(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2014년09월23일
(11) 등록번호 10-1443932
(24) 등록일자 2014년09월17일

(51) 국제특허분류(Int. Cl.)
G06F 11/22 (2006.01) *G06F 11/30* (2006.01)
G06F 15/16 (2006.01)
(21) 출원번호 10-2009-7013768
(22) 출원일자(국제) 2007년11월30일
심사청구일자 2012년11월16일
(85) 번역문제출일자 2009년07월01일
(65) 공개번호 10-2009-0095627
(43) 공개일자 2009년09월09일
(86) 국제출원번호 PCT/US2007/086195
(87) 국제공개번호 WO 2008/070587
국제공개일자 2008년06월12일
(30) 우선권주장
11/566,170 2006년12월01일 미국(US)
(56) 선행기술조사문헌
US20050114285 A1
US20060047713 A1
WO2009641495 A1
US6374401 A
전체 청구항 수 : 총 14 항

(73) 특허권자
마이크로소프트 코포레이션
미국 워싱턴주 (우편번호 : 98052) 레드몬드 원 마이크로소프트 웨이
(72) 발명자
베르보우스키, 차드
미국 98052-6399 워싱턴주 레드몬드 원 마이크로소프트 웨이
리, 주한
미국 98052-6399 워싱턴주 레드몬드 원 마이크로소프트 웨이
(뒷면에 계속)
(74) 대리인
제일특허법인

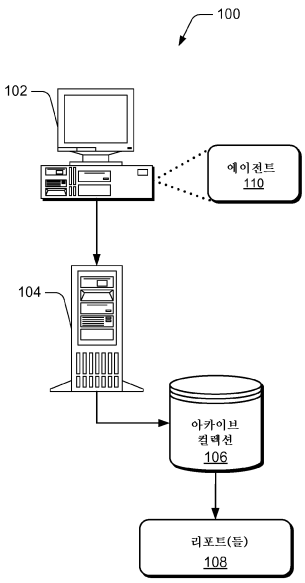
심사관 : 구대성

(54) 발명의 명칭 **시스템 분석 및 관리**

(57) 요 약

하나 이상의 프로그램과 나타나기 쉬운 지속 상태 사이의 상호작용의 검토에 기초하여 시스템 관리를 구현하는 시스템 및 방법이다. 본 시스템은 수정이 인가되었는지를 확인하고, 알려지지 않은 수정을 검출하면서 통지를 생성하여, 시스템 내에서 발생하는 수정을 검출하기 위해 제공된다.

대 표 도 - 도1



(72) 발명자

리우, 시아오강

미국 98052-6399 워싱턴주 레드몬드 원 마이크로소프트 웨이

로우세브, 로우시

미국 98052-6399 워싱턴주 레드몬드 원 마이크로소프트 웨이

왕, 이민

미국 98052-6399 워싱턴주 레드몬드 원 마이크로소프트 웨이

특허청구의 범위

청구항 1

컴퓨팅 장치상에서 명령어들을 실행하도록 구성된 프로세서에 의해 구현되는 방법에 있어서,

상기 명령어들은 상기 프로세서에 의해 실행되는 경우 상기 컴퓨팅 장치로 하여금,

무효 파일들(stale files)을 검출하는 단계;

상기 컴퓨팅 장치에 로딩된 프로그램들의 카탈로그를 작성하는 단계;

상기 컴퓨팅 장치에 등록된 상기 프로그램들과 연관된 파일들의 최종 로드(load) 시간을 획득하는 단계;

상기 컴퓨팅 장치에 등록된 상기 프로그램들과 연관된 파일들의 최종 수정 시간을 획득하는 단계;

상기 컴퓨팅 장치에 등록된 상기 프로그램들과 연관된 파일들의 상기 최종 수정 시간을 상기 최종 로드 시간과 비교하는 단계; 및

상기 비교 동안 발견된 임의의 불일치(inconsistencies)를 기록하는(noting) 단계- 상기 기록은 상기 프로그램이 최종 시도된 수정에 응답하지 않는다라는 보고, 및 상기 프로그램의 상기 최종 시도된 수정을 다시 시도하려는 시도를 포함함 -

를 수행하게 하는, 방법.

청구항 2

제1항에 있어서,

상기 카탈로그를 작성하는 단계는 상기 컴퓨팅 장치의 운영 시스템에 등록된 프로그램들의 목록을 생성하는 단계를 포함하는, 방법.

청구항 3

제1항에 있어서,

상기 카탈로그를 작성하는 단계는 상기 프로그램을 목록에 배치함으로써 상기 프로그램들을 열거하는 단계를 포함하는, 방법.

청구항 4

제1항에 있어서,

상기 카탈로그를 작성하는 단계는 상기 컴퓨팅 장치에 등록된 모든 프로그램들을 검출하도록 스캐닝하는 단계를 포함하는, 방법.

청구항 5

제1항에 있어서,

상기 획득하는 단계는 로그 파일에 쿼리하는 단계를 포함하는, 방법.

청구항 6

제1항에 있어서,

상기 비교하는 단계는 로그 파일에 쿼리하는 단계를 포함하는, 방법.

청구항 7

컴퓨팅 장치로서,

프로세서 유닛;

상기 컴퓨팅 장치에 등록된 프로그램들과 연관된 파일들의 최종 로드 시간과, 상기 컴퓨팅 장치에 등록된 프

로그램들과 연관된 파일들의 최종 수정 시간을 저장하도록 구성된 로그 스토리지 컴포넌트; 및

상기 컴퓨팅 장치에 등록된 상기 프로그램들의 카탈로그를 작성하고, 상기 컴퓨팅 장치 내의 무효 파일들을 검출하기 위해 상기 최종 수정 시간을 상기 최종 로드 시간과 비교하며- 무효 파일들이 존재하면 상기 프로그램들의 업그레이드는 무시되고, 상기 업그레이드가 무시되면 상기 프로그램들은 예전 파일들로부터 계속 실행될 것임 -, 상기 무효 파일들을 사용자 또는 시스템 관리자와 같은 엔티티에 보고하도록 구성된 쿼리 로그 컴포넌트- 상기 보고는 최종 시도된 수정에 응답하지 않는 프로그램과, 상기 프로그램들의 상기 최종 시도된 수정을 다시 시도하려는 시도를 포함함 -를 포함하는

컴퓨팅 장치.

청구항 8

제7항에 있어서,

상기 컴퓨팅 장치에 등록된 상기 프로그램들과 연관된 파일들의 상기 최종 로드 시간 및 상기 최종 수정 시간을 저장하도록 구성된 아카이브 컬렉션 컴포넌트를 더 포함하는, 컴퓨팅 장치.

청구항 9

제7항에 있어서,

상기 최종 로드 시간 및 상기 최종 수정 시간은 상기 최종 로드 시간 및 상기 최종 수정 시간의 날짜를 포함하는, 컴퓨팅 장치.

청구항 10

제7항에 있어서,

상기 쿼리 로그 컴포넌트는 또한 상기 컴퓨팅 장치에 등록된 모든 프로그램들을 스캐닝하도록 구성된, 컴퓨팅 장치.

청구항 11

컴퓨터에 의해 실행되는 경우 방법을 구현하는 컴퓨터 판독가능 명령어들을 저장한 컴퓨터 판독가능 저장 매체에 있어서,

상기 방법은,

컴퓨팅 기반 장치에 등록된 프로그램들과 연관된 파일들 및 설정들의 카탈로그를 작성하여 상기 파일들 및 설정들을 열거하는 단계;

상기 컴퓨팅 기반 장치에 등록된 상기 프로그램들과 연관된 파일들 및 설정들의 최종 로드 시간 및 날짜를 획득하는 단계;

상기 컴퓨팅 기반 장치에 등록된 상기 프로그램들과 연관된 파일들 및 설정들의 최종 수정 시간 및 날짜를 획득하는 단계; 및

상기 컴퓨팅 기반 장치에 등록된 상기 프로그램들과 연관된 파일들 및 설정들의 상기 최종 수정 시간 및 날짜를 상기 최종 로드 시간 및 날짜와 비교하는 단계를 포함하고,

상기 비교하는 단계는 최종 시도된 수정에 응답하지 않는 프로그램을 포함하는 보고를 발생시키는 상기 컴퓨팅 기반 장치에 등록된 상기 프로그램들의 불일치(inconsistencies)를 기록하는(noting) 단계를 더 포함하는,

컴퓨터 판독가능 저장 매체.

청구항 12

제11항에 있어서,

상기 비교하는 단계는 상기 컴퓨팅 기반 장치에 등록된 언인스톨된 프로그램들과 연관된 파일들 및 설정들을 비교하여 누설 파일들(leaked files)을 판정하는 단계를 더 포함하는, 컴퓨터 판독가능 저장 매체.

청구항 13

제12항에 있어서,

상기 누설 파일들은 상기 컴퓨팅 기반 장치로부터 제거되는, 컴퓨터 판독가능 저장 매체.

청구항 14

제11항에 있어서,

상기 비교하는 단계는 로그 파일에 쿼리하는 단계를 포함하는, 컴퓨터 판독가능 저장 매체.

청구항 15

삭제

청구항 16

삭제

청구항 17

삭제

청구항 18

삭제

청구항 19

삭제

청구항 20

삭제

청구항 21

삭제

청구항 22

삭제

명 세 서

기 술 분 야

[0001] 본 발명은 시스템 관리 및 분석에 관한 것이다.

배 경 기 술

[0002] 신뢰성 있고 안전한 컴퓨터 시스템을 구축하는 것에 대한 주요 과제는 시스템이 기능하는 방법을 결정하는 실행가능 파일들, 구성 설정들 및 기타 데이터 모두를 포함하는 시스템의 지속 상태(Persistent State; PS)를 관리하는 것이다. 오구성들 및 기타 PS 문제들은 개별 데스크톱 머신에서부터 대규모 인터넷 서비스에 이르는 다양한 시스템의 고장 및 보안 취약성의 주요 원인들이다. PS 문제들은, 하드웨어 컴포넌트들 및 프로그래밍 논리와 같은 시스템 요소들의 고장에 의해 발생하는 문제들과 함께, 전체 시스템에 악영향을 미칠 수 있다.

[0003] 시스템 PS를 비효율적으로 관리하는 손실은 크다. 예를 들어, PS 문제점들은 시스템 재부팅 또는 애플리케이션 재시동 후에 재발할 수 있다. 또한, 애플리케이션 관련 업데이트 및 패치(patch)와 같은 변화로 인해 PS 상태는 수행시간 동안 표류한다. 시스템에서 발생하는 변경사항에 대해 루프를 끊는 효과적인 방법이 현재는 존재하지 않는다. 이러한 시나리오에서, 공지된 문제 식별에 실패하고 후속의 시스템 재부팅/애플리케이션

재시동이 PS 문제의 치유에 실패하는 경우, 시스템을 수동으로 검사하여 PS의 근본 원인을 식별하는 것 외에는 다른 선택이 존재하지 않을 수 있다.

[0004] PS의 근본 원인을 식별하기 위한 시스템의 수동 검사는 다양한 잠재적인 문제점들로 인해 어렵고 비용이 많이 든다. 예를 들어, 문제점이 있는 애플리케이션에 악영향을 줄 수 있는 잠재적인 상태들의 세트는 거대하며, 따라서 잠재적 근본 원인 리스트는 시스템상에서 상태들의 전체 세트를 포함할 수 있다. 또한, 세트의 모든 가능한 조합에 대해 제대로 고려하지 않는 경우, 특히 단일의 PS의 근본 원인이 없는 경우에는, 상황은 잠재적으로 더욱 나빠질 수 있다.

발명의 상세한 설명

[0005] 본 요약은 모델 기반 라이센스 카운팅의 간략화된 개념들을 소개하기 위해 제공되는데, 이 모델 기반 라이센스 카운팅은 아래에서 추가적으로 설명된다. 본 요약은 청구 발명의 필수적인 특징들을 식별하고자하는 의도도 없고, 청구 발명의 범위를 결정하는데 사용하기 위한 의도도 없다.

[0006] 일 실시예에 있어서, 컴퓨팅-기반 장치의 프로그램은 분류되고 열거되며, 컴퓨팅-기반 장치에 등록된 프로그램의 최종 로드(load) 시간이 획득되며, 컴퓨팅-기반 장치에 등록된 프로그램에 연관된 파일의 최종 수정 시간과 최종 로드 시간의 비교가 이루어진다.

[0007] 본 발명을 구현하기 위해서, 시스템은 시스템 내에서 발생하는 수정과 연관된 데이터를 보고하는 하나 이상의 컴퓨터 프로그램 또는 에이전트를 포함한다. 이러한 데이터는 파일 및 설정들과의 모든 상호작용과 연관된 정보를 포함한다. 이러한 유형의 상호작용은 레지스트리 항목, 파일 등에 대한 읽기 액세스(access) 및 쓰기 액세스와 같은 동작은 물론, 로드와 같은 바이너리 모듈 상호작용들 등도 포함한다. 에이전트는 수집된 정보를 백엔드(backend) 서비스에 보고하는데, 이는 시스템 관리를 수행하기 위한 웹 리포트의 생성, 경고 또는 다른 서비스와의 통합과 같은 동작을 위해 보고된 정보를 처리한다. 또한, 이러한 처리는 데이터가 수집되는 단일의 머신 상에서도 수행될 수 있다. 이는 리포트의 생성, 경고 등을 포함한다. 특히, 시스템의 지속 상태(PS)는 시스템이 기능하는 방법을 결정하는 모든 실행가능 파일, 구성 설정들 및 기타 데이터를 포함한다. 지속 상태가 논의되지만, 논의된 기술 및 방법은 다른 종류의 상태들에도 적용될 수 있는 것으로 이해되어야 한다.

[0008] 보고된 데이터는 여러 목적으로 사용될 수 있다. 예를 들어, 데이터는 유발되고 있는 상호작용들이 설정된 정책에 일치하거나, 인증된 상호작용에 연관되는지를 검증하기 위해 검사될 수 있다.

[0009] 시스템 관리에 대해 기술된 시스템들 및 방법들의 양태들은 임의 수의 상이한 컴퓨터 시스템, 환경 및/또는 구성에서 구현될 수 있지만, 시스템 분석 및 관리의 실시예들은 이하의 예시적인 시스템 아키텍처(들)와 관련하여 설명된다.

실 시 예

[0022] 예시적인 시스템

[0023] 도 1은 하나 이상의 프로그램 사이의 상호작용과 연관된 정보가 수집되고 분석될 수 있는 예시적인 컴퓨터 시스템(100)을 도시한다. 시스템(100)은 하나 이상의 프로그램이 수행되거나 인스톨된 컴퓨팅-기반 장치(102), 컬렉션 서버(104), 아카이브(archive) 컬렉션(106) 및 리포트(들)(108)를 포함한다.

[0024] 하나 이상의 프로그램 및/또는 파일 시스템 또는 설정들 사이의 상호작용과 연관된 정보는 시스템(100)에서 일어날 수 있는 지속 상태(PS)에서의 수정을 표시한다. 컴퓨팅-기반 장치(102)는 임의 수의 컴퓨팅-기반 장치(102)를 포함할 수 있다. 예를 들어 일 실시예에 있어서, 시스템(100)은 컴퓨팅-기반 장치(102)로 동작하는, 수천 대의 사무실 개인용 컴퓨터(PC), 다양한 서버, 및 여러 국가에 배치된 기타 컴퓨팅-기반 장치를 포함하는 회사 네트워크도 포함할 수도 있다. 반면 다른 실시예에 있어서, 시스템(100)은 제한된 수의 PC가 있는 홈 네트워크를 포함할 수 있다. 컴퓨팅-기반 장치들(102)은 LAN, WAN, 또는 이 분야에 공지된 임의의 다른 네트워킹 기술을 포함하는 유무선 네트워크를 통해 다양한 조합으로 서로 접속될 수 있다.

[0025] 컴퓨팅-기반 장치(102)는 시스템(100)에서 하나 이상의 컴퓨팅-기반 장치(102) 및/또는 파일 시스템 및 설정들 사이의 상호작용과 연관된 정보를 캡쳐할 수 있는 함수를 실현할 수 있는 에이전트(110)를 포함할 수 있다. 일 실시예에 있어서, 에이전트(110)는 함수를 호출하는 스레드를 인터셉트하기 위해 함수에서 컴퓨터 판독가능 명령어를 삭제, 추가, 수정할 수 있는 스레드 데이터 레코더(TDR; Thread Data Recorder)일 수

있다. 다른 가능한 실시예에 있어서, 함수를 실현하는 것은, 스레드가 스레드와 연관된 데이터의 캡쳐를 가능하게 하는 함수의 컴퓨터 판독가능 명령어를 실행하도록 요구하기 위해, 함수의 컴퓨터 판독가능 명령어를 수정, 추가 및/또는 삭제하는 것을 포함한다. 그러나 다른 실시예에 있어서, 스레드와 연관된 데이터는 스레드가 연관된 프로그램과 관련된 정보, 스레드와 연관된 하나 이상의 상호작용, 및 스레드와 연관된 프로그램의 사용자와 관련된 정보를 포함한다. TDR이 논의되었지만, 인터셉션이 모든 알고리즘에 필수적으로 필요하지 않을 수 있으며, 따라서 논의된 기술 및 방법은 데이터 컬렉션에 기초한 TDR에 필수적으로 연관되지 않을 수 있다. 또한, 가상 머신(VM) 기반 실현은 코드가 동적으로 추가되는 TDR 기반 실현과 상이할 수 있다. VM에서 이러한 유형의 컬렉션을 수행하는 것은 VM의 내부의 하드 코딩된(hard-coded) 함수일 수 있다.

[0026] 실현된 함수들은 프로그램/프로세스에 의해 호출될 수 있는 임의의 함수들을 포함할 수 있다. 일 실시예에서, 실현된 함수들은 파일 시스템 드라이버, 레지스트리 함수, 새로운 프로세스 및/또는 서비스를 생성하는 함수 등과 같은 저 레벨 초크 포인트(choke point) 함수들을 포함할 수 있다.

[0027] 스레드 데이터 레코더에 의해 스레드들로부터 캡처된 데이터는 시스템(100)의 거동을 조절하고, 시스템(100)의 지속 상태 또는 조건을 검사하기 위해 저장 및/또는 처리될 수 있다. 스레드 데이터 레코더들에 의해 스레드들로부터 캡처될 수 있는 데이터의 유형들 및 스레드 데이터 레코더들의 동작은 발명의 명칭이 "Thread Interception and Analysis"이며 Verbowski 등에 의해서 2007년 10월 31일에 출원된 미국특허출원 11/993,749호에서 상세히 논의되어 있고, 이는 본 명세서에 참조로서 통합된다.

[0028] 컬렉션 서버(104)는 시스템(100)에서 발생할 수 있는 수정에 대한 정보를 수집하는 역할을 한다. 일 실시예에 있어서, 에이전트(110)는 컬렉션 서버(104)의 상호작용과 관련된 정보를 압축 로그로 저장한다. 그러나 다른 실시예에 있어서, 상호작용과 관련된 정보는 아카이브 컬렉션(106)에도 업로드될 수 있다. 컬렉션 서버(104) 또는 아카이브 컬렉션(106)에 수집된 정보의 분석은 리포트(들)(108)의 생성에 사용된다. 컬렉션 서버(104) 또는 아카이브 컬렉션(106)에서 수집된 정보에 대해 수행되는 분석의 결과로서 생성된 리포트(들)(108)는 하나 이상의 컴퓨터-기반 장치(102) 내에서 발생할 수 있는 수정 또는 상호작용에 관한 통찰력을 제공한다. 다른 실시예에 있어서, 리포터(들)(108)는 시각적 인터페이스를 통해 생성될 수 있다. 그러나 다른 실시예에 있어서, 시각적 인터페이스는 예전에 생성되거나 캐시된 리포트들을 디스플레이하거나 검색하는 브라우저를 통해 구현될 수 있다. 컬렉션 서버(104) 및 아카이브 컬렉션(106)은 컬렉션 서버(104) 또는 아카이브 컬렉션(106)로서 기능하는 단일의 장치에 존재하거나 그 일부분일 수 있다.

[0029] 상기 나타난 바와 같이, 컬렉션 서버(104) 또는 아카이브 서버(106)에 저장되고 에이전트(110)에 의해서 수집되는 정보는 시스템(100)의 기능에 대한 통찰력을 제공하기 위해 분석될 수 있다. 수행된 분석은 예외(anomaly) 검출, 변경 관리, 비정상적인 시스템 활동 관리, 보안 취약성 식별, 비인증 애플리케이션의 식별, 컴플라이언스 감사(compliance audit)의 수행 등을 포함할 수 있다.

[0030] 도 2는 에이전트(110)로부터의 데이터를 분석하고, 프로세스하고, 저장하도록 구성된 예시적인 컬렉션 서버(106)를 도시한다. 컬렉션 서버(106)는 하나 이상의 프로세서(들)(202) 및 메모리(204)를 포함한다. 프로세서(들)(202)는, 예를 들어 마이크로프로세서, 마이크로컴퓨터, 마이크로컨트롤러, 디지털 신호 프로세서, 중앙 프로세싱 유닛, 상태 머신, 논리 회로, 및/또는 구동적인 명령어에 기초하여 신호를 처리하는 임의의 장치를 포함한다. 특히, 프로세서(들)(202)는 메모리(204)에 저장된 컴퓨터-판독가능 명령어를 수행하고 패치하도록 구성된다.

[0031] 메모리(204)는, 예를 들어 휘발성 메모리(예를 들어, RAM) 및/또는 비휘발성 메모리(예를 들어, ROM, 플래시 등)를 포함하는 이 분야에 공지된 임의의 컴퓨터 판독가능 매체일 수 있다. 메모리(204)는 프로그램(들)(206) 및 데이터(208)를 또한 포함할 수 있다. 프로그램(들)(206)은, 하나 이상의 컴퓨팅-기반 장치(102) 및 파일 시스템 및/또는 설정들에서 수행되는 프로그램 사이의 상호작용과 연관된 데이터에 대해 특히 쿼리-관련 프로세스를 수행할 수 있다. 프로그램(들)(206)은, 예를 들어 쿼리 모듈(210), 통지 모듈(212), 운영체제(214) 및 기타 애플리케이션(들)(216)을 더 포함할 수 있다. 운영체제(214)는 프로그램(들)(206)에서 하나 이상의 모듈에서 기능하도록 하기 위한 구동 환경을 제공한다.

[0032] 쿼리 모듈(210)은 에이전트(110)에 의해 수집된, 로그 스토리지(218)에 포함된 정보와 같은 정보에 대해 쿼리-기반 동작들을 수행한다. 에이전트(110)에 의해 수집된 정보는 또한 아카이브 컬렉션(106)으로부터 검색가능할 수 있다. 쿼리(들)(220)는 사전정의된 쿼리들과 같은 복수의 쿼리를 포함한다. 이러한 사전정의된 쿼리는 보안 정책 정의와 같은 하나 이상의 정책 정의와 관련된 조건에 관련될 수 있는데, 이는 시스템(100)에 대하여 규정된다. 이러한 경우에 있어서, 쿼리 모듈(210)에 의해 수행될 수 있는 임의의 또는 모든 분석은

이렇게 사전정의된 정책 정의 또는 사전정의된 쿼리에 일치할 수 있다.

[0033] 쿼리 모듈(210)은 쿼리(들)를 하나 이상의 속성으로 제한할 수 있다. 이러한 속성들은, 파일명, 애플리케이션 유형, 실행 시간 등을 포함할 수 있다. 제한된 쿼리를 기초로 기능할 때, 쿼리 모듈(210)은 로그 스토리지(218) 및/또는 아카이브 컬렉션(106)에 저장된 모든 정보를 스캔닝한다. 예를 들어, 개체가 워드프로세서와 같은 소정의 애플리케이션에 관한 데이터에 대해 아카이브 컬렉션을 검색하기를 원하는 경우, 쿼리 모듈(210)은 워드프로세서에 의해 영향받고 초기화되는 상호작용과 연관된 항목 및 이벤트를 검색한다.

[0034] 쿼리(들)(220)는 시스템 관리자와 같은 하나 이상의 개체 또는 엔티티에 의해 입력되거나 프로그래밍된 쿼리들을 포함할 수 있다. 예를 들어, 쿼리(들)(220)는 소정의 사용자 ID와 연관된 모든 상호작용을 검출하기 위한 명령어를 포함할 수 있다. 또한, 쿼리(들)(220)는 하나 이상의 컴퓨팅-기반 장치(102)에서 실행되는 애플리케이션과 연관된 모든 상호작용을 검출하기 위한 명령어를 포함할 수 있다.

[0035] 컬렉션 서버(104)를 다시 참조하면, 파일 시스템 및/또는 설정들을 가진 하나 이상의 컴퓨팅-기반 장치(120)의 상호작용과 연관된 정보의 분석이 시스템(100)의 기능 및/또는 지속 상태를 결정하기 위해 수행된다. 에이전트(110)에 의해서 수집되어, 로그 스토리지(218) 및/또는 아카이브 컬렉션(106)에 저장되는 정보에 대한 분석을 수행하는데 쿼리 모듈(210)이 사용될 수 있다. 쿼리 모듈(210)은 쿼리(들)(220) 중에 특정된 하나 이상의 쿼리를 사용하여 로그 스토리지(218) 및/또는 아카이브 컬렉션(106)의 검색을 수행함으로써 이를 구현할 수 있다. 쿼리(들)(220)의 실행으로 생성되는 결과는 하나 이상의 컴퓨팅-기반 장치(102) 및 파일 시스템 및/또는 설정들 사이의 상호작용을 나타낸다.

[0036] 쿼리 모듈(210)은 통지 모듈(212)이 쿼리(들)(220)의 실행 결과로서 생성된 결과에 대한 통지를 발행하도록 명령한다. 통지 모듈(212)에 의해서 생성된 통지는 데이터(208)의 통지(들)(222)에 저장될 수 있다. 통지 모듈(212)에 의해 발행된 통지는 외부 저장소 장치와 같은 외부 데이터베이스에도 저장될 수 있다. 통지 모듈(212)은 쿼리(들)(220)의 수행 결과로 생성된 통지를 전달하도록 쿼리 모듈(210)에 의해서 명령될 수도 있다.

[0037] 쿼리 모듈(210)은 로그 스토리지(218) 및/또는 아카이브 컬렉션(106)을 검색하여 쿼리 모듈(210)에 의해서 실행되는 쿼리(들)(220)에 대한 하나 이상의 컴퓨팅-기반 장치(102) 사이의 상호작용과 연관된 정보의 편차를 검출할 수 있다. 이 경우, 특정 상호작용과 관련된 편차의 검출은 통지 모듈(212)에 의해 다시 통지될 수 있고, 대응하는 통지(들)(222)는 향후의 참조를 위해 저장될 수 있도록 시스템 관리자 또는 컴퓨팅 시스템과 같은 개체들에게 전달될 수 있다.

[0038] 통지 모듈(212)은 통지(들)(222)와 연관된 문맥 정보를 제공할 수도 있다. 문맥 정보는 추가적으로 설정을 특정할 수 있는데, 이는 대응하는 상호작용과 연관될 수 있다. 문맥 정보는 하나 이상의 스테이지에서의 관련된 통지(들)(222)에 대해 표시될 수 있다. 예를 들어, 하나의 레벨은 인스톨된 프로그램을 가질 수 있는 기계의 수와 관련된 통계적 정보, 가장 일반적인 버전의 파일 등을 제공한다. 다른 레벨의 표시는 예를 들어, 프로그램명, 버전 정보 등의 특성을 색인하는 데이터 컬렉션과 인스톨된 파일의 해시 값의 비교를 나타낸다. 그러나 다른 레벨의 표시는 공지된 문제점, 벤더 등과 관련된 임의의 부가 정보 또는 코멘트를 제공할 수 있다는 것을 나타낼 수 있다. 추가적인 레벨의 표시는 연관된 통지(들)(222)와 관련된 추가적인 속성을 특정하도록 구현될 수 있다. 통지(들)(222)는, 예를 들어 시스템 관리자와 같은 개체들이 통지(들)(222)를 검토하고, 필요한 경우에 적절한 동작을 취하도록 시각적 인터페이스를 통해 디스플레이될 수 있다.

[0039] 쿼리 모듈(210)은 하나 이상의 컴퓨팅-기반 장치(102) 및 파일 시스템 및/또는 설정들 상에서 수행되는 프로그램 사이의 상호작용 때문에 시스템(100)에서 발생하는 중요 변경의 검출에 사용될 수 있다. 중요 변경은 시스템의 PS에 대한 변경 또는 수정을 포함하는데, 이는 프로그램, 운영체제, 어카운팅(accounting)과 같은 특정 비즈니스 태스크를 완성하는데 사용되는 프로그램, 및 기타 프로그램의 예상하지 못한 실행으로 인한 결과일 수 있다. 시스템의 PS에 대한 중요 변경은 시스템이 보내져온 정보를 받아들이지 못하는 문제(choked), 보안 문제 등의 바람직하지 않은 상황을 방지하기 위해서 제어되고 인증된다. PS에 일어나는 모든 변경이 중요 변경은 아니라는 점에 또한 유의해야한다.

[0040] 중요 변경은 식별자에 의해서 표시될 수 있으며 표시가 지정될 때마다 분류될 수 있다. 중요 변경의 표시는 쿼리 모듈에 의해서 분류 규칙을 특정함으로써 수행될 수 있다. 분류 규칙에서 특정된 파라미터에 기초하여 적절한 파라미터가 중요 변경과 관련된 소정의 속성과 연관된다. 예를 들어, 쿼리 모듈(210)은 분류 규칙에 포함된 부분 문자열과 매치되는 것 각각을 각각의 중요 변경에 포함된 수정의 유형이나 이름에 연관시킨다. 중요 변경에 대한 분류가 우선값에 기초하여 할당될 수 있다. 예를 들어 이러한 경우에 있어서, 높은 우선순

위를 가지는 분류 부분 문자열에 대한 일치는 낮은 우선 순위를 가지는 분류 부분문자열에 우선한다. 높은 우선순위의 분류 부분문자열은 상대적 중요 변경에 대한 관련 분류로서 결정된다.

[0041] 중요 변경은 변경을 적어도 하나 이상의 다음 분류로서 라벨링(labeling)함으로써 분류할 수 있다.

[0042] 문제점: 인스턴트 PS의 제거 또는 존재로 인한 공지된 문제점 또는 결과를 나타냄.

[0043] 인스톨: 인스톨 또는 업그레이드의 결과로서의 PS의 변경을 나타냄.

[0044] 설정: 구성 PS 또는 구성 설정들에 대한 변경을 나타냄.

[0045] 콘텐츠: 웹 페이지, 이미지, 문자 또는 사용자 데이터를 나타냄.

[0046] 관리 변경: 시스템상에서 수행되는, 시스템을 관리하는 프로그램에 대한 인스톨, 패칭 또는 구성 변경을 나타냄.

[0047] 비인증: 금지된 값을 포함하는 구성 변경 또는 비인증되거나 금지된 애플리케이션의 인스톨을 나타냄.

[0048] 사용자 동작: 사용자가 윈도우 애플리케이션을 실행시키거나 윈도우 애플리케이션에 로그인한 결과로서의 PS에서의 변화를 나타냄.

[0049] 노이즈: 임시적이거나 캐시된 PS를 나타냄.

[0050] 알려지지 않음: 분류되지 않은 PS를 나타냄.

[0051] 추가적인 표시가 중요 변경을 다시 분류하고 다른 변경으로부터 식별되도록 하기 위해 제공될 수 있다.

[0052] 쿼리 모듈(210)은 시스템(100)에서 수행되는 인증되거나 인증되지 않은 프로그램의 상태를 결정하는데도 사용될 수 있다. 이는 인증된 프로세스나 프로그램은 시스템(100)상에서 수행되어야 한다는 사실에만 기초한다. 쿼리 모듈(210)은 쿼리(들)(220)에서 특정된 속성을 비교함으로써 시스템(100)에서 수행되는 인증되거나 인증되지 않은 프로그램의 상태를 결정하며, 시스템의 PS에서의 특정 변경 또는 수정을 정의하는 속성을 결정한다. 예를 들어, 쿼리 모듈(210)은 인증되지 않은 프로그램으로서의 애플리케이션 유형을 특정하는 쿼리(들)(220)를 실행한다. 쿼리(들)(220)의 실행의 결과로 얻은 결과는 특정 유형의 애플리케이션의 실행에 대한 응답으로 발생하는 PS에서의 변경에 대한 정보를 포함한다. 결과들을 얻으면서 쿼리 모듈(210)은 인증되지 않은 프로그램의 동작 또는 실행에 의해서 나타나는 변경으로서 이러한 결과를 표시한다.

[0053] 쿼리 모듈(210)은 인가되거나 인가되지 않은 프로그램의 사전정의된 리스트에서 특정된 속성들을 시스템의 PS에서의 특정 변경 또는 수정을 정의하는 속성과 비교할 수 있다. 인스턴트 케이스의 리스트는 특정 수의 인가되거나 인가되지 않은 프로그램을 포함할 수 있다. 사전정의된 리스트에서 인가되지 않았다고 식별된 프로그램과 유사한, 시스템(100)에서 수행되는 프로그램은 비인증 프로그램으로서 표시된다.

[0054] 사전정의된 리스트에서 특정된 인가되거나 인가되지 않은 프로그램은 프로그램의 다양한 특징 및/또는 속성을 나타내는 라벨과 같은 추가적인 정보를 또한 포함할 수 있다. 이러한 추가적인 정보의 예는 "인가됨", "유형", "카테고리", "함수", "생산품 정보", "제조자 정보" 및 "생산품 설명"과 같은 라벨을 포함한다. 예를 들어, "인가됨"이라고 라벨링된 프로그램은 시스템(100)의 하나 이상의 컴퓨터-기반 장치(102)에서 실행되는 인증된 프로그램으로 간주되며, "카테고리" 라벨은 프로그램의 의도적인 사용을 특정한다.

[0055] 처음으로 하나의 프로그램에 의해 수행된 변경 또는 수정은 디폴트로 인가되지 않고 "비인증"으로서 표시된다. 예를 들어, 프로그램에 의해서 처음으로 행해진 변경 또는 수정을 검출하면, 쿼리 모듈(210)은 프로그램 및 그 연관된 상호작용을 "비인증"으로서 표시한다. "비인증"으로 표시된 이러한 프로그램은, 예를 들어 진단 루틴을 수행할 필요가 있는지 또는 계류된 인가를 인가할지를 시스템 관리자가 검토할 수 있도록, 통지 모듈(212)에 의해 통지될 수 있다. 인가된 경우, 인가된 프로그램은 프로그램의 특성을 나타내는 적절한 라벨과 연관되며, 인가되거나 인가되지 않은 프로그램을 포함하는 사전정의된 리스트에 추가될 수도 있다.

[0056] 쿼리 모듈(210)은 확장성 포인트(EP)들을 검출할 수도 있다. EP들은 하나 이상의 컴퓨팅-기반 장치(102)에서 실행되는 운영체제 또는 프로그램과 연관된 명령어의 실행 및 동적 로딩을 나타내는 상호작용이다. 예들 들어, 하나 이상의 컴퓨팅-기반 장치(102)에서 실행되는 워드 프로세서, 스프레드시트 애플리케이션 등과 같은 주 프로그램이 시작되는 경우, 주 프로그램은 또한 주 프로그램의 실행에 추가 기능을 제공하는, 애드온 프로그램과 같은 기타 프로그램과 연관된 명령어를 트리거할 수 있다. 이러한 방법으로, 주 프로그램의 실행은, 주 프로그램과 파일 시스템 사이의 상호작용 및 주 프로그램 및 파일 시스템의 실행에 추가 기능을 제공하는 기타 프로그램들 사이의 상호작용을 포함하는 다양한 명령어를 생성할 수 있다. 이러한 정보는 주 프로그램

이 인스톨된 시스템의 기능에 대한 통찰력 및 이러한 인스톨이 시스템에 생성할 수 있는 충돌을 예상하는 통찰력을 제공할 수 있다.

[0057] 주 프로그램의 실행의 결과로 생성된 다양한 상호작용과 연관된 정보는, 예를 들어 에이전트(110)에 의해 인터셉트되고 복사될 수 있다. 다양한 상호작용과 연관된 이벤트 정보는 로그 스토리지(218) 및/또는 아카이브 컬렉션(106)에 압축된 로그로서 저장될 수 있다. 비록 이벤트 정보가 압축 저장소에 저장되지만, 압축 저장소가 필수적으로 사용될 필요는 없다. 그러나, 압축 저장소의 사용은 저장소가 보다 적은 공간을 차지하게 함으로써, 시스템을 보다 조정가능하게 한다. 주 프로그램 및 파일 시스템의 다른 프로그램과 연관된 상호작용을 검출하기 위해서, 저장된 이벤트 정보가 시스템 관리자와 같은 엔티티에 의해서 또는 쿼리 모듈(210)에 의해서 검토될 수 있다. 이러한 방법으로, 주 프로그램의 실행과 연관된 다른 프로그램이 검출될 수 있다.

[0058] 쿼리 모듈(210)은 또한 주 프로그램에 대한 직접 EP들을 검출하는데 사용될 수 있다. 예를 들어, 쿼리 모듈(210)은 상호작용들을 격리함으로써 직접 EP들을 검출할 수 있는데, 이러한 상호작용은 (1) 주 프로그램의 실행 전에 실행을 위해 시스템 메모리 내에 로딩된 다양한 프로그램들에 관련되고, (2) 주 프로그램을 참조하거나 주 프로그램의 실행과 연관된다.

[0059] 예시적인 일 구현에서, 쿼리 모듈(210)은 주 프로그램의 실행 전에 실행을 위해 시스템 메모리 내에 로딩된 다양한 프로그램에 관련된 상호작용들에 대해 로그 스토리지(218) 및/또는 아카이브 컬렉션(106)에 쿼리함으로써 주 프로그램에 대한 잠재적인 직접 EP들을 식별할 수 있다. 예를 들어, 쿼리 모듈(210)은 주 프로그램의 실행 전에, 1초와 같은 소정 시간 범위 내에 실행을 위해 시스템 메모리 내에 로딩된 다양한 프로그램에 관련된 상호작용들에 대해 쿼리할 수 있다. 쿼리 모듈(210)은 주 프로그램을 참조하거나 주 프로그램의 실행과 연관되는 상호작용들에 대해 잠재적 EP들에 쿼리함으로써 잠재적 EP들로부터 주 프로그램과 연관된 직접 EP들을 식별할 수 있다. 직접 EP들은 기타 데이터(들)(224)에 저장될 수 있다.

[0060] 쿼리 모듈(210)은 또한 간접 EP들을 검출하는데 사용될 수 있다. 예를 들어, 위의 예로 돌아가면, 쿼리 모듈(210)은 직접 EP들과 연관되거나 이를 참조하는 상호작용에 대해 로그 스토리지(218) 및/또는 아카이브 컬렉션(106)에 쿼리할 수 있다. 이러한 상호작용은 간접 EP들이라 지칭한다. 간접 EP들은 다른 데이터(들)(224)에 저장될 수 있다.

[0061] 쿼리 모듈(210)은 직접 EP들을 모니터링함으로써 악성 소프트웨어 애플리케이션들의 존재를 검출하는 데 사용될 수도 있다. 악성 소프트웨어 애플리케이션들은 정상적인 환경에서는 프로그램과 연관되지 않는 "스파이웨어", "트로이 목마", "웜", "바이러스" 등과 같은 애플리케이션들을 포함할 수 있다. 예를 들어, 쿼리 모듈(210)은 하나 이상의 컴퓨팅-기반 장치(102) 상에서 실행되는 프로그램에 대한 EP들을, 프로그램이 악성 소프트웨어의 부재하에 컴퓨팅-기반 장치(102) 상에서 실행된 때 발견된 동일 프로그램에 대한 제어 EP들과 비교할 수 있다. EP들과 제어 EP들 간의 차이들은 이 차이들이 프로그램과 함께 실행되는 악성 소프트웨어의 존재를 나타내는지를 결정하기 위해 쿼리 모듈(210) 및 시스템 관리자와 같은 엔티티들에 의해 검사될 수 있다. EP들을 이용하여 발견된 악성 소프트웨어는 쿼리 모듈(210), 시스템 관리자 등에 의해, 영향 받은 컴퓨팅-기반 장치(102)로부터 제거될 수 있다.

[0062] 다른 실시예에 있어서, 쿼리 모듈(210)은 EP의 검출에 대한 응답으로 통지 모듈(212)이 통지(들)(222)를 생성하는데 사용될 수 있다. 그러나 다른 실시예에 있어서, 다시 분석하거나 필요한 진단 루틴을 수행하기 위해서 시스템 관리자와 같은 개체가 통지(들)(222)를 검토하는데 사용되는 시각적 인터페이스를 통해 생성된 통지(들)(222)가 다시 보여지거나 탐색될 수 있다.

[0063] 도 3은 가능한 일 구현에 있어서, 생성된 통지(들)(222)을 도시하는 예시적인 시각적 인터페이스(300)를 도시한다. 도시된 실시예에 있어서, 시각적 인터페이스(300)는 주 프로그램(예를 들어, 웹 브라우저)에 의해서 수행되는 다운로드를 그 수행상에서 도시한다. 시각적 인터페이스(300)는 그 수행동안에 주 프로그램에 의해 다운로드된 프로그램의 리스트를 세그먼트 302 및 304로 표시한다(도 3에 도시된 예에 있어서, 특히 "MSN 서치 툴바" 및 "윈앰프 미디어 플레이어"). 도시에 있어, 세그먼트 302 및 304로 표시된 프로그램의 다운로드는 또한 세그먼트 302 및 304로 표시된 프로그램 및 주 프로그램 외의 다른 프로그램에 대응하는 프로그램 파일의 생성으로 귀결한다는 점이 세그멘트 306으로 표시된다. 따라서 시각적 인터페이스(300) 형식의 시각적 표현은 주 프로그램을 인스톨하거나 실행하거나 다운로드하는 동안 시스템(100)의 하나 이상의 컴퓨팅-기반 장치(102)상에서 우연하게 인스톨되는 프로그램의 리스트를 제공한다.

[0064] 세그멘트 306은 또한 주 프로그램외의 프로그램의 실행 또는 인스톨의 결과로서 시스템(100)의 PS에서의 수정 또는 충돌을 나타낼 수도 있다. 또한 주 프로그램 외의 프로그램의 실행은 주 프로그램의 실행에 의존할 수

있다. 예를 들어, 도시된 바와 같이, "MSN 서치 툴바"는 주 프로그램의 프로그램 파일의 실행 상에서 활성화될 수 있다. 이러한 효과에 대한 결정은 주 프로그램에 대응하는 EP들을 검출함으로써 구현될 수 있다. 주 프로그램과 연관된 EP들을 모니터링함으로써, 주 프로그램의 실행에 의존하는 다른 프로그램의 프로그램 실행의 인스턴스들이 검출되며, 필요한 경우에 정정하는 동작이 취해질 수 있다.

[0065] 도 4에 도시된 바와 같이, 주 프로그램에 의존하는 기타 프로그램의 실행의 인스턴스들은 다른 시각적 인터페이스(400)를 통해 디스플레이될 수 있다. 도 4는 세그멘트 402로 표시된 주프로그램, 예를 들어 "iexplorer.exe"의 수행이 세그멘트 404로 표시된 윈앰프의 수행을 일으키며, 이는 다시 세그멘트 406으로 표시된 "emusic.exe"을 또한 실행한다. 이러한 도시로부터, 주 프로그램과 연관된 EP의 검출은 시각적인 수단을 통해서 자세하게 구현될 수 있다.

[0066] 누설된 PS를 검출하는데 쿼리 모듈(210)이 사용될 수 있다. 누설 파일은 파일 또는 레지스트리 설정들을 생성했던 프로그램이 언인스톨된 후에, 시스템(100)과 같은 시스템상에 남겨진 파일 또는 레지스트리 설정들을 포함한다. 이는 인스톨의 결과로서 생성되었으나, 인스톨 과정이 완성된 후에 삭제에 실패한 파일 또는 설정들, 예를 들어 임시 파일을 포함할 수 있다. 또한, 프로그램의 실행시간 동안(즉, 인스톨 후에) 생성된 PS와 같은 다른 유형의 PS가 누설될 수 있다. 이러한 예는 초기 인스톨 후에 별도로 인스톨된 프로그램에 대한 확장 또는 최초 사용시에 생성된 상태일 수 있다.

[0067] 누설 파일들을 검출하기 위해서, 쿼리 모듈(210)은 시스템(100)상에 로드된 각각의 프로그램과 연관된 인스톨 파일 및 설정 변화를 분류하는데, 이는 최초 인스톨 및 프로그램의 사용을 통해 추적될 수 있다. 나중에, 프로그램이 언인스톨된 경우, 그 프로그램에 대한 레지스트리 설정 또는 인스톨 파일 및 구성의 대응하는 카탈로그가 회수될 수 있으며, 모든 인스톨 파일들 및 레지스트리 설정들이 제거되고 리셋되었다는 것을 확인하기 위해 시스템(100)이 검사될 수 있다. 컴퓨팅-기반 장치(102) 상에서의 누설 파일들을 검출하기 위해서, 쿼리 모듈(210)은 하나 이상의 컴퓨팅-기반 장치(102)를 통해서 스캔을 실행함으로서 인스톨 파일들을 분류하여, 컴퓨팅-기반 장치(102)상에 인스톨된 애플리케이션과 같은 모든 프로그램을 검출한다.

[0068] 쿼리 모듈(210)은 하나 이상의 컴퓨팅-기반 장치(102)의 운영체제의 인스톨러 데이터베이스에 등록된 모든 프로그램의 리스트를 획득할 수도 있다. 인스톨러 데이터베이스의 예는 해당 컴퓨팅-기반 장치에 인스톨된 프로그램의 등록 리스트를 생성하는 컴포넌트를 포함한다.

[0069] 쿼리 모듈(210)은 레지스트리 구성 또는 설정 정보에 대해, 그리고 컴퓨팅-기반 장치(102)의 운영 체제에 등록된 프로그램의 리스트를 열거하기 위해서 로그 스토리지(218) 및/또는 아카이브 컬렉션(106)에 쿼리한다. 쿼리 모듈(210)은 그 후에 컴퓨팅-기반 장치(102)상에 인스톨된 모든 프로그램의 레지스트리 항목 및 파일을 열거하기 위해서 로그 스토리지(218) 및/또는 아카이브 컬렉션(106)을 스캔할 수 있는데, 이는 모든 PS에 대해 일반화될 수 있다. 레지스트리 항목 및 파일을 열거하기 위해서, 쿼리 모듈(210)은 컴퓨팅-기반 장치(102)상에 인스톨된 프로그램의 하나 이상의 속성, 예를 들어 프로그램 ID에 대응하는 레지스트리 항목 및 모든 파일에 대해 쿼리할 수 있다.

[0070] 컴퓨팅-기반 장치(102)상의 설정 또는 파일이 컴퓨팅-기반 장치(102)상에 인스톨된 프로그램의 프로그램 ID에 대응하는 레지스트리 항목 및 파일들에 포함되지 않는 경우, 쿼리 모듈(210)은 파일 및 설정이 누설 파일이라는 것을 추론할 수 있다. 누설 파일은 쿼리 모듈(210)이나 운영 체제, 시스템 관리자 등을 포함하여 기타 다양한 프로그램에 의해서 제거될 수 있다.

[0071] 예를 들어, 시스템 관리자와 같은 개체가 누설 파일을 검토하고 필요한 경우에 적절한 행동을 취할 수 있도록 하는 시각적 인터페이스를 통해서 검출된 누설 파일들(PS)이 디스플레이될 수 있다. 또한, 디스플레이된 누설 파일들 (PS) 및 연관된 정보는 향후에 참조할 수 있도록, 외부 저장소 컬렉션, 예를 들어 외부 데이터베이스에 저장될 수 있다. 주 애플리케이션이 제거된 경우, 시스템에 의해 누설 상태를 자동적으로 제거하기 위해 누설 PS 리스트가 사용될 수 있다. 이러한 누설 PS 리스트는 시스템상의 각각의 PS를 소유자의 애플리케이션과 연관시키기 위해서 사용될 수도 있다.

[0072] 쿼리 모듈(210)은 일반적인 오구성, 구식 소프트웨어 버전 등을 포함하여 무효 모듈, 변경된 파일이나 설정들로 인한 무효 프로세스를 검출할 수 있다. 예를 들어, 온-디스크(on-disk) 실행가능 파일, 프로그램 파일 또는 설정들을 교체한 후, 그 영향을 받는 프로세스들을 재시작하는데 시스템 업그레이드가 실패한 경우에 무효 프로세스가 발생한다. 결과적으로, 무효 프로세스가 발견된 컴퓨팅-기반 장치는 업그레이드를 무시하고, 구식 수행가능 파일, 프로그램 파일 또는 설정들에 기초하여 실행을 계속할 것이다.

[0073] 무효 프로세스를 검출하기 위해서, 쿼리 모듈(210)은 로그 스토리지(218) 및/또는 아카이브 컬렉션(106)에 저장된 프로그램의 상호작용과 연관된 정보에 대해 쿼리할 수 있다. 쿼리 모듈(210)은 하나 이상의 컴퓨팅-기반 장치(102)상에 인스톨된 프로그램의 최근-로드 시간에 대해 로그 스토리지(218) 및/또는 아카이브 컬렉션(106)에 쿼리한다. 쿼리 모듈(210)은 인스톨된 소프트웨어와 연관된 파일 또는 레지스트리 설정들의 최근-로드 시간에 대해 로그 스토리지(218) 및/또는 아카이브 컬렉션(106)에 쿼리할 수도 있다. 예시적인 일 실시예에 있어서, 쿼리 모듈(210)은 소프트웨어와 함께 인스톨된 동적 링크 라이브러리(Dynamic Link Libraries: DLLs)와 연관된 소프트웨어와 연관된 레지스트리 설정 및 파일의 최근-로드 시간에 대해 로그 스토리지(218) 및/또는 아카이브 컬렉션(106)에 쿼리한다. 쿼리 모듈(210)은 컴퓨팅-기반 장치상에 인스톨된 소프트웨어의 최근 수정 날짜 또는 시간에 대해 로그 스토리지(218) 및/또는 아카이브 컬렉션(106)에 쿼리할 수도 있다. 이러한 수정은, 예를 들어 인스톨된 소프트웨어의 최근 공개된 버전과 연관된 하나 이상의 프로그램 설정 또는 파일에 대한 액세스를 포함한다.

[0074] 소프트웨어의 최근 공개된 수정의 날짜 또는 시간보다 소프트웨어의 최근 로드 시간이 더 최근 경우에 있어서, 최근 로드된 업데이트를 사용하지 않는 프로그램으로 인한 불일치가 발생할 수 있다. 쿼리 모듈(210)에 의해서 검출된 이러한 불일치는 시스템 관리자와 같은 개체에 의해서 인식되어 정정될 수 있다.

[0075] 개체, 예를 들어 시스템 관리자가 무효 파일들을 검토하고 필요한 경우에 적절한 동작을 취할 수 있도록 시각적 인터페이스를 통해 검출된 무효 파일들이 디스플레이될 수 있다. 디스플레이된 무효 파일들 및 연관된 정보는 향후의 참조를 위해 외부 저장소 컬렉션, 예를 들어 외부 데이터베이스에 저장될 수 있다.

[0076] 쿼리 모듈(210)은 "말웨어(malware)", "스파이웨어", "트로이 목마", "바이러스" 등과 같은 애플리케이션들을 포함하는 공개된 부적절한(unwarranted) 프로그램의 발생을 검출할 수 있다. 이와 더불어, 쿼리 모듈(210)은 실행을 위해 하나 이상의 컴퓨팅-기반 장치(102)상의 메모리에서 로드된 프로그램에 대해 로그 스토리지(218) 및/또는 아카이브 컬렉션(106)에 쿼리하고 검색할 수 있다. 실행을 위해 메모리에 로드된 프로그램은, 예를 들어 쿼리 모듈(210)에 의해서 공개된 부적절한 프로그램의 리스트와 비교될 수 있다.

[0077] 예를 들어, 쿼리 모듈(210)은 프로그램과 연관된 프로그램 ID와 같은 식별자에 기초하여, 실행을 위해 컴퓨팅-기반 장치(102)상의 메모리에 로드된 프로그램의 발생을 검출할 수 있다. 쿼리 모듈(210)은 실행을 위해 컴퓨팅-기반 장치(102)상의 메모리에 로드된 프로그램의 식별자를 공개된 부적절한 프로그램의 식별자, 예를 들어 프로그램 ID의 리스트와 비교한다. 실행을 위해 메모리에 로드된 프로그램의 식별자가 공개된 부적절한 프로그램의 식별자와 일치하는 경우, 쿼리 모듈(210)은 실행을 위해 컴퓨팅-기반 장치(102) 상의 메모리에 로드된 프로그램을 제거할 수 있다. 가능한 일 구현에 있어서, 공개된 부적절한 프로그램의 식별자의 리스트는 적어도 부분적으로 시스템 관리자에 의해서 입력될 수 있다.

[0078] 쿼리 모듈(210)에 의해 검출된 부적절한 프로그램은 시스템 관리자와 같은 개체가 부적절한 프로그램을 검토하고 부적절한 프로그램의 제거를 위한 적절한 동작을 취하도록 시각적 인터페이스를 통해 디스플레이될 수 있다. 동일하거나 유사한 부적절한 프로그램을 검출하기 위해 향후에 참조할 수 있도록, 디스플레이된 부적절한 프로그램 및 연관 정보는 외부 저장소 컬렉션, 예를 들어 외부 데이터베이스에 저장될 수 있다.

[0079] 프로그램과 연관된 식별자를 가지지 않는 하나 이상의 컴퓨팅 장치(102)상의 비식별 프로그램은 쿼리 모듈(210)에 의해서 검출될 수 있고, 비식별 프로그램이 부적절한 프로그램인지 아닌지를 확인하기 위해서 시스템 관리자에게 보고될 수 있다. 시스템 관리자는 보고서의 형태로 된 비식별 프로그램의 리스트를 검토함으로써, 비식별 프로그램의 속성을 검사할 수 있다. 시스템 관리자에 의한 검토는 비식별 프로그램의 목적, 비식별 프로그램의 다른 프로그램에 대한 의존성을 검사하고, 비식별 프로그램의 부적절한지를 결정하는 것을 포함할 수 있다. 또한, 시스템 관리자는 비식별 프로그램이 부적절한지를 결정하기 위해서, 비식별 프로그램의 특징과 유사한 특징을 가지는 프로그램들의 과거 경험을 검토할 수 있다.

[0080] 시스템 관리자가 비식별 프로그램이 부적절하다고 결정하는 경우, 시스템 관리자는 컴퓨팅-기반 장치(120)로부터 비식별 프로그램을 제거할 수 있다. 예를 들어, 시스템 관리자는 비식별 프로그램 그 자체를 제거할 수 있거나, 컴퓨팅-기반 장치(102)의 요소가 비식별 프로그램을 제거하도록 명령할 수 있다.

[0081] 또한, 시스템 관리자는 생성된 보고서나 하나 이상의 통지(들)(222)를 기초로, 프로그램 ID와 같은 식별자를 비식별 프로그램에 할당하고, 부적절한 프로그램의 리스트상에 식별자를 포함할 수 있다. 이러한 방법으로, 컴퓨팅-기반 장치(102) 상에 비식별 프로그램이 다시 나타나는 경우, 비식별 프로그램은 연관된 식별자에 기초하여 부적절한 프로그램으로서 신속하게 식별될 수 있다. 또한, 비식별 프로그램의 제거는 에이전트(110)

등의 컴퓨팅-기반 장치(102)의 요소에 의해서 구현될 수 있다.

[0082] 쿼리 모듈(210)에 의해서 검출된 비식별 프로그램 및 이와 연관된 프로세스는 시스템 관리자와 같은 개체가 비식별 프로그램을 검토하고 이를 제거하기 위한 적절한 동작을 취하도록 시각적 인터페이스를 통해 디스플레이될 수 있다. 향후에 동일하거나 유사한 부적절한 프로그램을 검출하기 위한 참조로 이용하기 위해서, 디스플레이된 부적절한 프로그램 및 연관 정보는 외부 저장소 컬렉션, 예를 들어 외부 데이터베이스에 저장될 수 있다. 또한, 원치않는 변경도 식별되거나 추적될 수 있다.

[0083] 쿼리 모듈(210)은, 하나 이상의 컴퓨팅-기반 장치(102)상에서 실행되는 프로그램이 이용가능한 기억장소 또는 네트워크 드라이브에 기록하는 것을 거부함으로써, 이러한 장소에 파일들을 복사하는 것을 차단할 수 있다. 쿼리 모듈(210)은 또한 향후의 이러한 기록을 방지하기 위해, 감사(auditing) 의도로 수행된 이전의 거기록 거부를 검토할 수도 있다.

[0084] 예시적인 방법

[0085] 스레드 인터셉션 및 분석을 위한 예시적인 방법들이 도 1 내지 4를 참조하여 설명된다. 이러한 예시적인 방법들은 일반적으로 컴퓨터 실행가능 명령어와 관련하여 기술될 수 있다. 일반적으로, 컴퓨터 실행가능 명령어는 특정 기능들을 수행하거나 특정 추상 데이터 유형을 구현하는 루틴, 프로그램, 개체, 컴포넌트, 데이터 구조, 프로시저, 모듈, 함수 등을 포함할 수 있다. 방법들은 통신 네트워크를 통해 연결되어 있는 원격 처리 장치들에 의해 기능들이 수행되는 분산 컴퓨팅 환경에서 실시될 수도 있다. 분산 컴퓨팅 환경에서, 컴퓨터 실행가능 명령어는 메모리 저장 장치들을 포함하여 로컬 및 원격 컴퓨팅 저장 매체 모두에 위치될 수 있다.

[0086] 도 5는 하나 이상의 컴퓨팅-기반 장치(102) 및/또는 파일 시스템 및 설정들 상에서 실행되는 프로그램 사이의 상호작용과 연관된 정보를 캡쳐하고 수집하기 위한 예시적인 방법(500)을 나타낸다. 방법이 설명되는 순서는 제한적으로 해석되어서는 아니되며, 임의 수의 설명되는 방법 블록들이 그 방법 또는 대안 방법을 구현하기 위해 임의 순서로 조합될 수 있다. 또한, 개별 블록들은 여기에 설명되는 발명의 사상 및 범위를 벗어나지 않고 방법으로부터 삭제될 수 있다. 또한, 방법은 임의의 적절한 하드웨어, 소프트웨어, 펌웨어, 또는 이들의 조합으로 구현될 수 있다.

[0087] 블록 502에서, 시스템 상에서 실행되거나 수행되는 프로그램과 연관된 정보나 데이터가 인터셉트된다. 일 구현에서, 프로그램/프로세스가 수정된 함수 코드를 포함하는 실현된 함수를 호출할 때 이러한 정보가 수집된다. 예를 들어, 컴퓨터 판독가능 명령어는 하나 이상의 함수에게 하나 이상의 컴퓨팅-기반 장치(120) 및/또는 파일 시스템 및 설정들 상에서 실행되는 프로그램 간의 상호작용과 연관된 데이터를 캡처할 것을 지시하도록 수정될 수 있다. 일 구현에서, 가상 머신은 수행되는 원 코드(original code)를 해석할 때, 데이터 컬렉션 논리를 직접 적용한다. 이러한 기법은 원 코드의 수정을 필요로 하지 않는다. 유사하게 이는 프로세서 하드웨어에 직접 구현될 수 있다.

[0088] 에이전트(110)와 같은 에이전트는 시스템(100)에서 하나 이상의 함수를 실현할 수 있다. 하나 이상의 함수는 하나 이상의 함수와 연관된 컴퓨터 판독가능 명령어를 수정함으로써 실현될 수 있다.

[0089] 스레드 데이터 레코더와 같은 에이전트(110)는 시스템(100) 내의 수정된 함수들을 호출하는 스레드들을 인터셉트할 수 있다. 스레드들과 연관된 프로그램들은 프로그램 계층, 미들웨어 계층, 운영 체제 계층 등과 같은 여러 동작 계층 중 하나에서 실행될 수 있다. 프로그램이 상호작용하려고 시도할 수 있는 파일 시스템은 파일들(데이터 파일, 실행가능 파일 등) 및 설정 정보(구성 설정 또는 레지스트리 설정 등) 등을 포함할 수 있다.

[0090] 블록 504에서, 하나 이상의 컴퓨팅-기반 장치(102) 상에서 실행되는 프로그램의 수행과 연관된 데이터 또는 다양한 정보가 메모리 장소에 복사되거나 수집된다. 수정된 함수에 의해서 유발된 상호작용을 포함하는, 파일 시스템 및/또는 설정들과 프로그램의 상호작용과 연관된 정보는 메모리 장소에 복사되고 전송된다. 예를 들어, 에이전트(110)는 상호작용과 연관된 데이터를 선택적으로 또는 모두를 복사하여, 컬렉션 서버(104)와 같은 메모리 장소에 데이터를 저장할 수 있다. 상호작용과 연관된 데이터는 실현된 함수에 의해서 유발된 상호작용으로 간주되는 정보를 포함할 수 있다.

[0091] 블록 506에서, 메모리 장소에 저장된 데이터가 압축된다. 일 구현에서, 압축된 데이터는 다른 메모리 장소에 저장될 수 있다. 예를 들어, 압축된 데이터는 컬렉션 서버(104)의 로그 스토리지(218) 및/또는 아카이브 컬렉션(106)에 저장될 수 있다.

[0092] 블록 508에서, 압축된 데이터는 분석을 위해 주기적으로 업로드된다. 압축된 데이터는 컬렉션 서버(106)에

업로드되거나 컬렉션 서버(106)로서 기능하는 메모리 장소에 업로드될 수 있다. 분석을 위한 압축된 데이터의 업로딩의 주기는 변할 수 있다. 일 구현에서, 압축된 데이터는 지정된 시간 간격 후에 업로딩된다. 다른 구현에서, 압축된 데이터는 압축된 데이터가 메모리의 사전정의된 임계치 제한을 초과할 때 업로딩될 수 있다.

[0093] 도 6은 중요 변경들을 분류하기 위한 예시적인 방법(600)을 나타낸다. 중요 변경은 어카운팅과 같은 특정 비즈니스 태스크를 수행하는데 사용되는 프로그램, 운영 체제, 기타 프로그램(들) 등의 예기 하지 못한 수행으로 인해 발생할 수 있는 변화 또는 수정을 포함한다. 방법이 설명되는 순서는 제한적으로 해석되어서는 아니되며, 임의 수의 설명되는 방법 블록들이 그 방법 또는 대안 방법을 구현하기 위해 임의 순서로 조합될 수 있다. 또한, 개별 블록들은 여기에 설명되는 발명의 사상 및 범위를 벗어나지 않고 방법으로부터 삭제될 수 있다. 또한, 방법은 임의의 적절한 하드웨어, 소프트웨어, 펌웨어, 또는 이들의 조합으로 구현될 수 있다.

[0094] 블록 602에서, 다양한 파라미터 값들에 의해 나타나는 분류 규칙이 특정된다. 예를 들어, 분류 규칙은 분류 규칙을 정의하는 파라미터의 값에 따라 쿼리 모듈(210)에 의해서 특정될 수 있다.

[0095] 블록 604에서, 분류 규칙을 정의하는 파라미터들은 중요 변경의 특성 및 특징을 정의하는 하나 이상의 속성과 연관된다. 쿼리 모듈(210)은 분류 규칙을 정의하는 파라미터들과 중요 변경을 특정하는 속성을 연관시킨다. 하나 이상의 파라미터 값들과 고려할 중요 변경을 특정하는 속성을 연관시켜 가능한 분류의 집합을 만든다. 예를 들어, 쿼리 모듈(210)은 분류 규칙에 포함된 부분 문자열과 매치되는 것 각각을 각각의 중요 변경에 포함된 PS의 이름에 연관시킨다.

[0096] 블록 606에서, 하나 이상의 가능한 분류가 우선 값으로 할당된다. 쿼리 모듈(210)은 우선 값을 하나 이상의 가능한 분류에 할당할 수 있다. 예를 들어, 긴 시간동안 일어난 특정 중요 변경은 보다 높은 우선 값에 할당될 수 있다.

[0097] 블록 608에서, 가장 높은 우선순위 값을 가지는 가능한 분류가 해당 중요 변경에 할당된다. 일 구현에 있어서, 쿼리 모듈(210)은 가능한 분류에 할당될 가장 높은 우선순위 값을 결정하고, 이 분류를 해당 중요 변경에 할당한다.

[0098] 도 7은, 예를 들어, 하나 이상의 컴퓨팅-기반 장치(102) 상에서, 시스템 관리자에 의해 정의된 비인증 상호작용의 수행을 금지하기 위한 예시적인 방법(700)을 도시한다. 비인증 상호작용의 예는 판독 동작이나 기록 동작을 수행하도록 인증되지 않은 프로그램 또는 엔티티에 의해서 파일 시스템 상에서 수행되는 이러한 동작을 포함한다.

[0099] 방법이 설명되는 순서는 제한적으로 해석되어서는 아니되며, 임의 수의 설명되는 방법 블록들이 그 방법 또는 대안 방법을 구현하기 위해 임의 순서로 조합될 수 있다. 또한, 개별 블록들은 여기에 설명되는 발명의 사상 및 범위를 벗어나지 않고 방법으로부터 삭제될 수 있다. 또한, 방법은 임의의 적절한 하드웨어, 소프트웨어, 펌웨어, 또는 이들의 조합으로 구현될 수 있다.

[0100] 블록 702에서, 시스템 상에서 실행되는 프로그램과 연관된 정보가 수신된다. 이러한 정보는 프로그램 및 파일 시스템 및/또는 구성 설정들 사이의 상호작용과 연관된다. 쿼리 모듈(210)은 하나 이상의 컴퓨팅-기반 장치(102) 상의 프로그램에 의해서 수행되는 상호작용과 관련된 정보에 대해 로그 스토리지(218) 및/또는 아카이브 컬렉션(106)을 쿼리할 수 있다. 쿼리의 수행에 의해서 구한 정보는 하나 이상의 속성을 특징으로 한다.

[0101] 블록 704에서, 시스템상에서 실행되는 프로그램의 속성은 사전정의된 리스트에 포함된 복수의 인가되거나 인가되지 않은 프로그램/프로세서의 속성과 비교된다. 예를 들어, 쿼리 모듈(210)은 프로그램의 속성, 가령 프로그램 유형과 사전정의된 리스트에 포함된 프로그램의 속성을 비교한다.

[0102] 블록 706에서, 이러한 속성이 인가되지 않은 프로그램/프로세서 또는 상호작용의 속성에 대응하는지가 결정된다. 예를 들어, 시스템(100) 상에서 실행되는 프로그램의 속성이 인가되지 않은 상호작용과 연관된 속성에 대응하는 경우(즉, 블록 706에서 '예' 경로를 따르는 경우), 프로그램과 연관된 상호작용은 계속될 수 없다(즉, 블록 708). 반대로, 시스템(100) 상에서 실행되는 프로그램의 속성이 비인가 상호작용과 연관된 속성에 대응하지 않는 경우(즉, 블록 706에서 '아니오' 경로를 따르는 경우), 프로그램과 연관된 상호작용은 계속될 수 있다(즉, 블록 710).

[0103] 도 8은 하나 이상의 컴퓨팅-기반 장치(들)상에 인스톨된 프로그램의 하나 이상의 확장성 포인트(EP)를 검출하기 위한 예시적인 방법(800)을 나타낸다. EP들은 컴퓨터 애플리케이션의 동적 로딩 및 실행을 제어하는 상호작용들을 포함한다. 방법이 설명되는 순서는 제한적으로 해석되어서는 아니되며, 임의 수의 설명되는 방법

블록들이 그 방법 또는 대안 방법을 구현하기 위해 임의 순서로 조합될 수 있다. 또한, 개별 블록들은 여기에 설명되는 발명의 사상 및 범위를 벗어나지 않고 방법으로부터 삭제될 수 있다. 또한, 방법은 임의의 적절한 하드웨어, 소프트웨어, 펌웨어, 또는 이들의 조합으로 구현될 수 있다.

[0104] 블록 802에서, 사전 상호작용들(즉, 주 프로그램의 실행 전에 실행을 위해 시스템 메모리에 로딩된 다양한 프로그램에 관련된 상호작용들)이 검사된다. 예를 들어, 쿼리 모듈(210)은 주 프로그램의 실행 전에 실행을 위해 하나 이상의 컴퓨팅-기반 장치(102)의 메모리에 로딩된 다양한 프로그램에 관련된 상호작용들에 대해 로그 스토리지(218) 및/또는 아카이브 컬렉션(106)에 쿼리함으로써 주 프로그램에 대한 잠재적 직접 확장성 포인트(EP)들을 식별할 수 있다. 쿼리 모듈(210)은 주 프로그램의 실행 전 2초와 같은 소정 시간 범위 내의 실행을 위해 메모리에 로딩된 다양한 프로그램에 관련된 상호작용들에 대해 쿼리할 수 있다.

[0105] 블록 804에서, 실행을 위해 컴퓨팅-기반 장치의 시스템 메모리에 로딩된 주 프로그램의 파일명을 참조하는 사전 상호작용들을 발견하기 위한 검사가 수행된다. 예를 들어, 쿼리 모듈(210)은 주 프로그램을 참조하는 다양한 프로그램과 연관된 상호작용에 대해 쿼리할 수 있거나, 컴퓨팅-기반 장치(102) 상의 주 프로그램의 실행과 연관된 상호작용들에 대해 쿼리할 수 있다. 쿼리 모듈(210)은 주 프로그램의 파일명, 주 프로그램의 프로그램 ID 등과 같은 다양한 속성을 포함하는 상호작용들에 대해 쿼리할 수 있다.

[0106] 블록 806에서, 주 프로그램의 파일명을 참조하는 사전 상호작용들이 직접 EP들로서 플래그된다. 예를 들어, 쿼리 모듈(210)은 주 프로그램을 참조하거나 주 프로그램의 실행과 연관된 모든 사전 상호작용들에 대해 쿼리함으로써 주 프로그램에 대한 직접 EP들을 식별할 수 있다.

[0107] 도 9는 하나 이상의 컴퓨팅-기반 장치(102)로부터 프로그램을 언인스톨한 결과 후에 남겨진 누설 항목들을 검출하기 위한 예시적인 방법(900)을 나타낸다. 방법이 설명되는 순서는 제한적으로 해석되어서는 아니되며, 임의 수의 설명되는 방법 블록들이 그 방법 또는 대안 방법을 구현하기 위해 임의 순서로 조합될 수 있다. 또한, 개별 블록들은 여기에 설명되는 발명의 사상 및 범위를 벗어나지 않고 방법으로부터 삭제될 수 있다. 또한, 방법은 임의의 적절한 하드웨어, 소프트웨어, 펌웨어, 또는 이들의 조합으로 구현될 수 있다.

[0108] 블록 902에서, 컴퓨팅-기반 장치 및/또는 시스템상에 로딩된 각각의 프로그램과 연관된 인스톨 파일 및 설정 변경들이 분류되고 열거된다. 일 구현에서, 열거는 컴퓨팅-기반 장치의 운영 체제에 등록된 프로그램들의 리스트의 생성을 포함한다.

[0109] 예를 들어, 시스템 내의 컴퓨팅-기반 장치들 상에 인스톨된 모든 프로그램은 물론, 컴퓨팅-기반 장치들 상의 프로그램들과 연관된 모든 운영 체제 인스톨 파일을 검출하기 위해 시스템(100)이 스캐닝될 수 있다. 컴퓨팅-기반 장치들 상에 인스톨된 모든 프로그램 및/또는 그 프로그램들과 연관된 모든 운영 체제 인스톨 파일을 리스트 내에 배치함으로써 이들을 열거할 수 있다.

[0110] 쿼리 모듈(210)은 시스템(100)의 스캐닝을 통해, 하나 이상의 컴퓨팅-기반 장치(102) 상에 인스톨된 모든 프로그램을 검출함으로써 컴퓨팅-기반 장치들(102) 상의 모든 운영 체제 인스톨 파일을 분류하고 열거한다. 예를 들어, 쿼리 모듈(210)은 시스템(100) 내의 하나 이상의 컴퓨팅-기반 장치들(102)의 운영 체제에 등록된 모든 프로그램에 대해 로그 스토리지(218) 및/또는 아카이브 컬렉션(106)에 쿼리할 수 있다. 발견된 프로그램들은 쿼리 모듈(210), 에이전트(110) 등과 같은 다양한 장치에 의해 분류되고 열거될 수 있다. 더욱이, 발견된 프로그램들과 연관된 컴퓨팅-기반 장치들(102) 상의 모든 운영 체제 인스톨 파일이 쿼리 모듈(210), 에이전트(110) 등과 같은 다양한 장치에 의해 분류되고 열거될 수 있다.

[0111] 블록 904에서, 언인스톨된 프로그램들과 연관된 파일들 및 레지스트리 설정들을 포함하는, 컴퓨팅-기반 장치 및/또는 시스템 상에 존재하는 모든 파일 및 레지스트리 설정이 열거된다. 이는 언인스톨된 프로그램들에 대한 파일들 및 레지스트리 설정들을 포함하는, 컴퓨팅-기반 장치 및/또는 시스템 상에 인스톨된 모든 프로그램의 파일들 및 레지스트리 설정들에 대한 컴퓨팅-기반 장치 및/또는 시스템 상의 메모리의 스캐닝을 포함할 수 있다. 예를 들어, 쿼리 모듈(210)은 컴퓨팅-기반 장치들(102) 상에 인스톨된 모든 프로그램의, 프로그램 ID들과 같은 식별자들에 대응하는 모든 파일 및 레지스트리 설정들을 구하기 위해 로그 스토리지(218) 및/또는 아카이브 컬렉션(106)에 쿼리할 수 있다.

[0112] 블록 906에서, 운영 체제에 등록된 프로그램들과 연관된 파일들 및 레지스트리 설정들이 컴퓨팅-기반 장치 및/또는 시스템(100) 상에 인스톨된 프로그램들의 파일들 및 레지스트리 설정들과 비교된다. 예를 들어, 쿼리 모듈(210)은 컴퓨팅-기반 장치들(102)의 운영 체제에 등록된 프로그램들과 연관된 열거된 파일들 및 레지스트리 설정들의, 프로그램 ID들과 같은 식별자들을 컴퓨팅-기반 장치들(102) 상에 인스톨된 모든 프로그램의 파

일들 또는 설정들의 식별자들과 비교할 수 있다.

[0113] 블록 908에서, 양쪽 리스트 상에 존재하는, 프로그램들과 연관된 파일들 및 레지스트리 설정들은 고려에서 배제될 수 있다. 컴퓨팅-기반 장치(102) 및/또는 시스템(100)으로부터 언인스톨된 프로그램들에 대응하는 파일들 및 레지스트리 설정들인 나머지 파일들 및 레지스트리 설정들은 누설 파일들이라 할 수 있으며, 컴퓨팅-기반 장치 및/또는 시스템으로부터 제거될 수 있다. 예를 들어, 쿼리 모듈(210)은 컴퓨팅-기반 장치들(102)의 운영 체제에 등록된 프로그램들과 연관된 파일들 및 레지스트리 설정들의 프로그램 ID들과 같은 식별자들을 컴퓨팅-기반 장치들(102) 상에 인스톨된 프로그램들과 연관된 파일들 및 레지스트리 설정들의 식별자들과 상관시킬 수 있다. 상관되지 않은 프로그램들과 연관된 파일들 및 레지스트리 설정들은 쿼리 모듈(210)에 의해 누설 파일들로서 지칭되며, 에이전트(110) 및 쿼리 모듈(210) 등과 같은 요소들에 의해 컴퓨팅-기반 장치들(102)로부터 제거될 수 있다.

[0114] 도 10은 하나 이상의 컴퓨팅-기반 장치(102) 상에 인스톨된 일반적인 오구성, 구식 소프트웨어 버전 등을 포함하는 무효 파일들을 검출하기 위한 예시적인 방법(1000)을 나타낸다. 예를 들어, 온-디스크 실행가능 파일들을 대체한 후에, 소프트웨어 업그레이드가 영향받는 프로세스들의 재시작에 실패할 때, 무효 파일들이 발생한다. 결과적으로, 무효 파일이 발견된 컴퓨팅-기반 장치(102)는 업그레이드를 무시하고, 구식 파일로부터의 프로그램을 계속 실행할 것이다. 방법이 설명되는 순서는 제한적으로 해석되어서는 아니되며, 임의 수의 설명되는 방법 블록들이 그 방법 또는 대안 방법을 구현하기 위해 임의 순서로 조합될 수 있다. 또한, 개별 블록들은 여기에 설명되는 발명의 사상 및 범위를 벗어나지 않고 방법으로부터 삭제될 수 있다. 또한, 방법은 임의의 적절한 하드웨어, 소프트웨어, 펌웨어, 또는 이들의 조합으로 구현될 수 있다.

[0115] 블록 1002에서, 컴퓨팅-기반 장치 및/또는 시스템 상에 로딩된 프로그램들이 분류되고 열거된다. 일 구현에서, 열거는 컴퓨팅-기반 장치의 운영 체제에 등록된 프로그램들의 리스트의 생성을 포함한다. 예를 들어, 시스템 내의 컴퓨팅-기반 장치들 상에 인스톨된 모든 프로그램을 검출하기 위해 시스템이 스캐닝될 수 있다. 컴퓨팅-기반 장치들 상에 인스톨된 모든 프로그램을 리스트에 배치함으로써 이들을 열거할 수 있다.

[0116] 가능한 일 구현에서, 쿼리 모듈(210)은 하나 이상의 컴퓨팅-기반 장치(102) 상에 등록된 모든 프로그램을 검출하는, 시스템(100)의 스캐닝을 실행하여 컴퓨팅-기반 장치들(102) 상의 운영체제에 등록된 모든 프로그램을 분류하고 열거한다. 예를 들어, 쿼리 모듈(210)은 시스템(100) 내의 하나 이상의 컴퓨팅-기반 장치들(102)의 운영 체제에 등록된 모든 프로그램에 대해 로그 스토리지(218) 및/또는 아카이브 컬렉션(106)에 쿼리할 수 있다. 발견된 프로그램들은 쿼리 모듈(210), 에이전트(110) 등과 같은 다양한 장치 및/또는 엔티티에 의해 분류되고 열거될 수 있다.

[0117] 블록 1004에서, 컴퓨팅-기반 장치 및/또는 시스템 상에 등록된 모든 프로그램의 최종 로드 시간들은 물론, 컴퓨팅-기반 장치 및/또는 시스템 상에 등록된 프로그램들과 연관된 파일들에 대한 최종 로드 시간들이 취득된다. 예를 들어, 쿼리 모듈(210)은 컴퓨팅-기반 장치들(102) 상에 등록된 프로그램들의 최종 로드 시간 및/또는 시스템(100)내의 컴퓨팅-기반 장치들(102) 상에 등록된 프로그램들과 함께 인스톨된 시스템 동적 링크 라이브러리(DLL)과 같은 파일의 최종 로드 시간들에 대해 로그 스토리지(218) 및/또는 아카이브 컬렉션(106)에 쿼리한다.

[0118] 블록 1006에서, 컴퓨팅-기반 장치 및/또는 시스템 상에 등록된 프로그램들과 연관된 파일들 또는 설정들의 최종 수정 시간들이 취득되고, 프로그램들의 최종 로드 시간들과 비교된다. 예를 들어, 쿼리 모듈(210)은 컴퓨팅-기반 장치(102)상의 운영 체제에 등록된 프로그램의 최종 수정 시간 또는 날짜에 대해 로그 스토리지(218) 및/또는 아카이브 컬렉션(106)에 쿼리할 수 있다. 쿼리 모듈(210)은 최종 수정 시간 또는 날짜와 프로그램의 최종 로드 시간을 비교할 수 있다.

[0119] 블록 1008에서, 비교 동안 발견되는 임의의 불일치가 인식된다. 예를 들어, 컴퓨팅-기반 장치 및/또는 시스템 상에 등록된 프로그램의 최종 로드 시간이 프로그램의 최종 공지된 수정의 시간 또는 날짜보다 더 최근에 발생한 경우, 프로그램은 최종 수정에 응답하지 않았을 가능성이 있다. 이 경우, 프로그램이 최종 시도된 수정에 응답하지 않고 있음을 보고하는 에러 보고가 사용자 또는 시스템 관리자와 같은 엔티티들에게 발행될 수 있다. 대안으로, 프로그램이 수정 버전을 로딩하게 하는 시도가 이루어질 수 있다.

[0120] 예시적인 일 구현에서, 쿼리 모듈(210)은 시스템(100) 내의 컴퓨팅-기반 장치(102) 상의 운영체제에 등록된 프로그램의 최종 로드 시간 및 최종 수정 시간 양자에 대해 쿼리할 수 있다. 쿼리 모듈(210)은 최종 로드 시간과 최종 수정 시간을 비교할 수 있고, 프로그램의 최종 로드 시간이 프로그램의 최종 수정 시간보다 더 최근인 경우, 쿼리 모듈(210)은 프로그램이 최종 시도된 수정에 응답하고 있지 않음을 보고하는 에러 리포트를

사용자 또는 시스템 관리자와 같은 엔티티들에게 발행할 수 있다. 다른 실시예에서, 쿼리 모듈(210)은 프로그램의 최종 시도된 수정을 다시 시도할 수도 있다.

[0121] 예시적인 컴퓨터 환경

[0122] 도 11은 여기에 설명되는 기술들을 구현하는 데 사용될 수 있고 여기에 설명되는 요소들을 전체적으로 또는 부분적으로 나타낼 수 있는 예시적인 범용 컴퓨터 환경(1100)을 나타낸다. 컴퓨터 환경(1100)은 컴퓨팅 환경의 일례에 불과하며, 컴퓨터 및 네트워크 아키텍처들의 용도 또는 기능성의 범위에 관한 어떤 제한을 암시하고자 하는 것이 아니다. 컴퓨터 환경(1100)은 예시적인 컴퓨터 환경(1100) 내에 도시된 컴포넌트들 중 어느 하나 또는 조합에 관한 임의의 종속성 또는 필요 조건을 갖는 것으로 해석되지 않아야 한다.

[0123] 컴퓨터 환경(1100)은 컴퓨터(1102) 형태의 범용 컴퓨팅-기반 장치를 포함한다. 컴퓨터(1102)는 예를 들어, 데스크톱 컴퓨터, 핸드-헬드 컴퓨터, 노트북 또는 랩톱 컴퓨터, 서버 컴퓨터, 게임 콘솔 등일 수 있다. 컴퓨터(1102)의 컴포넌트들은 하나 이상의 프로세서 또는 처리 장치(1104), 시스템 메모리(1106), 및 프로세서(1104)를 비롯한 다양한 시스템 컴포넌트를 시스템 메모리(1106)에 연결시키는 시스템 버스(1108)를 포함할 수 있지만, 이에 제한되는 것은 아니다.

[0124] 시스템 버스(1108)는 메모리 버스 또는 메모리 컨트롤러, 주변 버스, 가속 그래픽 포트, 및 각종 버스 아키텍처 중 임의의 아키텍처를 이용하는 프로세서 또는 로컬 버스를 비롯한 임의의 몇몇 유형의 버스 구조 중 하나 이상을 나타낸다. 예를 들어, 이러한 아키텍처는 ISA(industry standard architecture) 버스, MCA(micro channel architecture) 버스, EISA(Enhanced ISA) 버스, VESA(video electronics standard association) 로컬 버스, 및 메자닌 버스(mezzanine bus)로도 공개된 PCI(peripheral component interconnect) 버스를 포함할 수 있다.

[0125] 컴퓨터(1102)는 통상적으로 각종 컴퓨터 판독가능 매체를 포함한다. 이러한 매체는 컴퓨터(1102)에 의해 액세스 가능하고 휘발성 및 비휘발성, 이동식 및 비이동식 매체를 포함하는 임의의 이용 가능 매체일 수 있다.

[0126] 시스템 메모리(1106)는 랜덤 액세스 메모리(RAM: 1110)와 같은 휘발성 메모리, 및/또는 판독 전용 메모리(ROM: 1112)와 같은 비휘발성 메모리 형태의 컴퓨터 판독가능 매체를 포함한다. 시동 중과 같은 때에, 컴퓨터(1102) 내의 구성요소들 사이의 정보 전송을 돕는 기본 루틴들을 포함하는 기본 입/출력 시스템(BIOS)(1114)이 ROM(1112)에 저장되어 있다. RAM(1110)은 통상적으로 처리 장치(1104)가 즉시 액세스할 수 있고 및/또는 처리 장치 상에서 현재 동작시키고 있는 데이터 및/또는 프로그램 모듈을 포함한다.

[0127] 컴퓨터(1102)는 또한 기타 이동식/비이동식, 휘발성/비휘발성 컴퓨터 저장 매체를 포함할 수 있다. 예를 들어, 도 11은 비이동식, 비휘발성 자기 매체(도시되지 않음)에 기록을 하거나 그로부터 판독을 하는 하드 디스크 드라이브(1116), 이동식, 비휘발성 자기 디스크(1120)(예를 들어, "플로피 디스크")에 기록을 하거나 그로부터 판독을 하는 자기 디스크 드라이브(1118), 및 CD-ROM, DVD-ROM 또는 기타 광 매체 등의 이동식, 비휘발성 광 디스크(1124)에 기록을 하거나 그로부터 판독을 하는 광 디스크 드라이브(1122)를 도시하고 있다. 하드 디스크 드라이브(1116), 자기 디스크 드라이브(1118) 및 광 디스크 드라이브(1122)는 각각 하나 이상의 데이터 매체 인터페이스(1126)에 의해 시스템 버스(1108)에 접속된다. 대안으로, 하드 디스크 드라이브(1116), 자기 디스크 드라이브(1118) 및 광 디스크 드라이브(1122)는 하나 이상의 인터페이스(도시되지 않음)에 의해 시스템 버스(1108)에 접속될 수 있다.

[0128] 디스크 드라이브들 및 이들과 연관된 컴퓨터 판독가능 매체들은 컴퓨터 판독가능 명령어, 데이터 구조, 프로그램 모듈들 및 컴퓨터(1102)용의 다른 데이터의 비휘발성 저장을 제공한다. 하드 디스크(1116), 이동식 자기 디스크(1120) 및 이동식 광 디스크(1124)가 예로서 도시되지만, 자기 카세트 또는 다른 자기 저장 장치, 플래시 메모리 카드, CD-ROM, DVD 또는 다른 광 저장 장치, RAM, ROM, EEPROM 등과 같이 컴퓨터에 의해 액세스 가능한 데이터를 저장할 수 있는 다른 유형의 컴퓨터 판독가능 매체들도 예시적인 컴퓨팅 시스템 및 환경을 구현하는데 사용될 수 있다는 것을 알아야 한다.

[0129] 예를 들어, 운영 체제(1127), 하나 이상의 애플리케이션 프로그램(1128), 다른 프로그램 모듈(1130) 및 프로그램 데이터(1132)를 포함하는 임의 수의 프로그램 모듈들이 하드 디스크(1116), 자기 디스크(1120), 광 디스크(1124), ROM(1112) 및/또는 RAM(1110)에 저장될 수 있다. 이러한 운영 체제(1127), 하나 이상의 애플리케이션 프로그램(1128), 기타 프로그램 모듈(1130) 및 프로그램 데이터(1132)(또는 이들의 소정 조합) 각각은 분산형 파일 시스템을 지원하는 상주 컴포넌트들의 전부 또는 일부를 구현할 수 있다.

[0130] 사용자는 키보드(1134) 및 포인팅 장치(1136)(예를 들어, "마우스") 등의 입력 장치들을 통해 명령 및 정보를

컴퓨터(1102)에 입력할 수 있다. 다른 입력 장치들(1138)(구체적으로 도시되지 않음)은 마이크, 조이스틱, 게임 패드, 위성 안테나, 병렬 포트, 스캐너 등을 포함할 수 있다. 이들 및 기타 입력 장치는 시스템 버스(1108)에 결합된 입출력 인터페이스(1140)를 통해 처리 장치(1104)에 접속되지만, 병렬 포트, 게임 포트, 또는 USB(universal serial bus) 등의 다른 인터페이스 및 버스 구조에 의해 접속될 수도 있다.

[0131] 모니터(1142) 또는 다른 유형의 디스플레이 장치도 비디오 어댑터(1144) 등의 인터페이스를 통해 시스템 버스(1108)에 접속될 수 있다. 모니터(1142) 외에, 다른 출력 주변 장치들은 입출력 인터페이스(1140)를 통해 컴퓨터(1102)에 접속될 수 있는 스피커(도시되지 않음) 및 프린터(1146)와 같은 컴포넌트들을 포함할 수 있다.

[0132] 컴퓨터(1102)는 원격 컴퓨팅-기반 장치(1148)와 같은 하나 이상의 원격 컴퓨터로의 논리적 접속을 사용하여 네트워크화된 환경에서 동작할 수 있다. 예를 들어, 원격 컴퓨팅-기반 장치(1148)는 퍼스널 컴퓨터, 휴대형 컴퓨터, 서버, 라우터, 네트워크 컴퓨터, 피어 장치 또는 기타 통상의 네트워크 노드 등일 수 있다. 원격 컴퓨팅기반 장치(1148)는 컴퓨터(1102)와 관련하여 여기서 설명되는 요소들 및 특징들의 대부분 또는 모두를 포함할 수 있는 휴대형 컴퓨터로서 도시된다.

[0133] 컴퓨터(1102)와 원격 컴퓨터(1148) 사이의 논리적 접속은 근거리 네트워크(LAN: 1150) 및 범용 원거리 네트워크(WAN: 1152)로서 도시된다. 이러한 네트워킹 환경들은 사무실, 전사적 컴퓨터 네트워크(enterprise-wide computer network), 인트라넷 및 인터넷에서 일반적인 것이다.

[0134] LAN 네트워킹 환경에서 구현될 때, 컴퓨터(1102)는 네트워크 인터페이스 또는 어댑터(1154)를 통해 LAN(1150)에 접속된다. WAN 네트워킹 환경에서 구현될 때, 컴퓨터(1102)는 통상적으로 WAN(1152)을 통해 통신을 설정하기 위한 모뎀(1156) 및 기타 수단을 포함한다. 컴퓨터(1102)에 대해 내장형이거나 외장형일 수 있는 모뎀(1156)은 입출력 인터페이스(1140) 또는 기타 적절한 메커니즘을 통해 시스템 버스(1108)에 접속될 수 있다. 도시된 네트워크 접속들은 예시적이며, 이 컴퓨터들(1102, 1148) 사이에 통신 링크(들)를 설정하기 위한 기타 수단이 사용될 수 있다는 것을 이해할 것이다.

[0135] 컴퓨팅 환경(1100)과 함께 도시된 것과 같은 네트워킹 환경에서, 컴퓨터(1102) 및 그의 일부에 관해 도시된 프로그램 모듈들은 원격 메모리 저장 장치에 저장될 수 있다. 예를 들어, 원격 애플리케이션 프로그램들(1158)은 원격 컴퓨터(1148)의 메모리 장치에 위치한다. 예시적인 목적으로, 애플리케이션 프로그램들, 및 운영 체제 등의 기타 실행가능 프로그램 컴포넌트들은 여기서 개별 블록들로서 도시되지만, 이러한 프로그램들 및 컴포넌트들은 다양한 시간에 컴퓨팅-기반 장치(1102)의 상이한 저장 컴포넌트들 내에 위치하며, 컴퓨터의 데이터 프로세서(들)에 의해 실행된다는 것을 인식한다.

[0136] 다양한 모듈 및 기술은 본 명세서에서, 하나 이상의 컴퓨터 또는 다른 장치에 의해 실행되는 프로그램 모듈들과 같은 컴퓨터 실행가능 명령어와 일반적으로 관련하여 설명될 수 있다. 일반적으로, 프로그램 모듈은 특정 태스크를 수행하거나 특정 추상 데이터 유형을 구현하는 루틴, 프로그램, 개체, 컴포넌트, 데이터 구조 등을 포함한다. 통상적으로, 프로그램 모듈의 기능은 다양한 실시예에서 원하는 대로 조합되거나 분산될 수 있다.

[0137] 이러한 모듈 및 기술의 구현은 소정 형태의 컴퓨터 판독가능 매체들 상에 저장되거나 전송될 수 있다. 컴퓨터에 의해 액세스 가능한 매체는 그 어떤 것이든지 컴퓨터 판독가능 매체가 될 수 있다. 예로서, 컴퓨터 판독가능 매체는 "컴퓨터 저장 매체" 및 "통신 매체"를 포함할 수 있으나, 이에 제한되는 것은 아니다.

[0138] "컴퓨터 저장 매체"는 컴퓨터 판독가능 명령어, 데이터 구조, 프로그램 모듈 또는 기타 데이터와 같은 정보를 저장하는 임의의 방법 또는 기술로 구현되는 휘발성 및 비휘발성, 이동식 및 비이동식 매체를 포함한다. 컴퓨터 저장 매체는 RAM, ROM, EEPROM, 플래시 메모리 또는 기타 메모리 기술, CD-ROM, DVD(digital versatile disk) 또는 기타 광 저장 장치, 자기 카세트, 자기 테이프, 자기 디스크 저장 장치 또는 다른 자기 저장 장치, 또는 원하는 정보를 저장하는 데 사용될 수 있고 컴퓨터에 의해 액세스될 수 있는 임의의 다른 매체를 포함하지만, 이에 제한되는 것은 아니다.

[0139] 대안으로, 프레임워크의 일부는 하드웨어 또는 하드웨어, 소프트웨어 및/또는 펌웨어의 조합으로 구현될 수 있다. 예를 들어, 하나 이상의 ASIC(application specific integrated circuit) 또는 PLD(programmable logic device)가 프레임워크의 하나 이상의 부분을 구현하도록 설계 또는 프로그래밍될 수 있다.

[0140] 결론

[0141] 시스템 관리 및 분석을 위한 실시예들이 구조적인 특징들 및/또는 방법들에 고유한 언어로 설명되었지만, 첨부된 청구범위의 주제는 설명된 특정 특징들 또는 방법들로 제한될 필요가 없다는 것을 이해해야 한다. 오히

려, 특정 특징들 및 방법들은 스레드 인터셉션 및 분석의 예시적인 구현들로서 개시된다.

도면의 간단한 설명

[0010] 첨부 도면들을 참조하여 상세한 설명이 기술된다. 도면들에서, 참조 번호의 가장 좌측의 숫자(들)는 참조 번호가 최초로 나타나는 도면을 식별한다. 도면들 전반에서 동일 번호들은 동일한 특징들 및 컴포넌트들을 참조하는 데 사용된다.

[0011] 도 1은 시스템 관리를 위한 예시적인 아키텍처를 도시하는 도면.

[0012] 도 2는 예시적인 컬렉션 서버를 도시하는 도면.

[0013] 도 3은 생성된 통지를 도시하는 예시적인 시각적 인터페이스를 도시하는 도면.

[0014] 도 4는 주 프로그램의 실행에 의존하는 일 프로그램의 실행의 의존성을 도시하는 예시적인 시각적 인터페이스를 도시하는 도면.

[0015] 도 5는 시스템의 PS에서의 수정과 연관된 데이터를 캡처하기 위한 예시적인 방법(들)을 도시하는 도면.

[0016] 도 6은 중요 변경을 분류하기 위한 예시적인 방법(들)을 도시하는 도면.

[0017] 도 7은 미인증 상호작용의 실행을 금지하기 위한 예시적인 방법(들)을 도시하는 도면.

[0018] 도 8은 하나 이상의 확장성 포인트를 검출하기 위한 예시적인 방법(들)을 도시하는 도면.

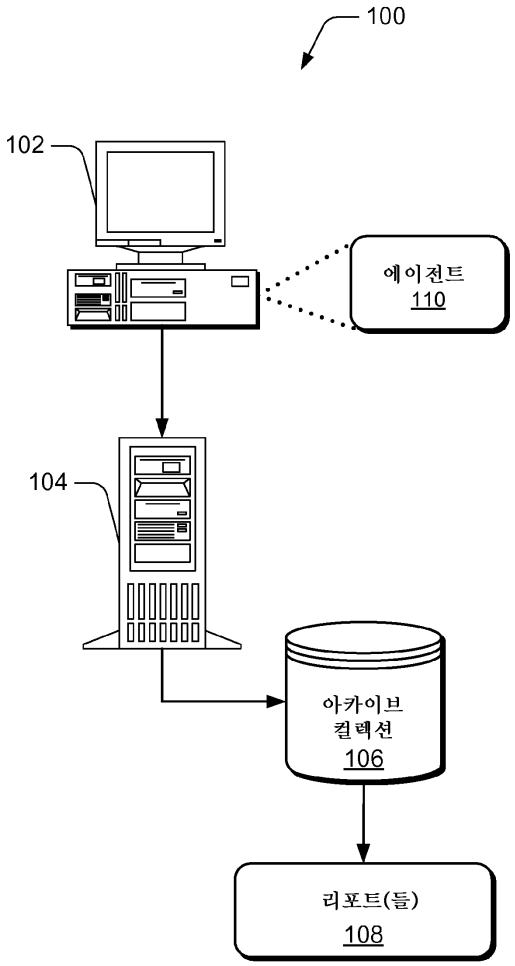
[0019] 도 9는 누설 항목들을 검출하기 위한 예시적인 방법(들)을 도시하는 도면.

[0020] 도 10은 일반적인 오구성 또는 무효 파일들을 검출하기 위한 예시적인 방법(들)을 도시하는 도면.

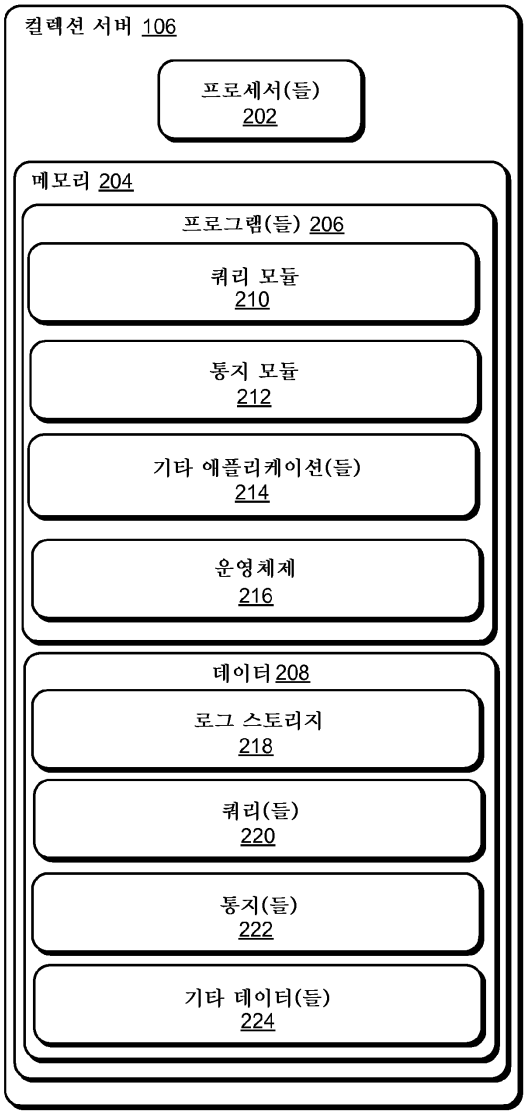
[0021] 도 11은 예시적인 컴퓨터 환경을 도시하는 도면.

도면

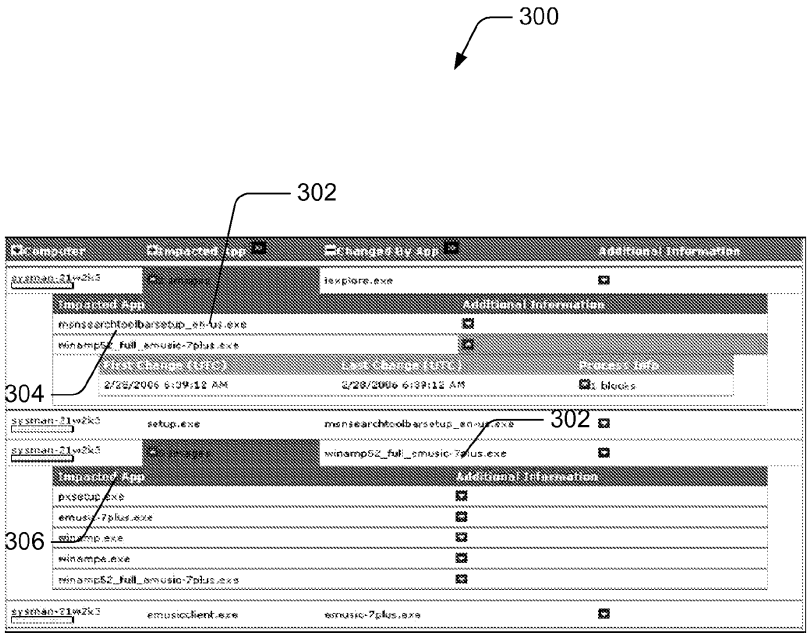
도면1



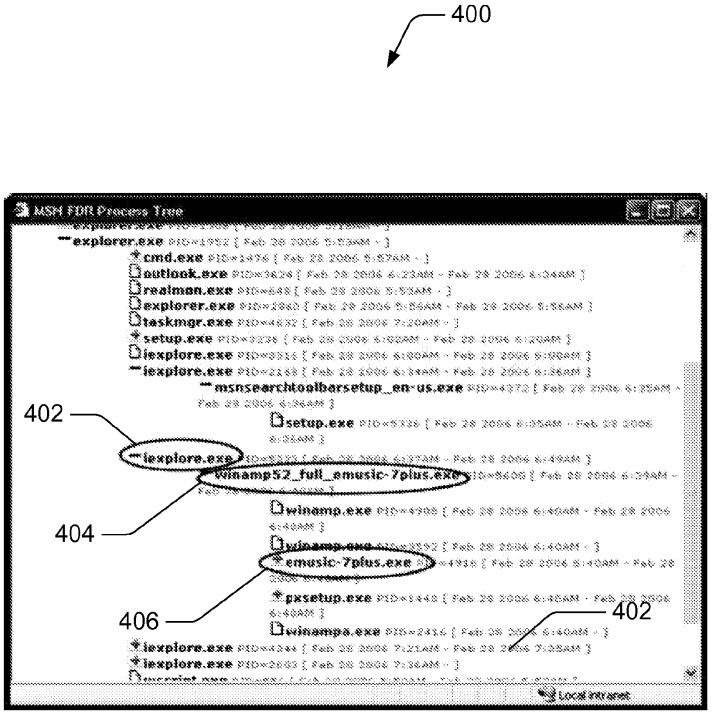
도면2


컬렉션 서버 106
프로세서(들)
202
메모리 204
프로그램(들) 206
쿼리 모듈
210
통지 모듈
212
기타 애플리케이션(들)
214
운영체제
216
데이터 208
로그 스토리지
218
쿼리(들)
220
통지(들)
222
기타 데이터(들)
224

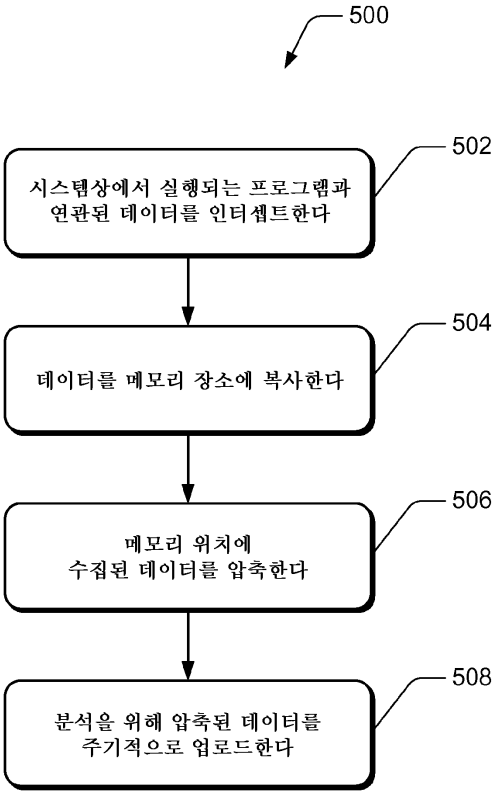
도면3


300
302
304
306
302
Computer
Impacted App
Changed By App
Additional Information
iexplore.exe
msnsearchtoolbarsetup_en-us.exe
winamp52_full_emusic-7plus.exe
setup.exe
pxsetup.exe
emusic-7plus.exe
winamp.exe
winampa.exe
emusicclient.exe

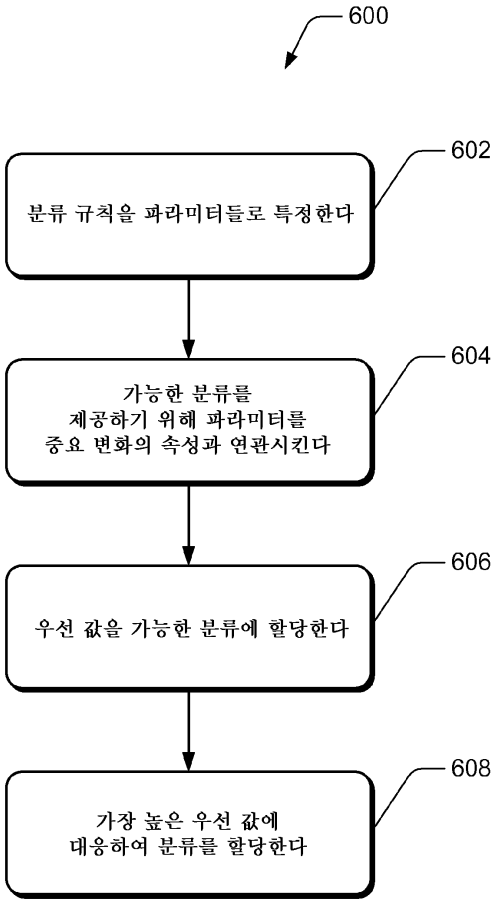
도면4


400
MSN FDR Process Tree
explorer.exe
cmd.exe
outlook.exe
realmon.exe
explorer.exe
taskmgr.exe
setup.exe
iexplore.exe
iexplore.exe
msnsearchtoolbarsetup_en-us.exe
setup.exe
iexplore.exe
winamp52_full_emusic-7plus.exe
winamp.exe
emusic-7plus.exe
pxsetup.exe
winampa.exe
iexplore.exe
iexplore.exe
402
404
406
402
Local intranet

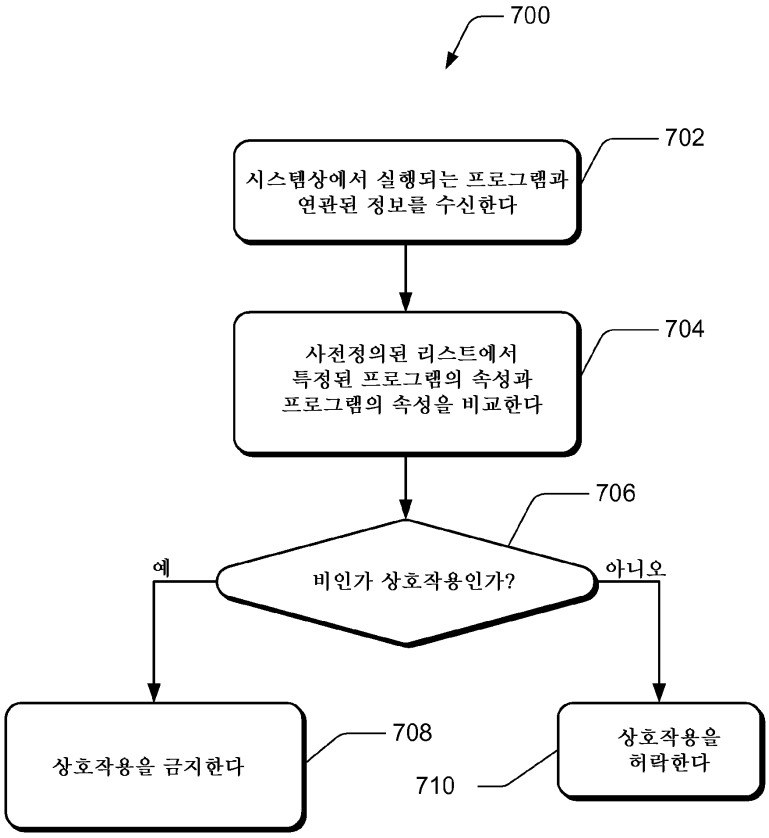
도면5


500
502
시스템상에서 실행되는 프로그램과 연관된 데이터를 인터셉트한다
504
데이터를 메모리 장소에 복사한다
506
메모리 위치에 수집된 데이터를 압축한다
508
분석을 위해 압축된 데이터를 주기적으로 업로드한다

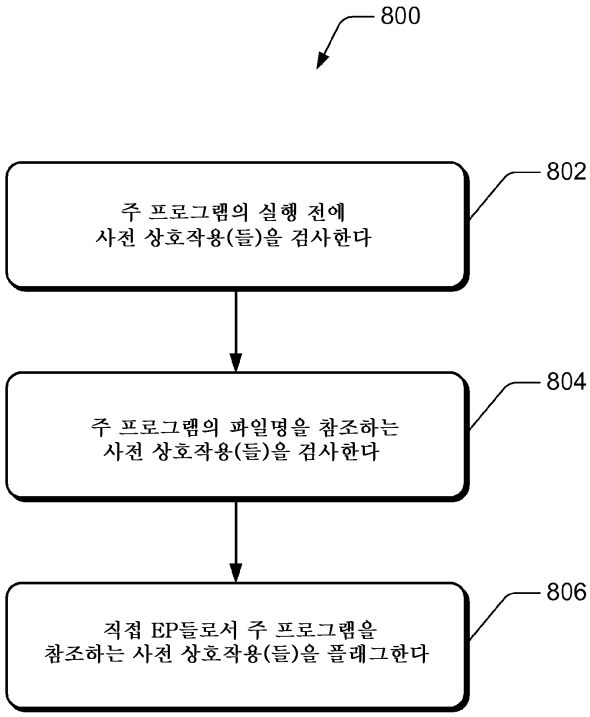
도면6


600
602
분류 규칙을 파라미터들로 특정한다
604
가능한 분류를
제공하기 위해 파라미터를
중요 변화의 속성과 연관시킨다
606
우선 값을 가능한 분류에 할당한다
608
가장 높은 우선 값에
대응하여 분류를 할당한다

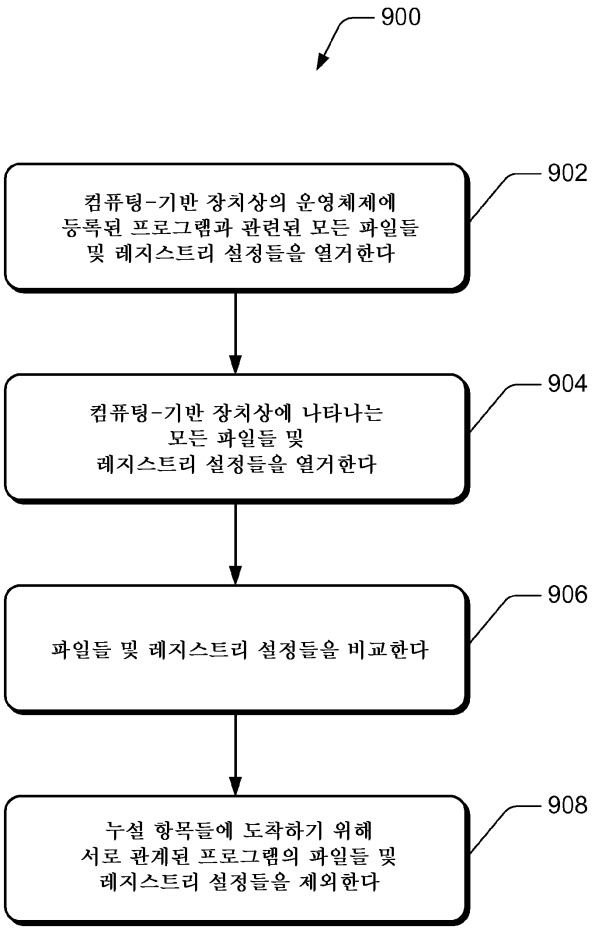
도면7


700
시스템상에서 실행되는 프로그램과 연관된 정보를 수신한다
702
사전정의된 리스트에서 특정된 프로그램의 속성과 프로그램의 속성을 비교한다
704
706
비인가 상호작용인가?
예
아니오
상호작용을 금지한다
708
상호작용을 허락한다
710

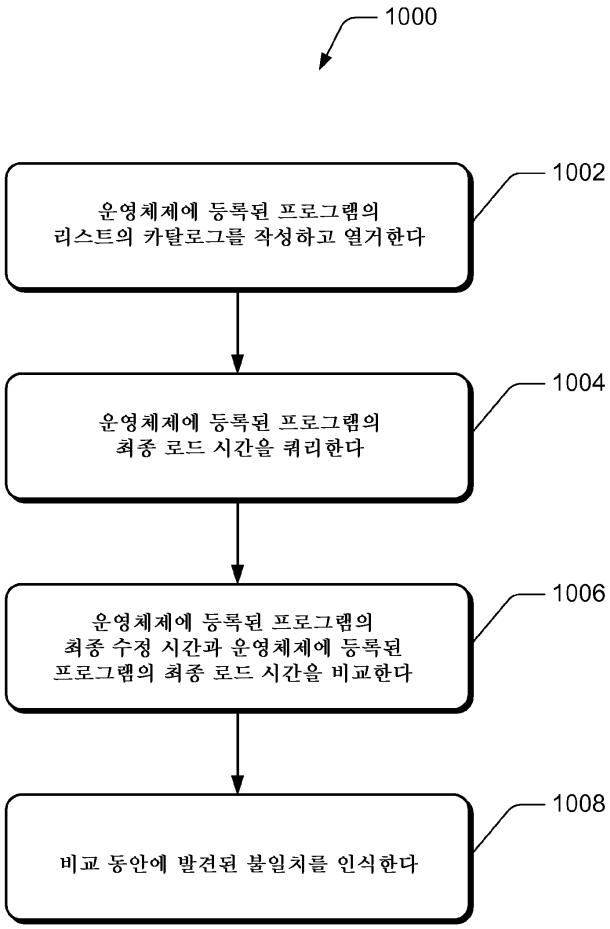
도면8


800
주 프로그램의 실행 전에 사전 상호작용(들)을 검사한다
802
주 프로그램의 파일명을 참조하는 사전 상호작용(들)을 검사한다
804
직접 EP들로서 주 프로그램을 참조하는 사전 상호작용(들)을 플래그한다
806

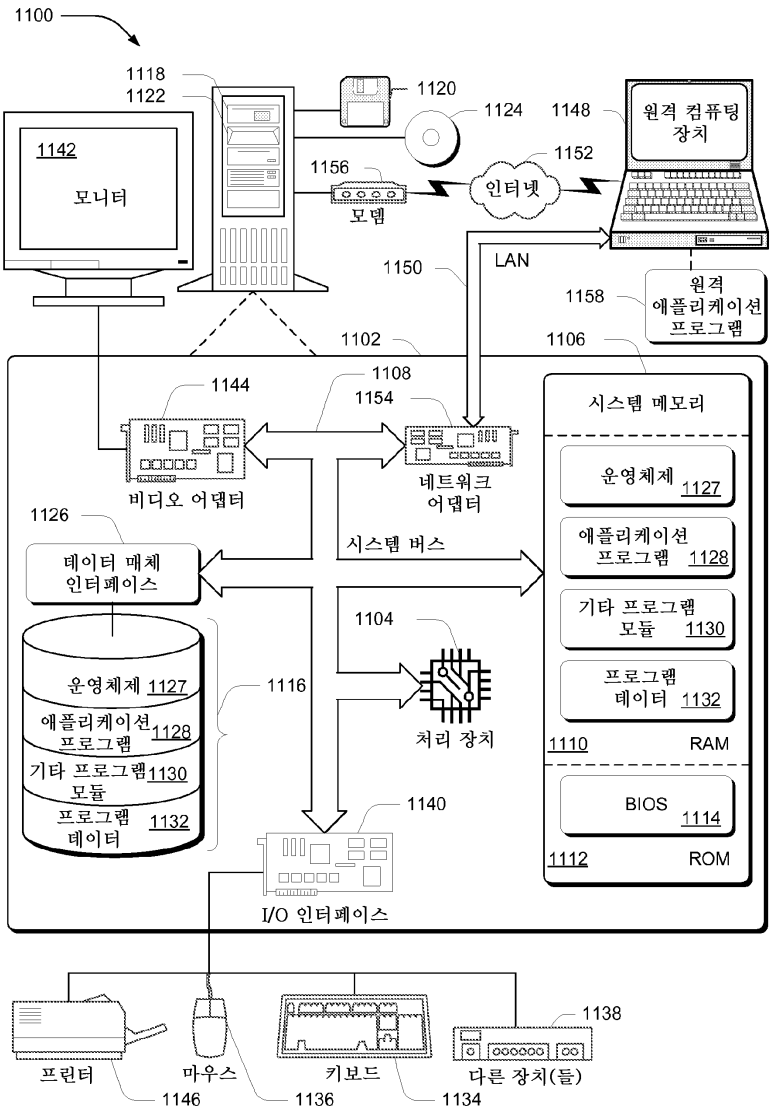
도면9


900
902
컴퓨팅-기반 장치상의 운영체제에
등록된 프로그램과 관련된 모든 파일들
및 레지스트리 설정들을 열거한다
904
컴퓨팅-기반 장치상에 나타나는
모든 파일들 및
레지스트리 설정들을 열거한다
906
파일들 및 레지스트리 설정들을 비교한다
908
누설 항목들에 도착하기 위해
서로 관계된 프로그램의 파일들 및
레지스트리 설정들을 제외한다

도면10


1000
운영체제에 등록된 프로그램의
리스트의 카탈로그를 작성하고 열거한다
1002
운영체제에 등록된 프로그램의
최종 로드 시간을 쿼리한다
1004
운영체제에 등록된 프로그램의
최종 수정 시간과 운영체제에 등록된
프로그램의 최종 로드 시간을 비교한다
1006
비교 동안에 발견된 불일치를 인식한다
1008

도면11


1100
1118
1122
1120
1124
1148
원격 컴퓨팅 장치
1142
모니터
1156
모뎀
인터넷
1152
1150
LAN
1158
원격 애플리케이션 프로그램
1102
1106
1108
1144
1154
시스템 메모리
운영체제 1127
네트워크 어댑터
비디오 어댑터
애플리케이션 프로그램 1128
1126
시스템 버스
데이터 매체 인터페이스
기타 프로그램 모듈 1130
1104
운영체제 1127
1116
애플리케이션 프로그램 1128
처리 장치
프로그램 데이터 1132
기타 프로그램 1130 모듈
1110
RAM
1140
BIOS 1114
프로그램 데이터 1132
1112
ROM
I/O 인터페이스
1138
프린터
마우스
키보드
다른 장치(들)
1146
1136
1134