 종료시킬 수 있다. 게다가, 이들 도구는 보호 에이전트가 운영 체제 리소스들에 대해 불변성(invariance)을 시행할 수 있게 해줌으로써, 그 후에 리소스 수정을 탐지할 필요가 없게 된다.

[0022] 이들 도구가 이들 동작 및 기타 동작들을 가능하게 해줄 수 있는 환경이 이하에서 *예시적인 운영 환경*이라는 제목의 섹션에 기술되어 있다. 그 다음에 오는 *자율 보호 에이전트*(Autonomous Protection Agents)라는 제목의 섹션은 2개의 서브섹션을 포함한다. *가상 기계 모니터 보호 에이전트*라는 제목의 첫번째 서브섹션은 보호 에이전트가 가상 기계 모니터 내에 존재하고 실행될 수 있는 한가지 예시적인 방식에 대해 설명하고 있다. 이 다음에 오는 *가상 파티션 보호 에이전트*라는 제목의 또 하나의 서브섹션은 보호 에이전트가 운영 체제의 파티션과

분리되어 있는 가상 파티션 내에 존재하고 실행될 수 있는 한가지 예시적인 방식에 대해 설명하고 있다.

[0023] 그 다음에 오는 *자율 보호 에이전트 특권 모드*라는 제목의 또 하나의 섹션도 역시 2개의 서브섹션을 포함하고 있다. 첫번째 서브섹션은 가상 기계 모니터 타이머가 하부의 프로세서에 보호 에이전트 특권 모드를 추가할 수 있는 한가지 예시적인 방식에 대해 설명하고 있으며, *가상 기계 모니터에 대한 보호 요청*이라는 제목으로 되어 있다. 그 다음에 오는 *보호 에이전트 가상 프로세서*라는 제목의 서브섹션은, 이 경우에, 보호 에이전트 특권 모드에서 보호 에이전트를 실행하도록 구성되어 있는 가상 프로세서를 비롯한 다수의 가상 프로세서를 사용하여, 보호 에이전트 특권 모드가 생성될 수 있는 또 하나의 방식에 대해 설명하고 있다. 그 다음에 오는 *본 발명의 도구들의 예시적인 용도*라는 제목의 섹션은 앞서 설명한 도구들의 동작의 일례에 대해 설명한다. 마지막으로, *본 발명의 도구들의 다른 실시예들*이라는 제목의 섹션은 본 발명의 도구들이 동작할 수 있는 다양한 다른 실시예들 및 방식들에 대해 설명하고 있다. 이들 섹션 제목 및 요약을 포함한 이 개요는 읽는 사람의 편의를 위해 제공된 것이며, 청구항들 또는 이들 섹션의 범위를 제한하려는 것이 아니다.

[0024] *예시적인 운영 환경*

[0025] 본 발명의 도구들에 대해 상세히 설명하기 전에, 예시적인 운영 환경에 대한 이하의 설명은 읽는 사람이 본 발명의 도구들의 다양한 측면들이 이용될 수 있는 몇몇 방식들을 이해하는 데 도움을 주기 위해 제공된 것이다. 이하에서 설명되는 환경은 단지 일례에 불과하며, 본 발명의 도구들의 응용을 임의의 한 특정의 운영 환경으로 제한하기 위한 것이 아니다. 청구된 발명 대상의 정신 및 범위를 벗어나지 않고 다른 실시예들이 사용될 수 있다. 예를 들어, 이하의 섹션들이 단일의 보호 에이전트를 갖는 실시예들을 설명하고 있지만, 다수의 보호 에이전트들이 이용될 수도 있다. 어떤 예들에서, 이들 보호 에이전트는 독립적으로 나란히 실행될 수 있다. 이러한 경우들에, 보호 에이전트들은 통상적으로 그 각자의 파티션 내의 메모리에 액세스할 수 있다. 게다가, 이하에서 설명하는 기법들이 동시에 이용될 수 있다. 즉, 서로 다른 보호 에이전트들이 동일한 운영 환경 내에서 서로 다른 기법들을 이용할 수 있다.

[0026] 본 실시예를 참조하면, 도 1은 전체적으로 100으로 나타낸 한가지 이러한 예시적인 운영 환경을 도시한 것이다. 이 환경은 컴퓨팅 장치(102)를 포함하고, 컴퓨팅 장치(102) 자체는 하나 이상의 프로세서(104)는 물론, 컴퓨터 판독가능 매체(106)를 포함한다. 컴퓨터 판독가능 매체(106)는 가상 기계 모니터(108)(예를 들어, 하이퍼바이저)를 포함하고, 이 가상 기계 모니터(108)는 하나 이상의 프로세서들을 다수의 가상 프로세서들로 가상화하는 것을 가능하게 해줄 수 있다. 가상 기계 모니터(108)는 또한 다수의 가상 파티션들을 가능하게 해줄 수 있다. 하나 이상의 프로세서들이 각각의 파티션과 연관되어 있을 수 있으며, 이 가상 프로세서들은 이용가능한 물리 프로세서들 상으로 스케쥴링된다. 도시된 바와 같이, 어떤 실시예들에서, 가상 기계 모니터는 제1 가상 파티션(110) 및 제2 가상 파티션(112)을 가능하게 해줄 수 있다. 이하에서 상세히 설명하는 바와 같이, 이 파티션들은 운영 체제 기능들을 보호 에이전트 서비스들과 분리시키는 기능을 할 수 있다.

[0027] 또한, 도시된 바와 같이, 컴퓨터 판독가능 매체(106)는 또한 운영 체제(OS)(114)는 물론, 하나 이상의 사용자 애플리케이션들(116)을 포함하고 있다. 운영 체제(114)는 사용자 애플리케이션들(116)에 운영 체제 서비스들(118)을 제공함으로써, 이 애플리케이션들이 컴퓨팅 장치 상에서 실행될 수 있게 해준다. 그에 부가하여, 하나 이상의 운영 체제 리소스들(120)이 운영 체제 상에 존재한다. 예시적인 리소스들로는 SSDT(system service dispatch table), IDT(interrupt dispatch table), GDT(global descriptor table), 기타 등등이 있다. 또한, 도시된 바와 같이, 운영 체제는 상기한 방식으로 또는 다른 방식으로 컴퓨팅 장치에 로드되어 있을지도 모르는 멀웨어(122)(즉, 악의적 의도를 갖는 코드)를 포함할 수도 있다. 이하에서 설명하는 하나 이상의 보호 에이전트들은 멀웨어에 의해 운영 체제 리소스들에 행해진 변경들을 탐지할 수 있고, 이 탐지에 응답하여 방어 조치를 취할 수 있다. 보호 에이전트는, 이러한 판정을 하는 경우, 운영 체제 및/또는 컴퓨팅 장치를 종료시키거나 다른 대응 조치를 취할 수 있다.

[0028] 컴퓨팅 장치의 구조에 대해 설명하였으며, 이제부터는 하부의 하나 이상의 물리 프로세서들(104) 상에 존재하는 다양한 특권 모드들에 대해 살펴본다. 가상 기계 모니터 특권 모드(124)는 도 1에 예시된 가장 많은 특권을 갖는 모드를 나타낸다. 이 특권 모드는 장치의 모든 또는 거의 모든 리소스들 및 메모리에 액세스할 수 있다. 가상 기계 모니터 특권 모드(124)로부터, 가상 기계 모니터는 프로세서들을 스케쥴링하고, 각각의 가상 파티션에 대한 메모리 영역들에의 액세스를 가능하게 해준다. 한 파티션 내에서 실행되는 운영 체제는 자기가 물리 프로세서의 모든 리소스들을 제어하는 것으로 생각할 수 있지만, 실제로는 가상 기계 모니터에 의해 결정된 일부분만을 제어한다.

[0029] 운영 체제 특권 모드(126)는, 가상 기계 모니터 특권 모드보다 더 적은 특권을 갖기 때문에, 모든 운영 체제 리

소스(120) 및 대부분의 또는 모든 운영 체제 메모리에 액세스할 수 있다. 그렇지만, 이 특권 모드가 제2 가상 파티션(112) 등의 다른 파티션과 연관된 리소스들 또는 메모리에는 액세스하지 못한다. 그럼에도 불구하고, 이 특권 모드는, 일반적으로 모든 운영 체제 메모리에 액세스할 수 있기 때문에, 때때로 "특권있는 모드(Privileged Mode)"라고 한다. "링 0", "수퍼바이저 모드", 또는 "커널 모드"도 역시 이 특권 모드를 말하는 것일 수 있다. 상기한 바와 같이, 운영 체제 특권 모드(126) 내에서 동작하는 사용자 애플리케이션은 일반적으로, 가상 기계 모니터 모드를 위해 예약된 명령어들을 제외한, 프로세서에 의해 제공되는 대부분의 명령어들을 실행할 수 있다.

[0030] 이 운영 체제 특권 모드는, 때때로 "비특권 모드", "링 3", 또는 간단히 "사용자 모드"라고 하는 사용자 특권 모드(128)와 대비된다. 또한, 상기한 바와 같이, 사용자 애플리케이션은, 사용자 특권 모드(128)로부터 동작할 때, 운영 체제와 연관된 어떤 메모리에 액세스하거나 그 메모리를 변경할 수 없다. 일반적으로, 컴퓨팅 장치 사용자 애플리케이션은, 기본적인 동작들을 수행할 때, 이 사용자 특권 모드에서 동작한다.

[0031] 상기한 모드들에 부가하여, 도 1은 또한 제2 가상 파티션 특권 모드(130) 및 보호 에이전트 특권 모드(132)도 나타내고 있다. 이하에서 상세히 설명하는 바와 같이, 보호 에이전트 특권 모드(132)는 운영 체제 특권 모드가 액세스하지 못하는 메모리 일부분에 액세스할 수 있지만, 일반적으로 가상 기계 모니터 특권 모드만큼의 메모리 액세스 권한을 갖고 있지는 않다. 따라서, 이 특권 모드는 운영 체제 특권 모드보다는 더 많은 특권을 가질 수 있지만, 가상 기계 모니터 특권 모드보다는 더 적은 특권을 가질 수 있다.

[0032] 또한 이하에서 상세히 설명하는 바와 같이, 제2 가상 파티션 특권 모드는 일반적으로 제2 가상 파티션(112)와 연관된 메모리에 액세스한다. 그에 부가하여, 이 모드는 제1 가상 파티션에 액세스할 수 있다. 이러한 부가적인 액세스에 의해, 예를 들어, 제2 가상 파티션에 존재하는 보호 에이전트가 제1 가상 파티션 및 그의 대응하는 운영 체제와 연관된 메모리를 스캔할 수 있게 될 수 있다. 이 모드는 일반적으로 가상 기계 모니터에 액세스하지 못하며, 따라서 가상 기계 모니터 특권 모드보다 더 적은 특권을 갖는다. 그럼에도 불구하고, 제2 가상 파티션 특권 모드는 여전히 운영 체제 특권 모드가 액세스하지 못하는 메모리 일부분에 액세스한다.

[0033] 한편, 도 2는 컴퓨팅 장치 메모리 권한(200)을 나타낸 것이다. 동 도면은 따라서 도 1의 모듈들에 의해 액세스 가능한 메모리의 양을 나타낸다. 도시된 바와 같이, 가상 기계 모니터 특권 모드(124)에서 동작하는 가상 기계 모니터(108)는 도시된 모듈들 전부 중에서 가장 많은 메모리 권한을 갖는다. 실제로, 가상 기계 모니터는 메모리 일부분(202)에 존재하며 가상 기계 모니터만이 그 일부분에 액세스한다. 그 다음에, 보호 에이전트(204)(예를 들어, 도 1에 도시된 보호 에이전트들 중 임의의 보호 에이전트)는 보호 에이전트 특권 모드(132)에서 동작하고, 가상 기계 모니터에 대응하는 일부분(202)을 제외한 모든 메모리에 액세스한다. 그렇지만, 보호 에이전트는 보호 에이전트 자체가 존재하는 메모리 일부분인 메모리 일부분(206)에 액세스한다.

[0034] 한편, 운영 체제(114)는 운영 체제 특권 모드(126)에서 동작하며, 일부분(202) 및 일부분(206)을 제외한 모든 메모리에 액세스한다. 운영 체제가 보호 에이전트와 연관된 메모리 일부분(206)에 액세스할 수 없지만, 운영 체제 및 그의 연관된 특권 모드는 메모리 일부분(208)에 액세스한다. 이 메모리 일부분(208)은 때때로 커널 메모리, 즉 운영 체제의 최저 수준 컴포넌트라고 하며, 일반적으로 도 1에 도시된 리소스들을 포함한다. 그렇지만, 멀웨어가 로드되어 메모리 일부분(208)에서 동작하더라도, 멀웨어는 보호 에이전트와 연관된 메모리 일부분(206)에 액세스할 수 없다.

[0035] 마지막으로, 도 2에서 사용자 애플리케이션(116)이 메모리 일부분(210)에만 액세스하는 것으로 나타내어져 있다. 이들 사용자 애플리케이션 및 대응하는 사용자 특권 모드는 운영 체제의 최저 수준 컴포넌트과 연관된 메모리 일부분(208)에 액세스하지 못한다. 이러한 운영 환경을 염두에 두고서, 이하의 4개의 섹션들은 보호 에이전트가 운영 체제 특권 모드로부터 변경되거나 액세스되지 않도록 할 수 있는 예시적인 방식들에 대해 상세히 설명한다.

[0036] *자율 보호 에이전트*

[0037] 이하의 섹션은, 운영 체제 특권 모드 내에서 동작하는 개체에 의해 액세스될 수 없는 메모리로부터, 하나 이상의 운영 체제 리소스들이 수정되었는지를 판정할 수 있는 도구들에 대해 설명한다. 따라서, 이 도구들은 보호 에이전트가 운영 체제 메모리 자체의 장소 이외의 장소에 존재할 수 있게 해줄 수 있다. 보다 상세하게는, 이하의 서브섹션들은 보호 에이전트가 가상 기계 모니터 내에 또는 자율 가상 파티션 내에 존재할 수 있는 방법에 대해 설명한다.

[0038] *가상 기계 모니터 보호 에이전트*

[0039] 이 서브섹션은, 도 1에 나타낸 바와 같이, 보호 에이전트(134)가 가상 기계 모니터 자체 내에 존재할 수 있는 방법에 대해 설명한다. 운영 체제 특권 모드가 가상 기계 모니터에 액세스할 수 없기 때문에, 이 장소는 보호 에이전트를 운영 체제 메모리에 위치해 있는 멀웨어로부터 보호한다. 보호 에이전트는, 이 장소로부터 동작하기 위해, 보호 에이전트(134)가 모니터링할 수 있는 하나 이상의 운영 체제 리소스들(120)의 식별(identification)을 수신한다. 이 식별은 리소스 식별자(136)를 통해 수신될 수 있다. 도시된 바와 같이, 운영 체제는 API(application programming interface) 호출을 통해 이 정보를 가상 기계 모니터에 제공할 수 있거나, 운영 체제는 그 정보를 매니페스트(manifest)(138)의 형태로 제공할 수 있다. 상기한 바와 같이, 이 리소스들로는 SSDT, IDT 및 GDT가 있을 수 있다.

[0040] 보호 에이전트(134)는, 리소스들의 식별을 수신하면, 보호 에이전트 서비스들(140)을 운영 체제(114)로 확장한다. 이 보호 에이전트 서비스들은 일반적으로 식별된 리소스들 중 임의의 리소스가 변경되었는지를 판정하는 것을 포함한다. 보호 에이전트 또는 가상 기계 모니터는, 변경된 것으로 판정된 경우, 예를 들어, 운영 체제를 종료시킬 수 있다. 보호 에이전트 서비스들은 또한 변경불가(unalterable)(예를 들어, "읽기 전용")로 표시된 임의의 리소스들에 대해 불변성(invariance)을 시행하는 것을 포함할 수 있다.

[0041] 이러한 아키텍처를 이용하는 것은 하나 이상의 운영 체제를 호스팅할 수 있는 가상 기계 모니터를 로드하여 초기화하는 것으로 시작한다. 이 예에서, 가상 기계 모니터는 하나의 운영 체제(114)를 호스팅하며, 이 운영 체제(114) 자체는 가상 기계 모니터가 로드된 후에 초기화를 시작한다. 운영 체제의 초기화 동안에, 운영 체제의 최저 수준 컴포넌트(예를 들어, 커널)와 연관된 메모리 일부분(208)이 먼저 로드된다. 운영 체제 리소스들(120)(예를 들어, SSDT, GDT, IDT) 중 몇몇 또는 그 전부는 일반적으로 이 메모리 일부분(208)을 금지시킨다.

[0042] 운영 체제가 초기화하기 전에 또는 초기화하는 동안에, 보호 에이전트(134)가 가상 기계 모니터 내로부터 실행되기 시작할 수 있다. 상기한 바와 같이, 보호 에이전트는 일반적으로 일련의 하나 이상의 운영 체제 리소스들의 식별을 수신하고, 식별된 리소스들 중 하나 이상이 변경되었는지 여부를 판정한다. 유의할 점은, 각각의 식별된 리소스가 종종 다수의 장소에 있는 다수의 컴포넌트를 포함하며, 보호 에이전트가 전체 리소스를 완전히 보호하기 위해 이들 컴포넌트 각각을 모니터링할 수 있다는 것이다. 예를 들어, 매니페스트가 SSDT를 모니터링하여 보호할 리소스로 식별해주는 경우, 보호 에이전트는 실제 테이블(actual table)을 보호할 뿐만 아니라, SSDT의 다른 컴포넌트들도 보호한다. 예를 들어, 보호 에이전트는 또한 테이블의 위치를 가리키는 레지스터를 모니터링하고 스캔할 수 있다. 게다가, 보호 에이전트는 또한 SSDT의 가상 주소를 물리 주소로 변환하는 메모리 변환 데이터 구조(memory translation data structure)(예를 들어, 페이지 테이블)를 모니터링할 수 있다. 보호 에이전트가 그렇게 하지 못하는 경우, 악의적 코드가 다른 페이지 테이블 매핑을 갖는 또 하나의 테이블을 생성할 수 있다(즉, SSDT 자체를 우회할 수 있다).

[0043] 식별에 부가하여, 보호 에이전트는 또한 대응하는 리소스를 보호하는 방식에 관해 보호 에이전트에게 알려주는 보호 속성을 수신할 수 있다. 예를 들어, 보호 에이전트는 SSDT 리소스의 식별은 물론, "읽기 전용(read-only)"이라는 대응하는 보호 속성도 수신할 수 있다. 따라서, 보호 에이전트는 SSDT가 읽기 전용인 채로 있어야 하고 따라서 변경되어서는 안된다는 것을 알게 된다. "초기화 읽기 전용(init read-only)"은 대응하는 리소스가 초기화 동안에 한번 쓰기를 할 수 있지만 초기화 이후에 리소스가 읽기 전용인 채로 있어야 한다는 것을 보호 에이전트에게 알려주는 또 하나의 가능한 보호 속성이다.

[0044] 보호 에이전트는, 능동적으로도 수동적으로도, 다수의 방식으로 리소스들 및 리소스 보호 속성들의 이러한 식별을 수신할 수 있다. 예를 들어, 운영 체제는 보호 에이전트가 모니터링할 수 있는 리소스들을 식별해주는 디지털 서명된 매니페스트를 제공할 수 있다. 이러한 디지털 서명된 매니페스트는 리소스들을 메모리 일부분(208) 내의 대응하는 장소들에 매핑하는 다수의 방식으로(이름(예를 들어, SSDT, IDT, GDT, 기타) 또는 주소 등에 의해) 리소스들을 식별해줄 수 있다. 주소의 경우에, 매니페스트는 리소스의 게스트 물리 주소(guest physical address), 게스트 가상 주소(guest virtual address), 또는 시스템 물리 주소(system physical address)를 식별해줄 수 있다. 유의할 점은, 어떤 경우들에, 대응하는 리소스 컴포넌트의 실제의 물리 장소를 알아내기 위해 게스트 물리 주소가 실제의 시스템 물리 주소에 매핑될 수 있다는 것이다.

[0045] 가상 기계 모니터 또는 보호 에이전트가 매니페스트를 수신한 후에, 이 컴포넌트들은 매니페스트가 변조 또는 수정되었는지를 판정할 수 있다. 가상 기계 모니터 또는 보호 에이전트는, 매니페스트가 변조 또는 수정된 것으로 판정하는 경우, 운영 체제를 기동시키지 않기로 결정할 수 있다. 그에 부가하여, 리소스들의 리스트와 연관된 암호화가 무효화됨으로써, 그의 보안을 보호할 수 있다.

[0046] 매니페스트에 부가하여 또는 매니페스트의 대안으로서, 보호 에이전트는 가상 기계 모니터 내로의 하나 이상의

API 호출(예를 들어, 하이퍼콜(hypercall))을 통해 리소스 및 보호 속성 식별을 수신할 수 있다. 운영 체제가 초기화할 때, 운영 체제(아마도 운영 체제(208)의 최저 수준 컴포넌트)는 가상 기계 모니터 내로 하이퍼콜을 하여, 모니터링되고 보호될 수 있는 어떤 리소스들을 보호 에이전트에 알려줄 수 있다. 이 하이퍼콜은 상기한 바와 동일한 방식으로 관련 리소스들을 식별해줄 수 있다. 또한, 상기한 바와 같이, 이 하이퍼콜은 또한 리소스들의 보호 속성들을 식별해줄 수 있다.

[0047] 디지털 서명된 매니페스트는 물론, 하나 이상의 하이퍼콜을 이용하는 실시예들에서, 운영 체제가 부팅하기 전에 또는 부팅하는 동안에, 보호 에이전트는 먼저 매니페스트에서 식별되는 리소스들을 스캔할 수 있다. 이러한 초기 스캔 이후에, 운영 체제는 가상 기계 모니터 내로 하이퍼콜을 하여, 하이퍼콜에 의해 식별된 페이지들이 변경되었는지 여부를 판정하도록 보호 에이전트에게 지시할 수 있다. 따라서, 매니페스트는 운영 체제 부팅 시마다 스캔할 리소스들을 식별해주는 반면, 하이퍼콜은 그 각자의 초기화 시에 동적으로 스캔할 리소스들을 식별해준다.

[0048] 보호 에이전트는, 모니터링할 리소스들을 식별한 경우, 리소스들(예를 들어, 상기한 SSDT의 일부분들 전부)이 변경되었는지 여부를 판정한다. 보호 에이전트는 또한 식별된 리소스들에 대해 불변성을 시행할 수 있다. 예를 들어, 보호 에이전트는 "읽기 전용(read-only)"으로 표시된 임의의 리소스가 "쓰기 가능(writable)"으로 변경되지 않도록 할 수 있다.

[0049] 이와 같이 리소스들을 모니터링하고 보호하기 위해, 가상 기계 모니터내에서 실행되는 코드는 가상 기계 모니터 가로채기 관리자(intercept manager)(예를 들어, 도 1의 관리자(146))를 이용할 수 있다. 이 가로채기 관리자는, 그렇게 지시받은 경우, 식별된 리소스들의 다양한 컴포넌트들에 대한 가로채기를 위해 등록을 할 수 있다. 가상 기계 모니터 내의 보호 에이전트는 이제, 이러한 등록으로 인해, 이 식별된 리소스들에 액세스하거나 그 리소스들을 수정하려는 시도가 있는 경우 가로채기를 수신할 수 있다. 따라서, 보호 에이전트는 식별된 리소스들의 다양한 컴포넌트들을 검사하고 스캔할 수 있다. 보호 에이전트는 또한 이 리소스들을 수정하려는 시도들을 능동적으로 차단할 수 있다.

[0050] 어떤 실시예들에서, 보호 에이전트는, 장래의 스캔들의 결과들을 비교하는 데 사용하기 위해, 리소스들을 스캔하고 리소스들의 초기 상태를 판정한다. 다른 실시예들에서, 보호 에이전트는 장래의 스캔들의 결과들을 비교하기 위해 리소스들의 초기 상태를 이미 알고 있다. 어쨌든, 보호 에이전트는 이 초기 상태의 해쉬 또는 체크섬 값을 계산할 수 있다. 이 계산 후에, 보호 에이전트는 운영 체제가 부팅하기 전에, 부팅한 후에, 또는 부팅하는 동안에 리소스들을 스캔한다. 이 스캔 이후에, 보호 에이전트는 결과들의 해쉬 또는 체크섬을 계산하고 이것을 초기 상태 해쉬 또는 체크섬 값과 비교한다. 이 값들이 같은 경우, 보호 에이전트는 대응하는 리소스들이 변경되지 않은 것으로 판정한다. 물론, 보호 에이전트는 해쉬 또는 체크섬 값을 무시하고 그 대신에 초기 상태를 스캔과 직접 비교할 수 있다.

[0051] 그렇지만, 이 값들이 서로 다른 경우, 보호 에이전트 및/또는 가상 기계 모니터는 하나 이상의 대응 조치들을 취할 수 있다. 먼저, 보호 에이전트 자체는 운영 체제 또는 운영 체제 특권 모드를 종료시킬 수 있거나, 가상 기계 모니터에게 그렇게 하도록 지시할 수 있다. 다시 말하지만, 보호 에이전트가 가상 기계 모니터에 존재하고 가상 기계 모니터가 운영 체제를 호스팅하기 때문에, 이 2개의 컴포넌트는 운영 체제를 그처럼 종료시킬 수 있다. 게다가, 보호 에이전트가 가상 기계 모니터 내에 존재하기 때문에, 운영 체제의 종료가 운영 체제 특권 모드로부터도 변조될 수 없다.

[0052] 운영 체제를 종료시키는 것에 부가하여, 보호 에이전트 및/또는 가상 기계 모니터는 먼저 운영 체제에게 임박한 종료를 경고할 수 있다. 가상 기계 모니터와 운영 체제 간의 통신 채널이 이러한 통신을 가능하게 해줄 수 있다. 다른 대안에서, 보호 에이전트 및/또는 가상 기계 모니터는 메모리 장소에 경고를 기입하거나 운영 체제가 모니터링하는 이벤트를 신호할 수 있다.

[0053] 경고를 받았는지 여부에 관계없이, 운영 체제 종료는 갑작스러운 것(abrupt)이거나 정상적인 것(graceful)일 수 있다. 갑작스런 종료인 경우에, 가상 기계 모니터는 단순히 해쉬 또는 체크섬 값이 상이하다는 것을 안 직후에 운영 체제를 종료시킬 수 있다. 정상적인 종료인 경우에, 가상 기계 모니터는 운영 체제에게 그 자신을 깨끗하게 종료시킬 일정량의 시간을 부여할 수 있다. 이 때, 운영 체제는, 예를 들어, 열린 파일들을 닫을 수 있고 대응하는 데이터를 플러싱(flush)할 수 있다. 운영 체제는 또한 할당된 리소스들을 해제시킬 수 있다. 게다가, 이 종료는 이 2가지 방식 모두를 이용할 수 있다. 예를 들어, 가상 기계 모니터는, 다수의 파티션들을 호스팅하는 경우, 상이한 해쉬 또는 체크섬 값을 갖는 파티션은 즉각 종료시킬 수 있는 반면, 나머지 파티션들은 깨끗이 종료할 수 있게 해준다. 어쨌든, 종료의 방식은 정책에 의해 구성가능할 수 있으며 조정가능할 수

있다.

[0054] 종료 및 대응하는 경고에 부가하여, 보호 에이전트 및/또는 가상 기계 모니터는 식별된 리소스의 무허가 변경에 응답하여 포스트-부팅 동작(post-boot action)을 취할 수 있다. 예를 들어, 가상 기계 모니터 및/또는 보호 에이전트는, 운영 체제의 재부팅 시에, 운영 체제에게 리소스 변경을 통지할 수 있다. 그에 응답하여, 운영 체제는 멀웨어가 실제로 운영 체제 메모리(일부분(208)(예를 들어, 커널) 등) 내에 존재하는지를 검출하기 위해 안티바이러스 스캔(antivirus scan)을 수행할 수 있다. 게다가, 가상 기계 모니터는 운영 체제를 안전 모드로 부팅할 수 있거나, 운영 체제 자체가 안전 모드로 부팅하기로 할 수 있다. 또한, 운영 체제는, 통지에 응답하여, 그 자신이 공격받은 것으로 식별할 수 있고, 따라서 그 자신에 연결되어 있는 어떤 네트워크에도 액세스하지 않을 수 있다.

[0055] *가상 파티션 보호 에이전트*

[0056] 보호 에이전트(예를 들어, 도 1의 보호 에이전트(142))는, 가상 기계 모니터 자체 내에 존재하지 않고, 별도의 가상 파티션(예를 들어, 도 1의 제2 가상 파티션(112))에 존재할 수 있다. 이 실시예들에서, 이러한 별도의 파티션은 가상 기계 모니터의 신뢰 델리게이트(trusted delegate)로서 기능한다. 따라서, 보호 에이전트(142)는 운영 체제 특권 모드로부터 액세스가능하지 않다. 상기한 바와 같이, 가상 기계 모니터(108)는 컴퓨팅 장치(102)의 이러한 가상화를 제공한다. 가상 기계 모니터가 컴퓨팅 장치를 임의의 수의 파티션들로 가상화할 수 있지만, 도 1은 운영 체제를 호스팅하는 제1 파티션 및 보호 에이전트를 호스팅하는 제2 파티션을 나타내고 있다. 보호 에이전트가 존재하는 제2 가상 파티션은, 어떤 경우들에, 1차적 또는 유일한 기능이 보호 에이전트를 실행하는 것인 전용의 보안 파티션일 수 있다. 다른 실시예들에서, 이러한 제2 가상 파티션은 또 하나의 운영 체제를 호스팅하는 것과 같은 부가의 기능들을 수행할 수 있다.

[0057] 제2 가상 파티션 내에 존재하는 보호 에이전트(142)는 가상 기계 모니터 내에 존재하는 보호 에이전트(134)와 관련하여 상기한 것들과 동일한 기능들 중 다수 또는 그 전부를 수행할 수 있다. 즉, 보호 에이전트(142)는 하나 이상의 운영 체제 리소스들(120)의 식별을 능동적으로 또는 수동적으로 수신할 수 있다. 보호 에이전트는, 이 식별에 응답하여, 또다시 보호 에이전트 서비스들(140)을 확장시킬 수 있으며, 이 서비스들(140)은 일반적으로 식별된 리소스들 중 하나 이상이 변경되었는지를 판정하고, 변경된 경우, 대응 조치를 취하는 것을 포함한다. 이 서비스들은 또한 지정된 리소스들의 불변성을 시행하는 것을 포함할 수 있다. 보호 에이전트(142)는 상기한 것들과 유사한 기법들을 통해 이 기능들을 수행할 수 있다.

[0058] 도시된 바와 같이, 보호 에이전트(142)가 제2 가상 파티션 특권 모드(130)로부터는 액세스될 수 있지만, 운영 체제 특권 모드(126)로부터는 액세스될 수 없다. 따라서, 그 결과의 아키텍처는, 멀웨어가 운영 체제의 최저 수준 컴포넌트와 연관된 메모리 일부분(208) 내에 존재하더라도, 보호 에이전트 자체를 운영 체제 내에 위치하는 멀웨어로부터 보호할 수 있게 해준다.

[0059] *자율 보호 에이전트 특권 모드*

[0060] 이 섹션은 보호 에이전트와 연관된 운영 체제 메모리 일부분이 운영 체제 특권 모드로부터 변경되거나 액세스될 수 없게 만들면서, 이 메모리 일부분이 여전히 운영 체제 물리 메모리 공간에 물리적으로 존재할 수 있게 해주는 도구들에 대해 설명한다. 따라서, 이 도구들은 보호 에이전트와 연관된 메모리 일부분은 물론, 운영 체제 특권 모드 내에서 액세스될 수 있는 나머지 메모리에 액세스하는 자율 보호 에이전트 특권 모드를 생성한다. 따라서, 이 특권 모드는 운영 체제 특권 모드보다 더 많은 특권을 갖는다.

[0061] 제1 서브섹션은 가상 기계 모니터에게 보호 에이전트와 연관된 메모리 일부분을 보호하도록 요청함으로써 보호 에이전트 특권 모드를 생성할 수 있는 도구들에 대해 설명한다. 한편, 제2 서브섹션은 물리 프로세서를 다수의 가상 프로세서들(보호 에이전트를 실행하는 전용의 가상 프로세서를 포함함)로 가상화함으로써 보호 에이전트 특권 모드를 생성할 수 있는 도구들에 대해 설명한다.

[0062] *가상 기계 모니터에 대한 보호 요청*

[0063] 이 서브섹션은 보호 에이전트가 보호 에이전트와 연관된 메모리, 따라서 보호 에이전트 자체를 보호하도록 가상 기계 모니터에게 요청할 수 있는 방법에 대해 설명한다. 이와 같이 보호함으로써, 도 1에 나타낸 바와 같이, 보호 에이전트(144)는 보호 에이전트 특권 모드(132)에서 동작한다. 도시된 바와 같이, 보호 에이전트(144)는, 보호 에이전트 특권 모드로 전환하기 전에, 초기에 운영 체제 특권 모드 내에 존재할 수 있다. 보호 에이전트는, 보호 에이전트 특권 모드에서 동작할 때, 일반적으로 운영 체제 특권 모드(126)에서 동작하는 개체들로부터

의 공격에 영향을 받지 않는다.

[0064] 개체는, 보호 에이전트 특권 모드(132)에서 동작할 때, 운영 체제 특권 모드(126)에서 동작하는 경우보다 약간 더 많은 특권을 갖지만, 여전히 가상 기계 모니터 특권 모드(124)보다 더 적은 특권을 갖는다. 도 2에 나타낸 바와 같이, 보호 에이전트 특권 모드에서 동작하는 보호 에이전트는, 보호 에이전트 자체와 연관된 메모리 일부분(206)에 부가하여, 운영 체제와 연관된 메모리의 전부에 액세스한다. 가상 기계 모니터(108)는 부가된 보호 에이전트 액세스 가능성을 시행한다.

[0065] 도 3 및 도 4는 이러한 보호 에이전트 특권 모드를 생성하는 예시적인 방식을 나타낸 것이다. 도 3은 컴퓨팅 장치 메모리(300)의 전부 또는 거의 전부를 나타낸 것이다. 컴퓨팅 장치 메모리(300)는 운영 체제 특권 모드와 연관된 메모리 일부분(302)(예를 들어, 커널) 및 사용자 특권 모드와 연관된 메모리 일부분(304)을 포함한다. 메모리 일부분(302)은 또한, 도시된 바와 같이, 보호 에이전트(144)와 연관된 메모리 일부분(306)은 물론, 드라이버가 로드되는 메모리 일부분(308)을 포함한다.

[0066] 도 4에 나타낸 바와 같이, 보호 에이전트 특권 모드(132)를 생성하는 프로세스(400)는, 동작 1에서, 메모리 일부분(302)(예를 들어, 커널)을 초기화하는 것으로 시작한다. 동작 2에서, 메모리 일부분(306) 또는 보호 에이전트(144) 자체는 보호 에이전트와 연관된 메모리 일부분을 보호하도록 가상 기계 모니터에게 요청하기 위해 가상 기계 모니터(108)를 호출한다. 그와 같이 요청함에 있어서, 보호 에이전트 또는 대응하는 메모리는 운영 체제 특권 모드 내에서 실행되는 코드가 이 메모리 일부분(306)을 변경하거나 다른 방식으로 손대지 말도록 요구한다. 보호 에이전트는 또한 (예를 들어, 디지털 서명에 의해) 가상 기계 모니터(108)에 대해 그 자신을 검증할 수 있다. 메모리의 이 일부분 또는 보호 에이전트 자체는 또한 타이머를 설정하고 타이머가 만료될 때 보호 에이전트를 실행하도록 가상 기계 모니터에게 요청할 수 있다. 동작 3은, 그 요청에 응답하여, 가상 기계 모니터가 운영 체제 특권 모드 내에서 동작하는 개체들로부터 메모리를 보호하고 타이머를 설정하는 것을 나타낸 것이다. 유의할 점은, 보호 에이전트와 연관된 이 메모리 일부분(306)이 이제 운영 체제 특권 모드로부터 변경 및/또는 액세스될 수 없기 때문에, 보호 에이전트가 이제 보호 에이전트 특권 모드에 있다는 것이다.

[0067] 동작 4에서, 드라이버들이 메모리 일부분에 로드된다(308). 유의할 점은, 멀웨어가 드라이버의 형태로 존재할 수 있기 때문에, 동작 2의 요청 및 동작 3의 대응하는 보호가 일반적으로 드라이버들이 메모리에 로드되기 전에 행해진다는 것이다. 이하의 "본 발명의 도구들의 예시적인 용도" 섹션에서 설명하는 바와 같이, 멀웨어 저작자는 종종 사용자를 속여 악의적 드라이버를 컴퓨팅 장치 상에 설치하게 한다. 메모리 일부분(306)이 보호되기 전에 하나 이상의 악의적 드라이버들이 메모리에 실제로 로드되는 경우, 이 악의적 드라이버들은 잠재적으로 보호 요청 자체에 패치될 수 있다. 이와 같이 패치되면 가상 기계 모니터를 통한 보호 에이전트의 주기적인 실행, 따라서 보호 에이전트 특권 모드의 생성을 방해하게 된다. 그렇지만, 조기에 타이머를 설정하도록 가상 기계 모니터에게 요청함으로써, 이 프로세스는 운영 체제 특권 모드 내의 코드가 보호 에이전트의 주기적인 실행을 그처럼 방해하지 못하도록 해준다.

[0068] 한편, 동작 5는 때때로 드라이버들이 로드된 후에 일어날 가능성이 있다. 도시된 바와 같이, 동작 5는 가상 기계 모니터 타이머의 만료, 따라서 보호 에이전트의 실행을 나타낸다. 보호 에이전트(144)는, 실행 중일 때, 이전의 섹션들에서 설명된 것들과 유사하거나 동일한 기능들을 수행한다. 또한, 상기한 바와 같이, 보호 에이전트는 하나 이상의 식별된 리소스들이 변경되었다는 판정에 응답하여 조치를 취할 수 있다. 보호 에이전트는 또한 운영 체제 특권 모드 내에서 동작하는 개체들로부터 보호 에이전트 또는 그의 대응하는 메모리에 액세스하거나 그를 변경하려는 시도에 응답하여 이러한 조치를 취할 수 있다.

[0069] 동작 6은 보호 에이전트가 실행을 종료할 때 가상 기계 모니터에 통지하는 것을 나타낸 것이다. 마지막으로, 동작 7은 동작 3, 동작 5 및 동작 6의 반복을 나타낸 것이다. 따라서, 가상 기계 모니터는 그의 타이머를 리셋하고 주기적 간격으로(예를 들어, 100 밀리초(ms)마다) 보호 에이전트를 실행할 수 있다.

[0070] 가상 기계 모니터에 고장-안전 타이머(fail-safe timer)를 설정함으로써, 프로세스(400)는 운영 체제 코드가 보호 에이전트와 연관된 메모리 일부분을 변조할 수 없게 만든다. 따라서, 이 프로세스는 보호 에이전트가 계속 실행되도록 해주고, 운영 체제 특권 모드 내에서 동작하는 멀웨어에 의해 패치되지 않는다. 그 대신에, 보호 에이전트는 자율 특권 모드 내에서 실행되면서 여전히 운영 체제에 할당된 물리 메모리 내에 존재한다.

[0071] *보호 에이전트 가상 프로세서*

[0072] 이 서브섹션은 가상 기계 모니터가 보호 에이전트(144)를 실행하도록 가상 프로세서를 스케쥴링함으로써 보호 에이전트 특권 모드를 생성할 수 있는 방법에 대해 설명한다. 도 5는 가상 기계 모니터(108)가 컴퓨팅 장치

(102)를 2개의 파티션(각각이 운영 체제를 포함함)으로 가상화하는 것을 포함하는 아키텍처(500)를 나타낸 것이다. 도시된 바와 같이, 컴퓨팅 장치는, 이 예에서, 2개의 실제 프로세서(real processor)(104(a), 104(b))를 포함하며, 이들 프로세서 각각에서 가상 프로세서는 다수의 가상 프로세서들을 스케쥴링할 수 있다. 또한, 도시된 바와 같이, 가상 기계 모니터는 제1 가상 파티션(502) 및 제2 가상 파티션(504)을 생성한다. 제1 가상 파티션은 제1 운영 체제를 실행하는 제1 가상 프로세서(506)를 포함한다. 마찬가지로, 제2 가상 파티션은 제2 운영 체제를 실행하는 제2 가상 프로세서(508)를 포함한다. 그렇지만, 이 경우에, 가상 기계 모니터는 또한 보호 에이전트(도 1의 보호 에이전트(144) 등)를 실행하는 보호 에이전트 가상 프로세서(510)도 포함한다.

[0073] 아키텍처(500)를 생성하기 위해, 가상 기계 모니터가 먼저 로드되고 초기화된다. 도 6에 도시된 바와 같이, 가상 기계 모니터는 이어서 다양한 가상 프로세서들을 가상화하고, 가상화 시에, 실제 프로세서 대역폭(600)을 할당한다. 이러한 가상화 및 할당을 시작하기 위해, 가상 기계 모니터는 제1 가상 프로세서를 제1 실제 프로세서 상으로 가상화한다. 이 예에서, 이러한 가상화는 도 6에 나타낸 바와 같이 일대일 기반으로 행해진다. 즉, 이 단일의 가상 프로세서(506)만이 실제 프로세서(104(a))에 대응하고, 따라서 가상 기계 모니터는 실제 프로세서 대역폭의 전부를 이 가상 프로세서에 할당한다. 이어서, 가상 기계 모니터는 제2 가상 프로세서(508)를 제2 실제 프로세서(104(b)) 상으로 가상화한다. 그렇지만, 일대일 기반 대신에, 가상 기계 모니터는 제2 실제 프로세서 대역폭의 어떤 일부분을 보유하고 있다. 가상 기계 모니터는 이어서, 역시 도 6에 나타낸 바와 같이, 보호 에이전트 가상 프로세서(510)를 제2 실제 프로세서(104(b))의 이러한 나머지 대역폭 상으로 가상화한다.

[0074] 제2 실제 프로세서 상에서 동작하는 각각의 가상 프로세서는 타임-슬라이스 기반(time-sliced basis)으로 동작한다. 즉, 제2 가상 프로세서는 제2 가상 프로세서의 동작이 일시 정지되기 전에 얼마간의 시간 동안 제2 실제 프로세서 상에서 동작할 수 있다. 제2 가상 프로세서의 동작이 일시 정지될 때, 제2 실제 프로세서는 얼마간의 다른 시간 동안 보호 에이전트 가상 프로세서의 동작으로 전환한다. 예를 들어, 제2 가상 프로세서는 90 ms 동안 제2 실제 프로세서 상에서 동작할 수 있으며, 이 시점에서 이 제2 가상 프로세서의 동작이 일시 정지하고 보호 에이전트 가상 프로세서의 동작이 시작되어 10 ms 동안 계속된다. 보호 에이전트 가상 프로세서는 일반적으로 운영 체제 파티션들 둘다와 제1 및 제2 가상 프로세서 둘다에 투명하다. 이와 같이, 운영 체제들 둘다는 자기의 대응하는 가상 프로세서가 각자의 실제 프로세서에 대응하는 것으로 생각한다.

[0075] 실제 프로세서 대역폭을 할당하는 것에 부가하여, 가상 기계 모니터는 또한 각각의 가상 프로세서가 액세스할 수 있는 메모리 일부분도 관리한다. 이 예에서, 제1 가상 프로세서는 제1 운영 체제와 연관된 메모리의 전부에 액세스할 수 있다. 한편, 제2 가상 프로세서는, 보호 에이전트와 연관된 메모리 일부분을 제외한, 제2 운영 체제와 연관된 메모리 전부에 액세스할 수 있다. 보호 에이전트 가상 프로세서만이, 제2 운영 체제에 할당된 메모리에 부가하여, 보호 에이전트와 연관된 메모리 일부분에 액세스한다.

[0076] 게다가, 제1 및 제2 가상 프로세서는 자기의 연관된 메모리를 변경할 수 있을 뿐이다. 따라서, 각자의 운영 체제를 동작시키는 가상 프로세서들 중 어느 것도 보호 에이전트와 연관된 메모리 일부분을 변경할 수 없다. 그렇지만, 보호 에이전트 가상 프로세서는 보호 에이전트와 연관된 메모리를 변경할 수 있고, 어떤 실시예들에서, 제2 가상 프로세서와 연관된 메모리도 변경할 수 있다.

[0077] 보호 에이전트 가상 프로세서는, 그의 프로그램된 특성에 따라, 주기적으로 보호 에이전트를 실행한다. 어떤 경우들에, 보호 에이전트 가상 프로세서가 다른 애플리케이션들을 실행할 수 있지만, 이 예는 전용의 보호 에이전트 가상 프로세서를 나타내고 있다. 따라서, 이 가상 프로세서는 일반적으로 보호 에이전트를 주기적으로 실행하는 역할만 한다. 다시 말하지만, 보호 에이전트는 상기한 보호 에이전트들과 유사하거나 동일한 기능들을 유사하거나 동일한 방식으로 수행할 수 있다.

[0078] 가상 기계 모니터는, 전용의 보호 에이전트 가상 프로세서를 스케쥴링함으로써, 보호 에이전트가 이 프로세서의 제어 하에서 자율 보호 에이전트 특권 모드에서 주기적으로 실행되도록 한다. 게다가, 이 보호 에이전트 가상 프로세서만이 보호 에이전트와 연관된 메모리 일부분에 액세스하기 때문에, 가상 기계 모니터는 이 메모리를 운영 체제 내의 코드로부터 보호한다. 따라서, 운영 체제 특권 모드 내에서 동작하는 멀웨어는 보호 에이전트 상에 패치되어 보호 에이전트가 실행되지 못하게 할 수 없다. 따라서, 이 기법은 본질적으로 운영 체제가 보호 에이전트를 변조할 수 없게 한다.

[0079] *본 발명의 도구들의 예시적인 용도*

[0080] 보호 에이전트의 보호를 보장해줄 수 있는 도구들에 대해 앞서 설명하였고, 이하의 섹션은 이 도구들의 한 예의 동작에 대해 설명한다. 먼저, 컴퓨터 사용자가 인터넷을 서핑하고, 어떤 웹 사이트를 서핑하는 동안, 악의적

의도를 갖는 대화 상자가 사용자의 디스플레이 상에 팝업하는 것으로 생각해보자. 이 대화 상자는 어떤 종류의 멀웨어를 사용자의 컴퓨터 상에 설치할 수 있게 해달라고 사용자에게 요청한다. 이 요청이 직접적일 수 있지만, 통상적으로 그러하듯이, 대화 상자가 그 요청을 변장시키고 있는 것으로 생각해보자. 이 대화 상자는, 예를 들어, 거짓으로 사용자에게 사용자가 당첨되었다고 알려줄 수 있다. 대화 상자는, 그것을 알려줄 때, 악의적으로 사용자가 상품을 받기 위해 대화 상자 상의 "OK" 버튼을 클릭하도록 지시한다. 사용자가 실제로 OK 버튼을 선택하고 컴퓨팅 장치 상에서 실행되고 있는 소프트웨어(예를 들어, 안티바이러스 애플리케이션)로부터의 하나 이상의 경고에도 불구하고 요청된 동작들을 계속하기로 하는 것으로 생각해보자.

[0081] 이 때, 컴퓨팅 장치는 멀웨어를 포함하는 드라이버의 설치를 시작한다. 드라이버의 경우 일반적으로 그러한 바와 같이, 이러한 악의적인 드라이버는 운영 체제 특권 모드에의 액세스가 허여되고 이 특권 모드와 연관된 메모리(예를 들어, 커널)에 로드된다. 악의적인 드라이버 및 그에 수반하는 멀웨어는, 일단 커널에 로드되면, 본질적으로 컴퓨터의 메모리 및 운영 체제에 마음대로 액세스한다. 사용자에게는 불행하게도, 이 멀웨어가 사용자의 키 스트로크를 로그하는 키 로거(key logger)를 포함하는 것으로 생각해보자. 이제, 사용자가 그의 은행 웹사이트로 이동하여 그의 은행 계좌에 로그인하는 것으로 생각해보자. 키 로거는, 키 스트로크를 로그할 수 있기 때문에, 사용자의 은행 계좌 비밀 번호를 알아내고 이 비밀 번호를 인터넷을 통해 악의적 드라이버의 작성자에게로 송출할 수 있다.

[0082] 이 상황을 더욱 악화시키기 위해, 멀웨어가 "루트킷"(즉, 보호 에이전트 및 사용자의 안티바이러스 소프트웨어가 모르게 능동적으로 숨으려고 하는 멀웨어)인 것으로 생각해보자. 종래의 시스템에서, 보호 에이전트는 커널 내에(즉, 악의적인 드라이버가 액세스하는 메모리에) 존재한다. 따라서, 이들 종래의 시스템에서, 멀웨어는 보호 에이전트에 액세스하고, 보호 에이전트에게 그 자신을 숨기려고 시도할 수 있다. 멀웨어는, 성공한 경우, 보호 에이전트에 대해 커널 내에 존재하지 않는 것으로 보이게 된다. 따라서, 사용자의 안티바이러스 소프트웨어가 보호 에이전트를 호출하여 컴퓨터의 메모리에 존재하는 모든 애플리케이션의 리스트를 요청할 때, 멀웨어는 리스트에 존재하지 않게 된다. 이와 같이 리스트에 존재하지 않기 때문에, 안티바이러스 소프트웨어는 멀웨어를 찾아내어 제거하는 데 아무 소용도 없게 된다. 게다가, 멀웨어는 보호 에이전트 상에 패치됨으로써, 보호 에이전트가 전혀 실행되지 못하게 한다. 따라서, 보호 에이전트는 멀웨어가 운영 체제 리소스들을 변경하는지를 알아채지 못할 수도 있다.

[0083] 그렇지만, 사용자의 컴퓨팅 장치 상의 보호 에이전트가, 종래의 시스템에서와 같이 커널 내에 존재하지 않고, 운영 체제 특권 모드로부터 액세스할 수 없는 메모리에 존재하거나 운영 체제 특권 모드로부터 액세스할 수 없는 모드에서 실행되는 것으로 생각해보자. 따라서, 악의적인 드라이버는, 커널 내에 로드될 때, 보호 에이전트가 존재하는 메모리에 또는 보호 에이전트가 실행되는 모드에 액세스하지 못한다. 따라서, 이 드라이버 및 그의 수반하는 멀웨어는 보호 에이전트 자체에 액세스하지 못한다. 따라서, 멀웨어는 보호 에이전트에게, 따라서 안티바이러스 소프트웨어에게도 그 자신을 숨길 수 없다. 따라서, 안티바이러스 소프트웨어가 보호 에이전트에게 컴퓨터의 메모리에 존재하는 모든 애플리케이션의 리스트를 요구할 때, 반환된 리스트는 멀웨어를 포함한다. 그러면, 안티바이러스 소프트웨어는 이 코드를 멀웨어인 것으로 인식하고, 그에 따라 그 코드를 사용자의 컴퓨터 장치로부터 제거한다. 게다가, 보호 에이전트 자체는 멀웨어가 운영 체제 리소스들을 변경하는지를 알아챌 수 있고, 그에 응답하여 사용자의 컴퓨팅 장치를 종료시킬 수 있다.

[0084] 따라서, 본 명세서에 기술된 실시예들은, 운영 체제 특권 모드로부터 액세스할 수 없는 메모리에 존재하거나 운영 체제 특권 모드로부터 액세스할 수 없는 모드에서 실행됨으로써, 멀웨어가 보호 에이전트에게 그 자신을 숨기거나 보호 에이전트 상에 패치되지 못하게 한다. 상기 예에서, 사용자의 컴퓨팅 장치는 그에 따라 멀웨어를 기계로부터 제거할 수 있거나, 어떤 경우들에, 멀웨어가 중요한 리소스들을 변경할 때 시스템을 종료시킬 수 있다. 어느 경우든지 간에, 이 실시예들은 멀웨어가 해를 끼치고자 하는 경우 멀웨어의 효과를 감소시키는 역할을 한다.

[0085] *본 발명의 도구들의 다른 실시예들*

[0086] 이상의 섹션들은 보호 에이전트가 운영 체제 특권 모드로부터 변경되거나 액세스될 수 없게 되어 있는 몇몇 특정의 예들에 대해 설명한다. 이 섹션에서, 하부의 프로세서 상에 존재하지 않는 특권 모드를 프로세서에 추가하는 것과 같은 본 발명의 도구들의 다른 실시예들에 대해 설명한다.

[0087] 이 예시적인 실시예들은 도 7 내지 도 11의 프로세스(700) 내지 프로세스(1100)의 일부로서 설명되어 있다. 이 프로세스들은 물론 도 1 내지 도 6을 참조하여 기술 또는 설명되어 있는 예시적인 프로세스들이 임의의 적당한 하드웨어, 소프트웨어, 펌웨어, 또는 이들의 조합으로서 구현될 수 있고, 소프트웨어와 펌웨어의 경우에, 이 프

로세스들은 컴퓨터 판독가능 매체에 저장되어 하나 이상의 프로세서들에 의해 실행될 수 있는 컴퓨터 실행가능 명령어들로서 구현되는 동작들의 세트이다. 이 섹션에서 설명되는 도구들의 이 실시예들은 본 발명의 도구들 또는 청구항들의 범위를 제한하기 위한 것이 아니다.

[0088] 도 7을 참조하면, 블록(702)은 하나 이상의 운영 체제 리소스들을 식별해주는 시행 정책(enforcement policy)을 수신한다. 암호화된 데이터를 포함할 수 있는 이 시행 정책은 디지털 서명된 매니페스트를 통해 또는 API(application program interface)를 운영 체제에 노출시킴으로써(예를 들어, 하이퍼콜에 의해) 수신될 수 있다. 블록(704)은, 운영 체제 특권 모드 내에서 동작하는 개체로부터 액세스될 수 없는 메모리로부터, 하나 이상의 운영 체제 리소스들을 식별한다. 예시적인 리소스들로는 SSDT(system service dispatch table), IDT(interrupt dispatch table), 및/또는 GDT(global descriptor table)가 있다. 상기한 바와 같이, 이 식별은 가상 기계 모니터 내에서(예를 들어, 도 1의 보호 에이전트(134)에 의해) 또는 별도의 가상 파티션 내에서(예를 들어, 도 1의 보호 에이전트(142)에 의해) 행해질 수 있다.

[0089] 한편, 블록(706)은 식별된 리소스들 중 임의의 리소스가 변경되었는지를 판정하는 것을 나타낸다. 다시 말하지만, 이것은 가상 기계 모니터 내에서 또는 별도의 파티션 내에서 행해질 수 있다. 블록(706)이 식별된 리소스들 중 하나 이상이 실제로 변경된 것으로 판정하는 경우, 블록(708)은 이 판정에 응답하여 운영 체제를 종료시킨다. 마지막으로, 블록(710)은 운영 체제의 재부팅 시에 불법적인 동작을 운영 체제에게 통지한다.

[0090] 도 8은 보호 에이전트가 가상 기계 모니터 내에서 실행될 수 있게 해주는 프로세스(800)를 나타낸 것이다. 블록(802)은 운영 체제 리소스와 연관된 메모리 페이지 또는 레지스터가 변경되었다는 표시를 수신할 수 있게 해주는 데 효과적인 가상 기계 모니터 가로채기 관리자를 변경한다. 이 리소스는 도 7을 참조하여 기술한 리소스들 중 하나를 포함할 수 있거나, 다른 운영 체제 리소스일 수 있다. 어쨌든, 블록(804)은 운영 체제 리소스 및 아마도 하나 이상의 다른 운영 체제 리소스들을 식별해주는 시행 정책을 수신한다. 다시 말하지만, 이 식별은 상기한 기법들을 통해 행해질 수 있다. 상기한 바와 같이, 리소스의 보호 속성(예를 들어, "읽기 전용" 또는 "초기화 읽기 전용")이 리소스의 식별에 수반할 수 있다. 한편, 블록(806)은 운영 체제 리소스와 연관된 메모리 페이지 또는 레지스터가 실제로 변경되었다는 표시를 수신하는 것을 나타낸다. 그에 응답하여, 블록(808)은 운영 체제 리소스와 연관된 운영 체제를 종료시키는 데 효과적인 운영 체제 특권 모드를 종료시킨다. 어떤 경우들에, 도 1의 가상 기계 모니터(108)가 운영 체제 특권 모드의 이러한 종료를 완수할 수 있다.

[0091] 그 다음에, 도 9는 도 1에 나타낸 보호 에이전트 특권 모드(132) 등의 보호 에이전트 특권 모드를 생성하는 예시적인 프로세스(900)에 대해 설명한다. 블록(902)은 메모리의 특정 범위가 운영 체제 특권 모드로부터 변경되거나 액세스될 수 없게 하라는 요청을 수신한다. 다시 말하지만, 가상 기계 모니터가 이러한 요청을 수신할 수 있으며, 이러한 요청은 메모리의 그 범위 자체로부터 또는 메모리의 그 범위에 존재하는 보호 에이전트로부터 온 것일 수 있다. 블록(904)은 메모리의 그 영역을 보호하고, 메모리의 그 영역에 존재하는 보호 에이전트를 주기적으로 실행하도록 타이머를 설정한다. 다시 말하지만, 가상 기계 모니터는 이러한 타이머를 설정할 수 있으며, 이 타이머는 가상 기계 모니터에게 보호 에이전트를 규칙적인 간격으로 실행하도록 지시할 수 있다.

[0092] 한편, 블록(906)은 운영 체제 리소스에 대해 설명하는 시행 정책을 수신한다. 다시 말하지만, 시행 정책 및 기술된 리소스는 상기한 것들과 유사하거나 동일할 수 있다. 블록(908)은 보호 에이전트를 실행하고, 이는 가상 기계 모니터에 의해 달성될 수 있다. 결정 블록(910)은 운영 체제 리소스가 변경되었는지를 질문한다. 보호 에이전트는 이상에서 상세히 설명한 방식으로 기능함으로써 이 판정을 할 수 있다. 블록(910)이 변경이 일어난 것으로 실제로 판정하는 경우, 블록(912)은 운영 체제를 종료시킨다. 그렇지만, 이와 같이 판정되지 않은 경우, 블록(914)은 보호 에이전트가 실행을 완료했다는 통지를 수신한다. 어떤 경우들에, 상기한 바와 같이, 보호 에이전트 자체가 가상 기계 모니터에게 그렇게 통지할 수 있다. 한편, 블록(916)은 보호 에이전트를 실행하는 것과 보호 에이전트를 실행하지 않는 것 간의 순환을 나타낸다. 마지막으로, 유의할 점은, 보호 에이전트가 실행되지 않는 동안에, 운영 체제 특권 모드 내에서 동작하는 개체로부터 보호 에이전트와 연관된 메모리의 그 범위에 액세스하려는 시도에 응답하여, 가상 기계 모니터가 운영 체제를 종료시킬 수 있다는 것이다.

[0093] 도 10은 도 1에 도시된 보호 에이전트 특권 모드(132) 등의 보호 에이전트 특권 모드를 생성하는 다른 예시적인 프로세스(1000)를 나타낸 것이다. 블록(1002)은 실제 컴퓨터 프로세서를 다수의 가상 컴퓨터 프로세서들로 가상화한다. 이 가상 프로세서들은 하나 이상의 운영 체제 가상 프로세서를 포함할 수 있고, 이 운영 체제 가상 프로세서는 각각이 그 자신의 운영 체제 메모리를 변경하고, 도 6에 나타낸 바와 같이, 실제 프로세서의 처리 대역폭의 일부분을 사용할 권한을 갖는다. 이 가상 프로세서들은 또한 적어도 하나의 보호 에이전트 가상 프로세서를 포함할 수 있고, 이 보호 에이전트 가상 프로세서는 그 자신의 보호 에이전트 메모리를 변경하고 실제

프로세서의 처리 대역폭의 다른 일부분을 사용할 특권을 갖는다. 가상 프로세서들 전부가 가상 기계 모니터에 의해 스케쥴링될 수 있는 반면, 보호 에이전트 가상 프로세서는 운영 체제 가상 프로세서들에 대해 투명할 수 있다. 어떤 경우들에, 운영 체제 가상 프로세서는 보호 에이전트 가상 프로세서에 할당된 메모리를 변경할 수 없을지도 모른다. 게다가, 보호 에이전트 가상 프로세서는 전용의 프로세서일 수 있으며 그의 1차적 또는 유일한 목적은, 상기한 바와 같이, 보호 에이전트가 실행되게 하는 것이다.

[0094] 그 다음에, 블록(1004)은 보호 에이전트 가상 프로세서로 하여금 보호 에이전트를 실행하게 하며, 이는 상기 운영 체제 메모리 일부분이 변경되었는지 여부를 판정하는 데 효과적일 수 있다. 한편, 블록(1006)은 운영 체제 메모리 일부분이 변경되었다는 표시를 수신한다. 그에 응답하여, 블록(1008)은 대응하는 운영 체제를 종료시킨다.

[0095] 마지막으로, 도 11은 실제 컴퓨팅 프로세서에 특권 모드를 추가하는 프로세스(1100)를 나타낸 것이다. 블록(1102)은 하부의 물리 프로세서 상에 존재하는 하나 이상의 특권 모드를 판정, 식별, 또는 분류하는 것을 나타낸다. 이 특권 모드들은 일반적으로 하부의 물리 프로세서 자체에 의해 정의된다. 어쨌든, 블록(1104)은 하부의 물리 프로세서 상에 존재하지 않는 특권 모드를 추가한다. 어떤 경우들에, 추가된 특권 모드는 하나 이상의 존재하는 특권 모드들에 의해 변경될 수 있는 메모리 일부분과 다른, 컴퓨팅 장치의 메모리 일부분을 변경할 수 있다. 추가된 특권 모드는 또한 이전에 존재하지 않았거나 하부의 프로세서에서 실행될 수 없었던 명령어들을 추가하여 실행할 수 있다.

[0096] 게다가, 하부의 물리 프로세서 상에 존재하는 하나 이상의 특권 모드들은 사용자 특권 모드 및 운영 체제 특권 모드를 포함할 수 있다. 이 실시예들에서, 추가된 특권 모드는 사용자 특권 모드 및 운영 체제 특권 모드 둘다보다 더 많은 특권을 가질 수 있고 사용자 특권 모드보다 더 많은 특권을 가질 수 있지만, 운영 체제 특권 모드보다 더 적은 특권을 갖거나 사용자 특권 모드 및 운영 체제 특권 모드 둘다보다 더 적은 특권을 가질 수 있다. 마지막으로, 유의할 점은, 특권 모드를 추가하는 한 경우가 상기한 다수의 방식으로 보호 에이전트 특권 모드(예를 들어, 도 1에 나타낸 보호 에이전트 특권 모드(132))를 추가하는 것을 포함할 수 있다는 것이다. 예를 들어, 보호 에이전트 또는 그의 연관된 메모리의 범위는 메모리의 그 범위가 운영 체제 특권 모드 내에서 동작하는 개체들로부터 액세스될 수 없게 하라고 요청할 수 있다. 가상 기계 모니터는 또한 보호 에이전트를 실행하도록 보호 에이전트 가상 프로세서를 스케쥴링함으로써 이러한 특권 모드를 생성할 수 있다.

산업상 이용 가능성

[0097] 결론

[0098] 상기한 도구들은, 보호 에이전트가 운영 체제 특권 모드로부터 액세스될 수 없는 장소에 존재할 수 있게 함으로써, 또는 보호 에이전트 특권 모드를 생성함으로써, 보호 에이전트를 운영 체제 특권 모드로부터 변경되거나 액세스될 수 없게 만들 수 있다. 이 도구들이 구조적 특징 및/또는 방법적 동작과 관련하여 기술되어 있지만, 첨부된 청구항들에 정의된 도구들이 기술된 특정의 특징들 또는 동작들로 꼭 제한되는 것은 아니라는 것을 잘 알 것이다. 오히려, 이 특정의 특징들 및 동작들은 이 도구들을 구현하는 예시적인 형태로서 기술된 것이다.

도면의 간단한 설명

[0009] 도 1은 본 발명의 도구들의 여러 가지 실시예들이 동작할 수 있는 예시적인 동작 환경을 나타낸 도면.

[0010] 도 2는 도 1에 예시된 모듈들의 다양한 컴퓨팅 장치 메모리 권한을 나타낸 도면.

[0011] 도 3은 도 1에 예시된 모듈들 중 몇몇 모듈이 존재하는 컴퓨팅 장치 메모리의 다양한 부분들을 나타낸 도면.

[0012] 도 4는 가상 기계 모니터가 보호 에이전트와 연관된 메모리 일부분을 보호하고 보호 에이전트를 실행하기 위해 타이머를 설정할 수 있는 예시적인 방식을 나타낸 흐름도.

[0013] 도 5는 물리 프로세서들을 다수의 운영 체제 가상 프로세서들 및 하나의 보호 에이전트 가상 프로세서로 가상화할 수 있는 가상 기계 모니터를 갖는 예시적인 아키텍처를 나타낸 도면.

[0014] 도 6은 도 5의 물리 프로세서들의 대역폭이 다수의 가상 프로세서들 간에 할당될 수 있는 방법를 나타낸 도면.

[0015] 도 7은 본 발명의 도구들이 운영 체제 특권 모드로부터 액세스가능하지 않은 장소에 존재하는 보호 에이전트를

작동시켜 실행할 수 있는 몇몇 방식들을 설명하는 예시적인 프로세스를 나타낸 도면.

[0016] 도 8은 본 발명의 도구들이 운영 체제 특권 모드로부터 액세스가능하지 않은 장소에 존재하는 보호 에이전트를 작동시켜 실행하도록 가상 기계 모니터를 변경할 수 있는 몇몇 방식들을 설명하는 예시적인 프로세스를 나타낸 도면.

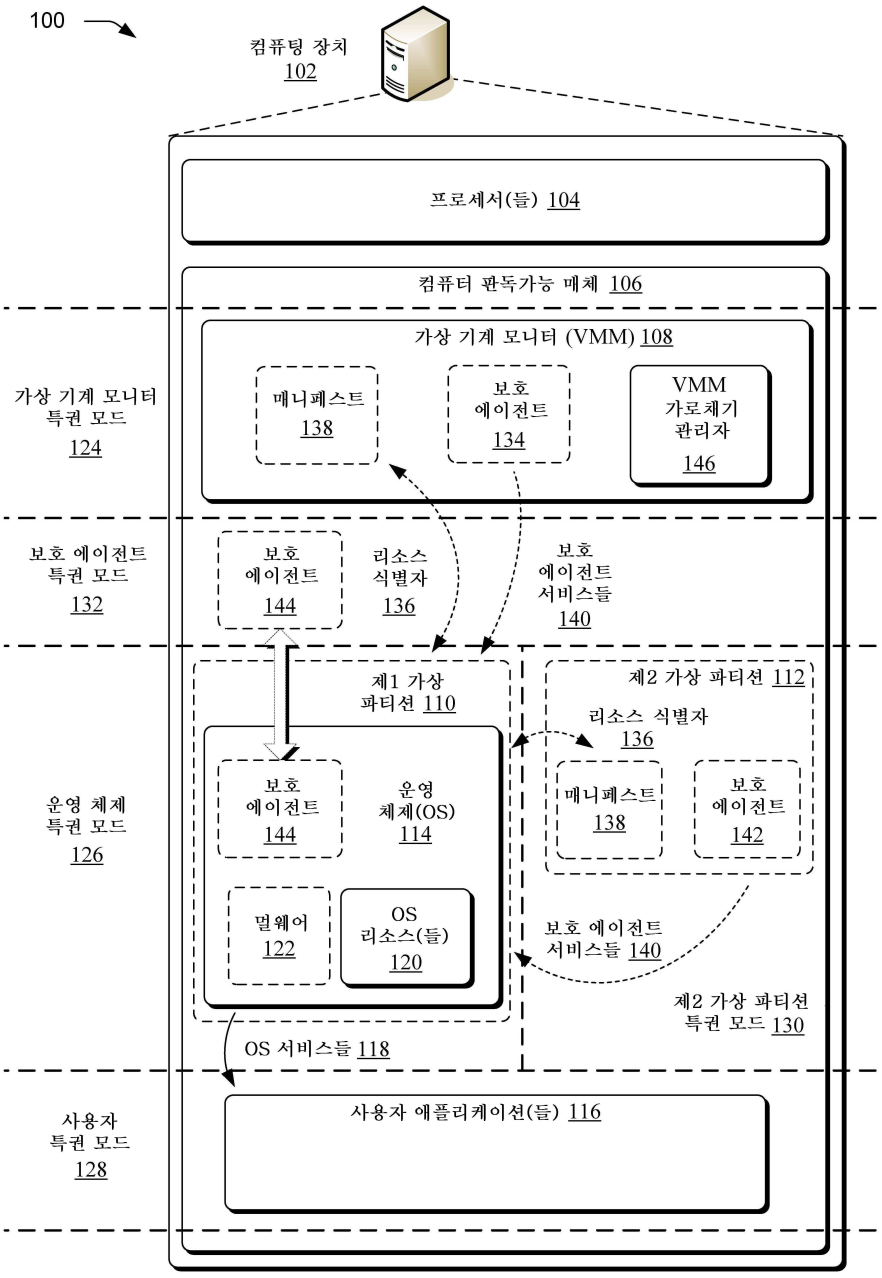
[0017] 도 9는 본 발명의 도구들이 가상 기계 모니터에 요청을 함으로써 보호 에이전트 특권 모드를 생성할 수 있는 몇몇 방식들을 설명하는 예시적인 프로세스를 나타낸 도면.

[0018] 도 10은 본 발명의 도구들이 실제 컴퓨터 프로세서를 가상 컴퓨터 프로세서들(이들 중 적어도 하나가 보호 에이전트를 실행해야 함)로 가상화함으로써 보호 에이전트 특권 모드를 생성할 수 있는 몇몇 방식들을 설명하는 예시적인 프로세스를 나타낸 도면.

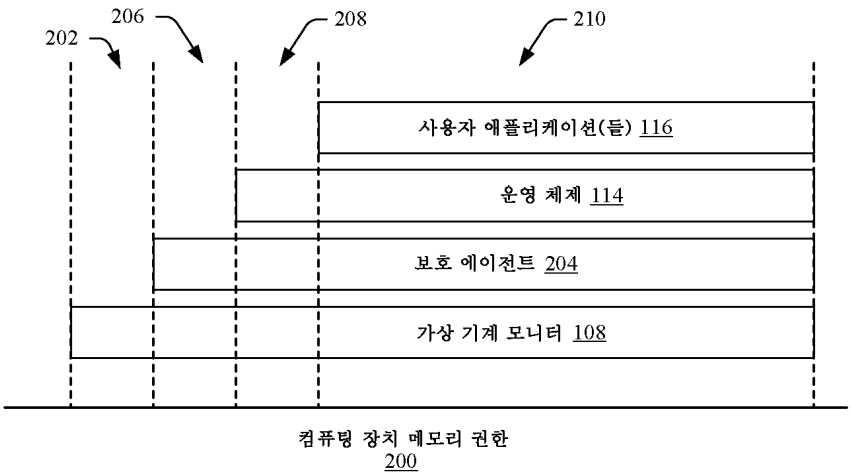
[0019] 도 11은 본 발명의 도구들이 하부의 물리 프로세서 상에 존재하지 않는 특권 모드의 추가를 가능하게 해줄 수 있는 몇몇 방식들을 설명하는 예시적인 프로세스를 나타낸 도면.

도면

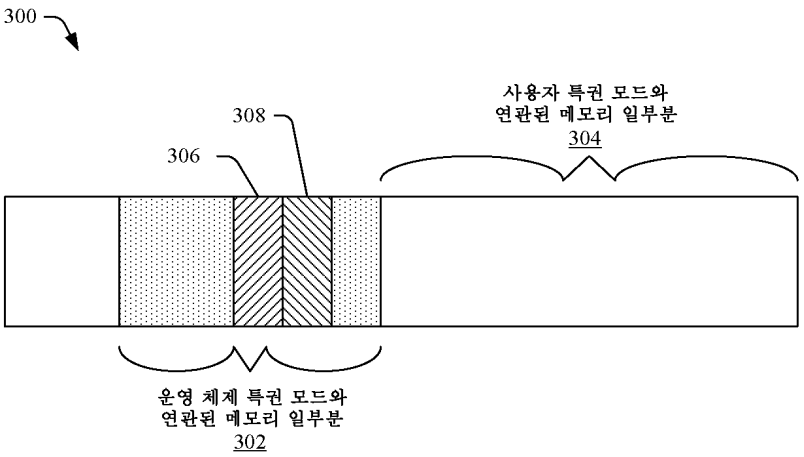
도면1


100
컴퓨팅 장치
102
프로세서(들) 104
컴퓨터 판독가능 매체 106
가상 기계 모니터 (VMM) 108
매니페스트
138
보호
에이전트
134
VMM
가로채기
관리자
146
가상 기계 모니터
특권 모드
124
보호 에이전트
특권 모드
132
보호
에이전트
144
리소스
식별자
136
보호
에이전트
서비스들
140
제1 가상
파티션 110
제2 가상 파티션 112
리소스 식별자
136
운영 체제
특권 모드
126
보호
에이전트
144
운영
체제(OS)
114
매니페스트
138
보호
에이전트
142
멀웨어
122
OS
리소스(들)
120
보호 에이전트
서비스들 140
제2 가상 파티션
특권 모드 130
OS 서비스들 118
사용자
특권 모드
128
사용자 애플리케이션(들) 116

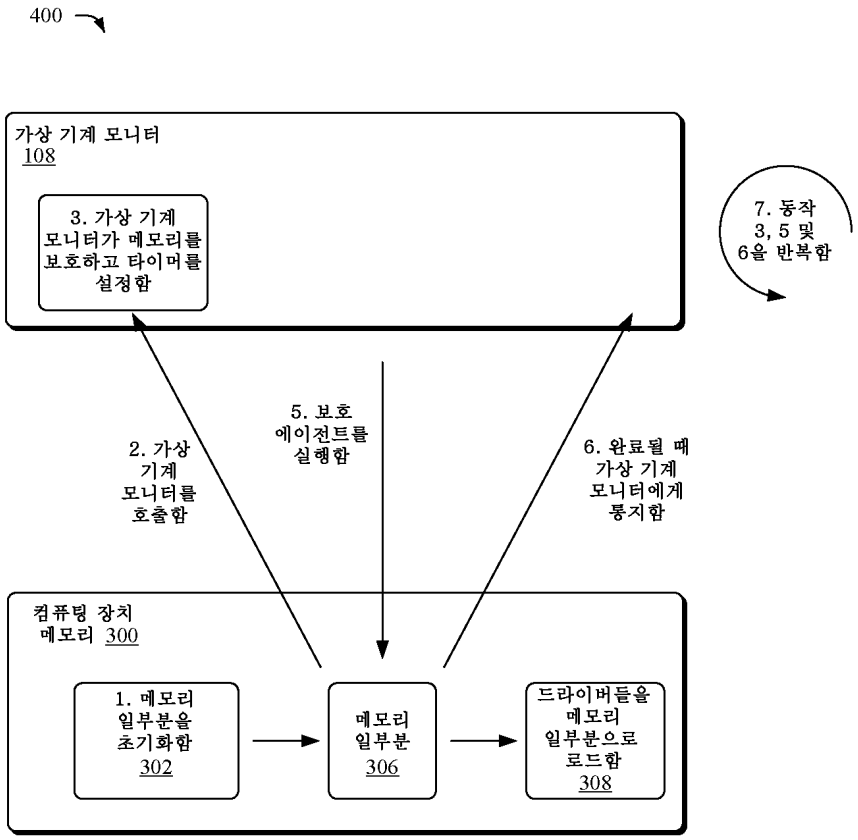
도면2


202
206
208
210
사용자 애플리케이션(들) 116
운영 체제 114
보호 에이전트 204
가상 기계 모니터 108
컴퓨팅 장치 메모리 권한
200

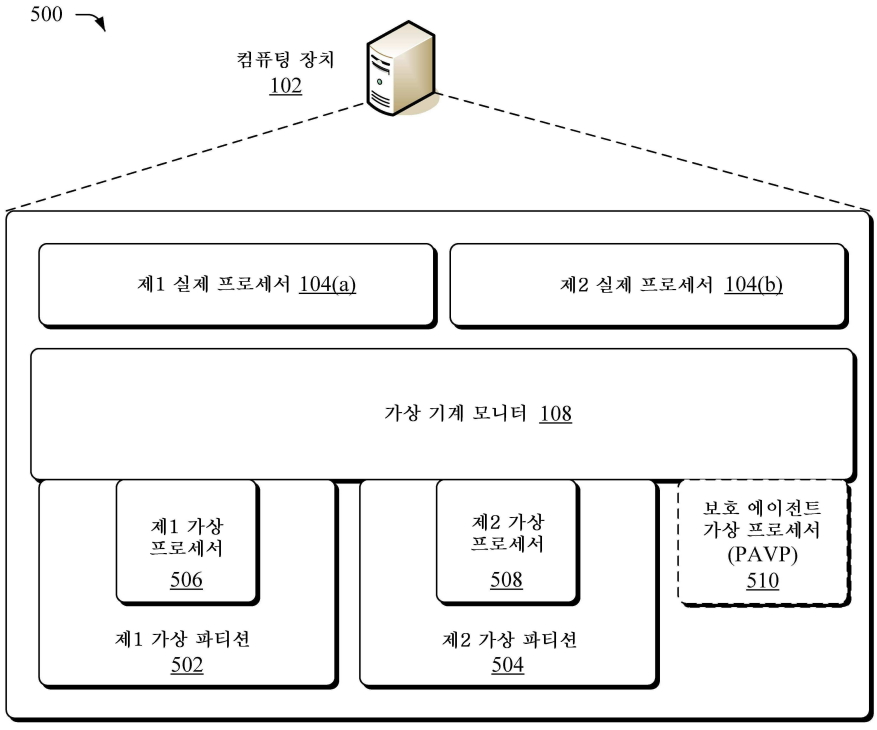
도면3


300
308
306
사용자 특권 모드와
연관된 메모리 일부분
304
운영 체제 특권 모드와
연관된 메모리 일부분
302

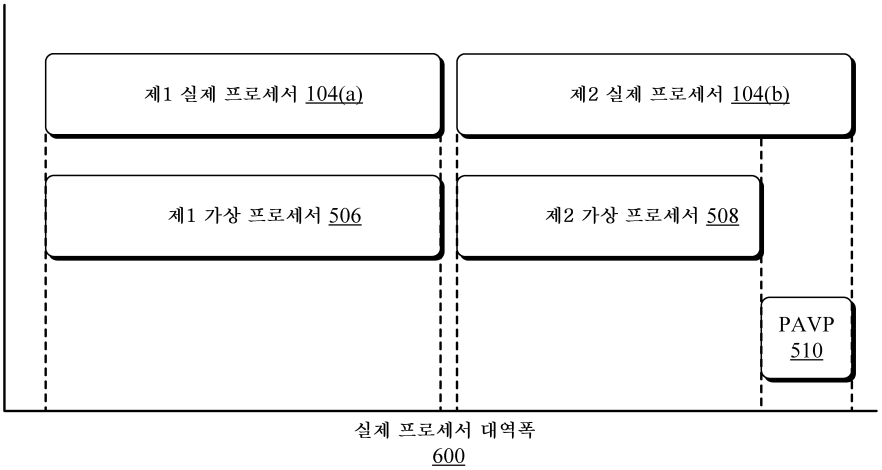
도면4


400
가상 기계 모니터
108
3. 가상 기계 모니터가 메모리를 보호하고 타이머를 설정함
7. 동작 3, 5 및 6을 반복함
2. 가상 기계 모니터를 호출함
5. 보호 에이전트를 실행함
6. 완료될 때 가상 기계 모니터에게 통지함
컴퓨팅 장치 메모리 300
1. 메모리 일부분을 초기화함 302
메모리 일부분 306
드라이버들을 메모리 일부분으로 로드함 308

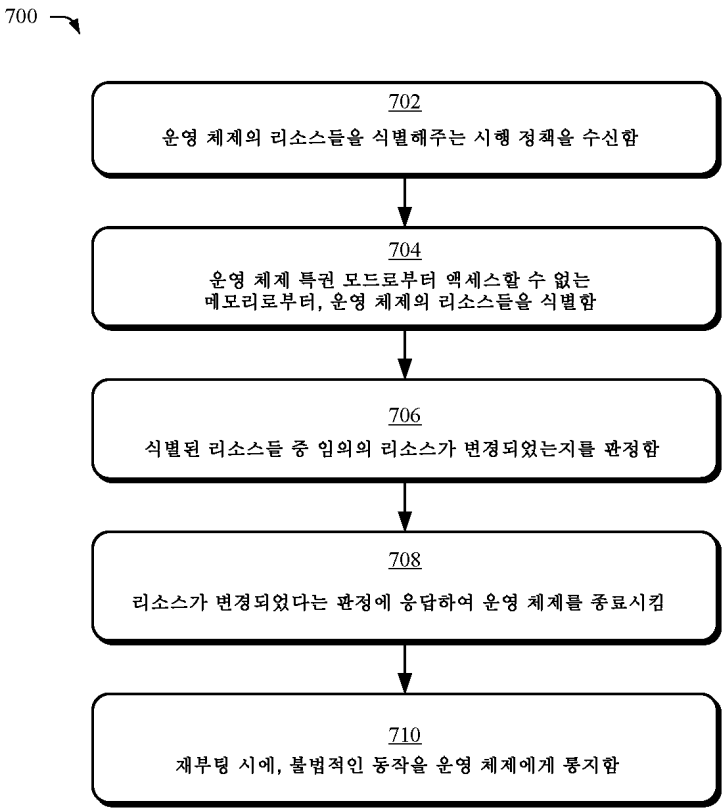
도면5


500
컴퓨팅 장치 102
제1 실제 프로세서 104(a)
제2 실제 프로세서 104(b)
가상 기계 모니터 108
제1 가상 프로세서 506
제2 가상 프로세서 508
보호 에이전트 가상 프로세서 (PAVP) 510
제1 가상 파티션 502
제2 가상 파티션 504

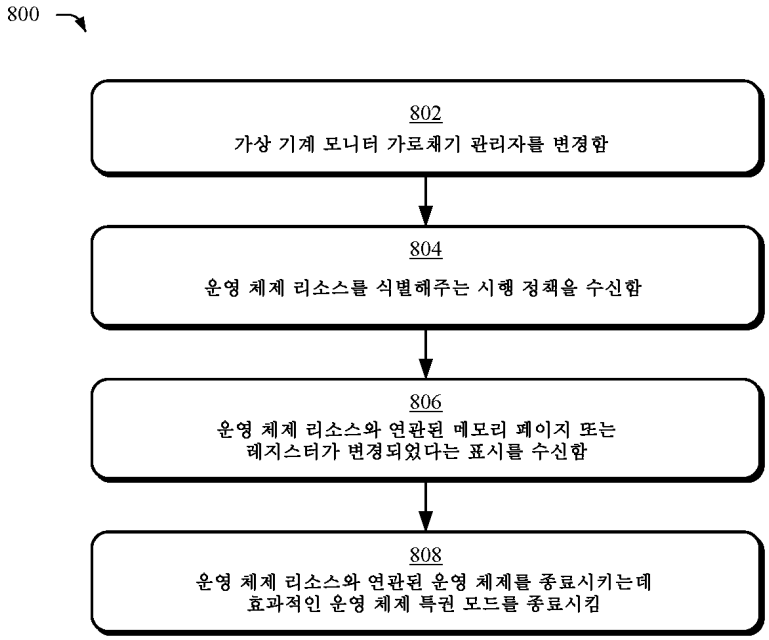
도면6


제1 실제 프로세서 104(a)
제2 실제 프로세서 104(b)
제1 가상 프로세서 506
제2 가상 프로세서 508
PAVP
510
실제 프로세서 대역폭
600

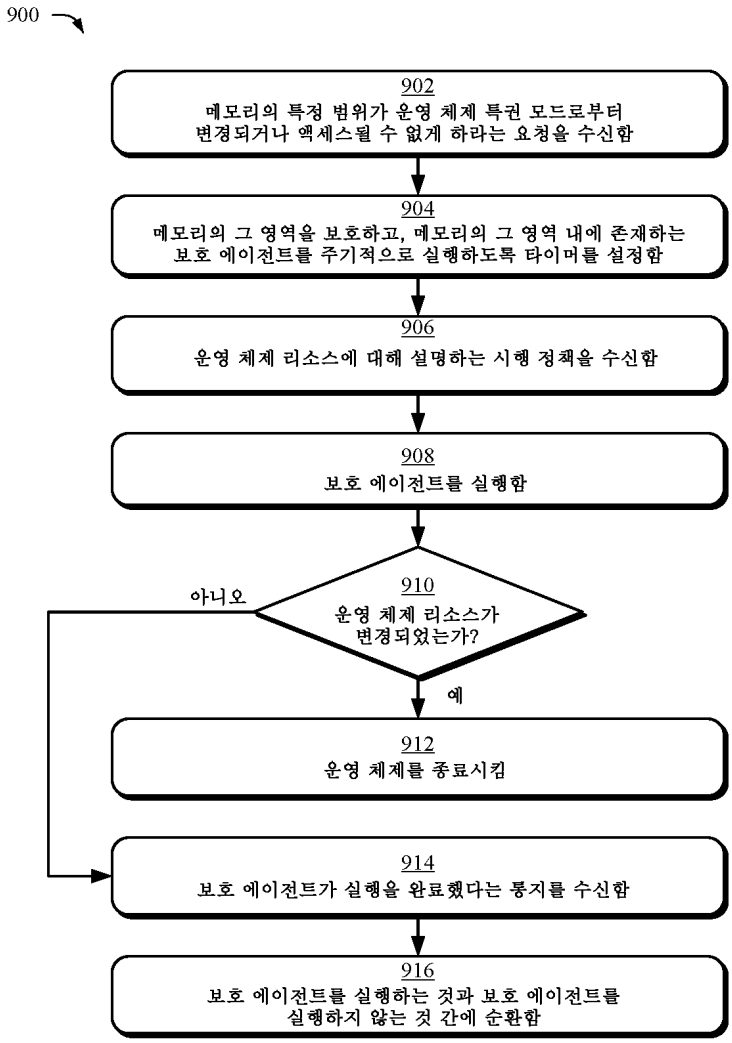
도면7


700
702
운영 체제의 리소스들을 식별해주는 시행 정책을 수신함
704
운영 체제 특권 모드로부터 액세스할 수 없는
메모리로부터, 운영 체제의 리소스들을 식별함
706
식별된 리소스들 중 임의의 리소스가 변경되었는지를 판정함
708
리소스가 변경되었다는 판정에 응답하여 운영 체제를 종료시킴
710
재부팅 시에, 불법적인 동작을 운영 체제에게 통지함

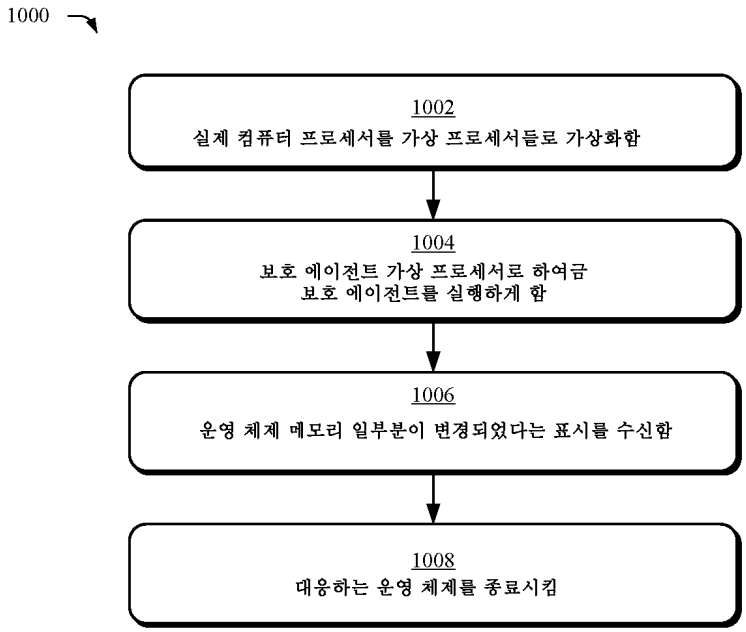
도면8


800
802
가상 기계 모니터 가로채기 관리자를 변경함
804
운영 체제 리소스를 식별해주는 시행 정책을 수신함
806
운영 체제 리소스와 연관된 메모리 페이지 또는
레지스터가 변경되었다는 표시를 수신함
808
운영 체제 리소스와 연관된 운영 체제를 종료시키는데
효과적인 운영 체제 특권 모드를 종료시킴

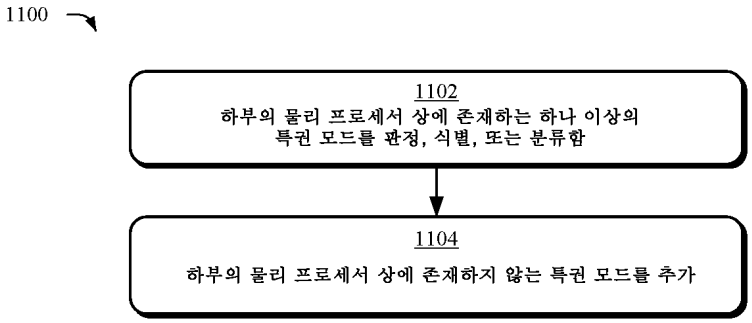
도면9


900
902
메모리의 특정 범위가 운영 체제 특권 모드로부터 변경되거나 액세스될 수 없게 하라는 요청을 수신함
904
메모리의 그 영역을 보호하고, 메모리의 그 영역 내에 존재하는 보호 에이전트를 주기적으로 실행하도록 타이머를 설정함
906
운영 체제 리소스에 대해 설명하는 시행 정책을 수신함
908
보호 에이전트를 실행함
910
운영 체제 리소스가 변경되었는가?
아니오
예
912
운영 체제를 종료시킴
914
보호 에이전트가 실행을 완료했다는 통지를 수신함
916
보호 에이전트를 실행하는 것과 보호 에이전트를 실행하지 않는 것 간에 순환함

도면10


1000
1002
실제 컴퓨터 프로세서를 가상 프로세서들로 가상화함
1004
보호 에이전트 가상 프로세서로 하여금
보호 에이전트를 실행하게 함
1006
운영 체제 메모리 일부분이 변경되었다는 표시를 수신함
1008
대응하는 운영 체제를 종료시킴

도면11


1100
1102
하부의 물리 프로세서 상에 존재하는 하나 이상의
특권 모드를 판정, 식별, 또는 분류함
1104
하부의 물리 프로세서 상에 존재하지 않는 특권 모드를 추가