(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2011년01월14일
(11) 등록번호 10-1007899
(24) 등록일자 2011년01월06일

(51) Int. Cl.
G06F 11/00 (2006.01)
(21) 출원번호 10-2006-7025550
(22) 출원일자(국제출원일자) 2005년05월04일
심사청구일자 2009년06월18일
(85) 번역문제출일자 2006년12월04일
(65) 공개번호 10-2007-0050402
(43) 공개일자 2007년05월15일
(86) 국제출원번호 PCT/US2005/015933
(87) 국제공개번호 WO 2005/107424
국제공개일자 2005년11월17일
(30) 우선권주장
10/839,613 2004년05월04일 미국(US)
(56) 선행기술조사문헌
KR1020020043994 A
US20040015719 A1
GB2391650 A
전체 청구항 수 : 총 30 항

(73) 특허권자
아크사이트, 인코퍼레이티드
미국 캘리포니아 95014 쿠퍼티노 리절츠 웨이 5
(72) 발명자
사우라브 쿠마르
미국 95051 캘리포니아주 산타 클라라 클라마스 애비뉴 2016아파트 1
티드웰 케니
미국 94924 캘리포니아주 로스알토스 에스. 스프링거 로드 914
(74) 대리인
문기상, 문두현

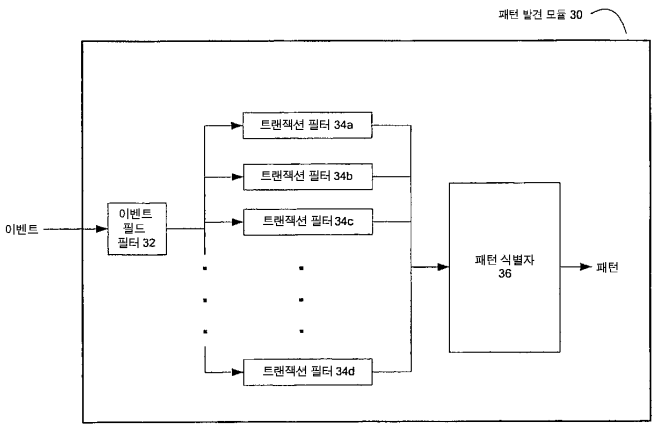
심사관 : 이경홍

(54) 네트워크 보안 시스템에서의 패턴 발견

(57) 요 약

패턴은 네트워크 보안 시스템에 의해 수집된 보안 이벤트에서 발견될 수 있다. 일 실시예에서, 본 발명은 여러 모니터 디바이스로부터 보안 이벤트를 수집하고 저장한다. 일 실시예에서, 저장된 보안 이벤트의 서브셋은 관리자에게 이벤트 스트림으로 제공된다. 일 실시예에서, 본 발명은 이벤트 스트림에서 이전에 알려지지 않은 하나 이상의 이벤트 패턴을 관리자가 발견한다.

대 표 도 - 도2



특허청구의 범위

청구항 1

복수의 보안 이벤트를 포함하는 이벤트 스트림에서 이벤트 패턴을 발견하는 방법으로서,

트랜잭션 파라미터에 기초하여 복수의 트랜잭션을 생성하는 단계 - 여기서 각 트랜잭션은 상기 복수의 보안 이벤트의 서브셋을 나타냄 - ;

상기 복수의 트랜잭션에 기초하여 트랜잭션 트리를 발생하는 단계 - 여기서 트랜잭션 트리는 하나의 루트 노드와 이 루트 노드 이외의 다른 노드를 포함하고, 각 다른 노드는 보안 이벤트를 나타냄 - ; 및

상기 트랜잭션 트리로부터 상기 이벤트 패턴을 추출하는 단계 - 여기서 상기 트랜잭션 트리로부터 상기 이벤트 패턴을 추출하는 단계는 상기 다른 노드 중 제 1 다른 노드로부터 제 2 다른 노드의 서포트(supprot)의 하락을 관찰함으로써 상기 트랜잭션 트리를 파싱하는 단계를 포함하고, 상기 다른 노드의 서포트는 상기 다른 노드에 의해서 나타나는 보안 이벤트를 포함하는 트랜잭션의 개수를 나타냄 - ;

를 포함하는 이벤트 패턴을 발견하는 방법.

청구항 2

삭제

청구항 3

제 1 항에 있어서,

상기 트랜잭션 파라미터는 시간의 간격을 포함하고, 여기서 각 트랜잭션은 상기 시간의 간격 동안 수신된 이벤트를 나타내는 이벤트 패턴을 발견하는 방법.

청구항 4

제 1 항에 있어서,

상기 트랜잭션 파라미터는 이벤트 필드의 값을 포함하고, 여기서 각 트랜잭션은 상기 이벤트 필드의 값을 포함하는 이벤트를 나타내는 이벤트 패턴을 발견하는 방법.

청구항 5

제 1 항에 있어서,

상기 트랜잭션 파라미터는 시점(point in time)을 포함하고, 여기서 각 트랜잭션은 상기 시점 근방에서 수신된 이벤트를 나타내는 이벤트 패턴을 발견하는 방법.

청구항 6

제 1 항에 있어서,

상기 복수의 트랜잭션에 기초하여 트랜잭션 트리를 발생하는 단계는, 상기 복수의 트랜잭션 내에서 보안 이벤트의 서포트에 기초하여 상기 트랜잭션 트리를 발생하는 단계를 포함하는 이벤트 패턴을 발견하는 방법.

청구항 7

삭제

청구항 8

제 1 항에 있어서,

상기 이벤트 패턴을 이벤트 상관 규칙(event correlation rule)으로 변환시키는 옵션을 유저에게 제공하는 단계를 더 포함하는 이벤트 패턴을 발견하는 방법.

청구항 9

제 1 항에 있어서,

상기 이벤트 패턴을 사용하여 이벤트 상관 규칙을 생성하는 단계를 더 포함하는 이벤트 패턴을 발견하는 방법.

청구항 10

제 1 항에 있어서,

상기 복수의 이벤트는 이벤트 데이터 베이스에 저장되어 있는 이벤트 패턴을 발견하는 방법.

청구항 11

규칙을 생성하는 방법으로서,

유저가 미리 저장된 보안 이벤트의 서브셋을 선택할 수 있도록 구성된 패턴 발견 툴을 표시하는 단계;

상기 유저 선택에 응답하여,

트랜잭션 파라미터에 기초하여 복수의 트랜잭션을 생성하는 단계 - 여기서 각 트랜잭션은 선택된 상기 보안 이벤트의 서브셋을 나타냄 - ;

상기 복수의 트랜잭션에 기초하여 트랜잭션 트리를 발생하는 단계 - 여기서 트랜잭션 트리는 하나의 루트 노드와 이 루트 노드 이외의 다른 노드를 포함하고, 각 다른 노드는 보안 이벤트를 나타냄 - ; 및

상기 트랜잭션 트리로부터 이벤트 패턴을 추출하는 단계 - 여기서 상기 트랜잭션 트리로부터 이벤트 패턴을 추출하는 단계는 상기 다른 노드 중 제 1 다른 노드로부터 제 2 다른 노드의 서포트의 하락을 관찰함으로써 상기 트랜잭션 트리를 파싱하는 단계를 포함하고, 상기 다른 노드의 서포트는 상기 다른 노드에 의해서 나타나는 보안 이벤트를 포함하는 트랜잭션의 개수를 나타냄 - ;

유저가 동작을 수행할 수 있도록 구성된 규칙 발생 툴을 표시하는 단계; 및

유저 동작에 응답하여, 상기 이벤트 패턴을 상관 규칙으로 변환하는 단계를 포함하는 규칙을 생성하는 방법.

청구항 12

제 11 항에 있어서,

상기 유저 동작은 단일 마우스-클릭을 포함하는 규칙을 생성하는 방법.

청구항 13

삭제

청구항 14

복수의 보안 이벤트를 포함하는 이벤크 스트림에서 이벤트 패턴을 발견하는 시스템으로서,

트랜잭션 파라미터에 기초하여 복수의 트랜잭션을 생성하도록 구성된 트랜잭션 필터 - 여기서 각 트랜잭션은 상기 복수의 보안 이벤트의 서브셋을 나타냄 - ;

상기 복수의 트랜잭션에 기초하여 트랜잭션 트리를 발생하도록 구성된 트리 구축자 - 여기서 상기 트랜잭션 트리는 하나의 루트 노드와 이 루트 노드 이외의 다른 노드를 포함하고, 각 다른 노드는 보안 이벤트를 나타냄 - ; 및

상기 트랜잭션 트리로부터 상기 이벤트 패턴을 추출하도록 구성된 트리 파서 - 여기서 상기 트리 파서는 상기 다른 노드 중 제 1 다른 노드로부터 제 2 다른 노드의 서포트의 하락을 관찰함으로써 상기 트랜잭션 트리를 파싱하고, 상기 다른 노드의 서포트는 상기 다른 노드에 의해서 나타나는 상기 보안 이벤트를 포함하는 트랜잭션의 개수를 나타냄 - ;

를 포함하는 이벤트 패턴을 발견하는 시스템.

청구항 15

삭제

청구항 16

제 14 항에 있어서,

상기 트랜잭션 파라미터는 시간의 간격을 포함하고, 여기서 각 트랜잭션은 상기 시간의 각격 동안 수신된 이벤트를 나타내는 이벤트 패턴을 발견하는 시스템.

청구항 17

제 14 항에 있어서,

상기 트랜잭션 파라미터는 이벤트 필드의 값을 포함하고, 여기서 각 트랜잭션은 상기 이벤트 필드의 값을 포함하는 이벤트를 나타내는 이벤트 패턴을 발견하는 시스템.

청구항 18

제 14 항에 있어서,

상기 트랜잭션 파라미터는 시점을 포함하고, 여기서 각 트랜잭션은 상기 시점 근방에서 수신된 이벤트를 나타내는 이벤트 패턴을 발견하는 시스템.

청구항 19

제 14 항에 있어서,

상기 복수의 트랜잭션에 기초하여 트랜잭션 트리를 발생하도록 구성된 트리 구축자는 상기 복수의 트랜잭선 내에서 보안 이벤트의 서포트에 기초하여 트랜잭션 트리를 발생하는 것을 포함하는 이벤트 패턴을 발견하는 시스템.

청구항 20

삭제

청구항 21

제 14 항에 있어서,

상기 벤트 패턴을 이벤트 상관 규칙으로 변환시키는 툴을 유저에게 제공하는 유저 인터페이스를 더 포함하는 이벤트 패턴을 발견하는 시스템.

청구항 22

프로세서에 의해서 실행될 때 프로세서가 동작을 수행하게 하는 명령을 나타낸 데이터를 저장한 기계 판독 가능한 매체로서,

상기 동작은,

트랜잭션 파라미터에 기초하여 복수의 트랜잭션을 생성하는 단계 - 여기서 각 트랜잭션은 상기 복수의 보안 이벤트의 서브셋을 나타냄 - ;

상기 복수의 트랜잭션에 기초하여 트랜잭션 트리를 발생하는 단계 - 여기서 트랜잭션 트리는 하나의 루트 노드와 이 루트 노드 이외의 다른 노드를 포함하고, 각 다른 노드는 보안 이벤트를 나타냄 - ; 및

상기 트랜잭션 트리로부터 상기 이벤트 패턴을 추출하는 단계 - 여기서 상기 트랜잭션 트리로부터 상기 이벤트 패턴을 추출하는 단계는 상기 다른 노드 중 제 1 다른 노드로부터 제 2 다른 노드의 서포트의 하락을 관찰함으로써 상기 트랜잭션 트리를 파싱하는 단계를 포함하고, 상기 다른 노드의 서포트는 상기 다른 노드에 의해서 나타나는 보안 이벤트를 포함하는 트랜잭션의 개수를 나타냄 - ;

을 포함하는 기계 판독 가능한 매체.

청구항 23

삭제

청구항 24

제 22 항에 있어서,

상기 복수의 트랜잭션에 기초하여 트랜잭션 트리를 발생하는 단계는 상기 복수의 트랜잭션 내에서 보안 이벤트의 서포트에 기초하여 상기 트랜잭션 트리를 발생하는 단계를 포함하는 기계 판독 가능한 매체.

청구항 25

삭제

청구항 26

제 22 항에 있어서,

상기 동작은, 상기 이벤트 패턴을 이벤트 상관 규칙으로 변환하는 옵션을 유저에게 제공하는 단계를 더 포함하는 기계 판독 가능한 매체.

청구항 27

제 22 항에 있어서,

상기 동작은, 상기 이벤트 패턴을 사용하여 이벤트 상관 규칙을 생성하는 단계를 더 포함하는 기계 판독 가능한 매체.

청구항 28

제 1 항에 있어서,

상기 복수의 보안 이벤트에서 각 이벤트로부터 필드를 제거하는 단계를 더 포함하는 이벤트 패턴을 발견하는 방법.

청구항 29

제 28 항에 있어서,

상기 필드의 값은 각 이벤트에 고유한 이벤트 패턴을 발견하는 방법.

청구항 30

제 4 항에 있어서,

상기 이벤트 필드는 소스 어드레스(source address)를 포함하는 이벤트 패턴을 발견하는 방법.

청구항 31

제 4 항에 있어서,

상기 이벤트 필드는 목적지 어드레스를 포함하는 이벤트 패턴을 발견하는 방법.

청구항 32

제 14 항에 있어서,

상기 복수의 보안 이벤트에서 각 이벤트로부터 필드를 제거하도록 구성된 이벤트 필드 필터를 더 포함하는 이벤트 패턴을 발견하는 시스템.

청구항 33

제 22 항에 있어서,

상기 동작은, 상기 복수의 보안 이벤트에서 각 이벤트로부터 필드를 제거하는 단계를 더 포함하는 기계 판독 가능한 매체.

청구항 34

제 1 항에 있어서,

상기 제 1 다른 노드로부터 상기 제 2 다른 노드의 서포트의 하락을 관찰함으로써 상기 트랜잭션 트리를 파싱하는 단계는,

상기 루트 노드에서 시작하는 상기 트랜잭션 트리의 브랜치를 가로지르는(traversing) 단계 - 여기서 상기 브랜치는 상기 제 1 다른 노드를 통해 상기 루트 노드로부터 제 2 다른 노드로 연장함 - ;

상기 제 1 다른 노드로부터 상기 제 2 다른 노드의 서포트의 하락을 관찰하는 단계; 및

상기 이벤트 패턴이 상기 제 1 다른 노드에 의해서 나타나는 보안 이벤트를 포함하고 있고, 상기 제 2 다른 노드에 의해서 나타나는 보안 이벤트를 포함하지 않고 있는지를 판정하는 단계를 포함하는 이벤트 패턴을 발견하는 방법.

청구항 35

제 11 항에 있어서,

상기 제 1 다른 노드로부터 상기 제 2 다른 노드의 서포트의 하락을 관찰함으로써 상기 트랜잭션 트리를 파싱하는 단계는,

상기 루트 노드에서 시작하는 상기 트랜잭션 트리의 브랜치를 가로지르는 단계 - 여기서 상기 브랜치는 상기 제 1 다른 노드를 통해 상기 루트 노드로부터 제 2 다른 노드로 연장함 - ;

상기 제 1 다른 노드로부터 상기 제 2 다른 노드의 서포트의 하락을 관찰하는 단계; 및

상기 이벤트 패턴이 상기 제 1 다른 노드에 의해서 나타나는 보안 이벤트를 포함하고 있고, 상기 제 2 다른 노드에 의해서 나타나는 보안 이벤트를 포함하지 않고 있는지를 판정하는 단계를 포함하는 규칙을 생성하는 방법.

청구항 36

제 14 항에 있어서,

상기 트리 파서는, 또한

상기 루트 노드에서 시작하는 상기 트랜잭션 트리의 브랜치를 가로지르고 - 여기서 상기 브랜치는 상기 제 1 다른 노드를 통해 상기 루트 노드로부터 제 2 다른 노드로 연장함 - ;

상기 제 1 다른 노드로부터 상기 제 2 다른 노드의 서포트의 하락을 관찰하고; 그리고

상기 이벤트 패턴이 상기 제 1 다른 노드에 의해서 나타나는 보안 이벤트를 포함하고 있고, 상기 제 2 다른 노드에 의해서 나타나는 보안 이벤트를 포함하지 않고 있는지를 판정하도록 구성된 이벤트 패턴을 발견하는 시스템.

청구항 37

제 22 항에 있어서,

상기 제 1 다른 노드로부터 상기 제 2 다른 노드의 서포트의 하락을 관찰함으로써 상기 트랜잭션 트리를 파싱하는 단계는,

상기 루트 노드에서 시작하는 상기 트랜잭션 트리의 브랜치를 가로지르는 단계 - 여기서 상기 브랜치는 상기 제 1 다른 노드를 통해 상기 루트 노드로부터 제 2 다른 노드로 연장함 - ;

상기 제 1 다른 노드로부터 상기 제 2 다른 노드의 서포트의 하락을 관찰하는 단계; 및

상기 이벤트 패턴이 상기 제 1 다른 노드에 의해서 나타나는 보안 이벤트를 포함하고 있고, 상기 제 2 다른 노드에 의해서 나타나는 보안 이벤트를 포함하지 않고 있는지를 판정하는 단계를 포함하는 기계 판독 가능한 매체.

명 세 서

기 술 분 야

[0001] 본 발명은 이종 및 동종 소스로부터 보안 이벤트를 캡쳐(capturing)하고, 캡쳐된 보안 이벤트에서 패턴을 발견하기 위한 컴퓨터 기반 시스템에 관한 것이다.

배 경 기 술

[0002] 컴퓨터 네트워크 및 시스템은 현대의 직무를 위해 긴요한 툴이 되고 있다. 오늘날 상상할 수 있는 사실상의 모든 주제에서의 테라비트(terabit)의 정보는 전 세계에 걸쳐 유저들에 의해 그러한 네트워크의 전역에 저장되고, 액세스(access)된다. 이들 정보 중 다수는 어느 정도는 기밀이고, 그 기밀의 보호가 필요하다. 따라서, 침입 탐지 시스템(intrusion detection system(IDS))은 인증되지 않은 사람들 및/또는 디바이스에 의해 컴퓨터 네트워크 및 그 내부에 저장된 정보에 대해 액세스를 획득하려는 시도의 적발을 돕도록 발전되어 왔다. 게다가, 라우터 및 방화벽과 같은 네트워크 디바이스는 그러한 시도를 적발하는데 사용될 수 있는 실행 로그(activity log)를 관리한다.

[0003] 침입 탐지는 컴퓨터 네트워크 또는 시스템 내에서나 컴퓨터 네트워크 또는 시스템에 대하여 부적당하고, 부정확한 또는 변칙적인 실행을 탐지하는 기술로서 간주될 수 있다. 침입 탐지에 대한 가장 일반적인 방식은 통계적인 편차 탐지 및 패턴 매칭 탐지이다. 호스트에서의 악성 실행을 탐지하도록 호스트에서 동작하는 IDS는 호스트 래퍼(host wrapper)/개인 방화벽 또는 에이전트 기반 소프트웨어의 형태로 존재하는 호스트 기반 IDS(HIDS)라 불리고, 네트워크 데이터 플로에서 동작하는 IDS는 네트워크 기반 IDS(NIDS)라 불린다. 호스트 기반 침입 탐지는 모니터되는 시스템(호스트)에서 소프트웨어를 로딩하고, 로그 파일 및/또는 호스트의 감시 에이전트(auditing agent)를 데이터 소스로서 사용한다. 이와 달리, 네트워크 기반 침입 탐지 시스템은 네트워크 세그먼트에서 트래픽을 모니터하고, 모니터된 트래픽을 데이터 소스로서 사용한다. 네트워크 인터페이스 카드에 의해 캡쳐되는 패킷이 서명(signature)과 매칭하는 경우, 관심 패킷으로서 고려된다.

[0004] 데이터 소스와 관계없이, 지식 기반 방식 및 행위 기반 방식의 두 개의 상보적인 침입 탐지 방식이 있다. 오늘날 사용하는데 있어서 대부분의 IDS 툴은 지식 기반 방식이다. 지식 기반 침입 탐지 기술은 캡쳐된 데이터와, 취약성을 이용하는 주지의 기술에 관한 정보를 비교한다. 매칭이 탐지될 경우에는, 경보가 트리거된다. 반면, 행위 기반 침입 탐지 기술은 시스템 또는 유저의 통상적인 또는 예상된 행위로부터의 일탈(다양한 수단에 의해 수집되는 참조 정보로부터 추출된 모델)을 적발함으로써 침입을 탐지하도록 시도된다. 의심되는 일탈이 적발될 경우, 경보를 발생한다.

[0005] 지식 기반 방식의 장점은 오경보율에 대한 가능성이 매우 낮고, 침입 탐지 시스템에 의한 침입 전후 관계데 대해 분석이 정밀하며, 지식 기반 방식의 침입 탐지 시스템을 사용하는 보안 책임자가 예방 또는 교정 작업을 하기 용이하다는 점들이 있다. 단점은 주지의 공격에 대해 필요한 정보를 수집하여, 새로운 취약성 및 환경으로 업데이트하여 관리하는데 있어서 어려움을 갖는다는 점이다.

[0006] 행위 기반 방식의 장점은 새롭고 예상하지 못한 취약성을 이용한 공격을 탐지할 수 있다는 것이다. 또한, 행위 기반 방식은 시스템 기종을 거의 가리지 않는다. 하지만, 높은 오경보율이 행위 기반 방식 기술의 중요한 단점으로서 일반적으로 입증되어 있고, 공격 행위가 시간이 지남에 따라 변화할 수 있기 때문에, 오경보율의 빈도가 증가할 수 있다.

[0007] 호스트 기반 또는 네트워크 기반 구현의 채택 여부와 상기 구현이 지식 기반 또는 행위 기반인가의 여부에 관계없이, 침입 탐지 시스템은 통상적인 시스템 사용과 진짜 침입(적절한 경보가 수반됨) 사이를 구별하는 능력으로서만 유용할 뿐이다. 침입이 탐지되어 적절한 담당자에게 즉각적인 방식으로 통지될 수 있을 경우에는, 보호되고 있는 시스템에 대해 손상을 피할 수 있는 조치가 취해질 수 있다. 그렇지 않을 경우에는, 손상을 피할 수 있는 조치가 제공될 수 없다. 따라서, 컴퓨터 네트워크 또는 시스템에 손상을 입히려는 공격에 대해 효과적으로 대처하도록 정확하고 시기적절한 침입 탐지 및 경보 발생을 제공할 수 있는 시스템이 요구된다.

발명의 상세한 설명

[0008] 패턴은 네트워크 보안 시스템에 의해 수집된 보안 이벤트에서 발견될 수 있다. 일 실시예에서, 본 발명은 여러 모니터 디바이스로부터 보안 이벤트를 수집하여 저장한다. 일 실시예에서, 저장된 보안 이벤트의 서브셋(subset)은 관리자에게 이벤트 스트림(event stream)으로 제공된다. 일 실시예에서, 본 발명은 제공된 이벤트 스트림에서 이전에 알려지지 않은 하나 이상의 이벤트 패턴을 관리자가 발견한다.

실 시 예

[0014] 본 시스템은 여러 도시된 예를 참조하여 기술되지만, 이들 예가 본 발명의 폭넓은 정신 및 범위를 한정하는 것은 아니다. 예를 들어, 본 발명의 일 실시예로, 분배된 에이전트, 관리자 및 콘솔을 기술하였다. 본 발명의 일반적인 개념 및 범위는 훨씬 넓고, 임의의 컴퓨터 기반 또는 네트워크 기반의 보안 시스템까지 연장할 수 있다. 또한, 본 발명을 더 기술하려는 시도로써 시스템의 구성 요소로부터 보내질 수 있는 메시지 및 시스템의 구성 요소에 의해 사용될 수 있는 데이터 개요의 예가 주어졌지만, 본 발명의 개념 및 범위 모두를 포함하는 예가 되는 것은 아니며, 그와 같이 간주하지 않아야 한다.

[0015] 이하, 상세한 설명의 일부는 컴퓨터 메모리에서의 데이터 동작을 나타내는 알고리즘 및 부호의 관점에서 표현된다. 이들 알고리즘의 설명 및 표현은 컴퓨터 과학 기술에 숙련된 당업자들에 의해 사용되는 수단이며, 그들 기술의 요지를 컴퓨터 과학 기술에 능숙한 다른 사람들에게 가장 효과적으로 전달한다. 알고리즘은 본 명세서에서는 바람직한 결과에 이르는 단계가 모순없이 배열되도록 일반적으로 이해된다. 상기 단계는 물리적인 양에 대해 물리적인 조작을 필요로 하는 단계이다. 통상적으로, 필수적이지는 않지만, 이들 양은 저장, 전송, 결합, 비교 등의 조작 가능한 전기 또는 자기적 신호의 형태를 취한다. 주로 통상적으로 사용하는데 있어, 이들 신호를 비트, 값, 요소, 부호, 문자, 용어, 수 등으로 참조하는 것이 때로는 편리하다. 하지만, 이들 용어 및 유사한 용어들 모두는 고유의 물리량과 관련되어 있으며, 단지 이들 물리량에 적용되는 편리한 라벨(label)이라는 점을 유념해야 한다. 특정하게 언급되지 않는 경우, 본 발명의 명세서를 통해, "처리", "계산", "산출", "결정", "표시" 등과 같은 용어를 사용하여 컴퓨터 시스템의 레지스터 및 메모리 내부의 물리(전자) 량으로 표현되는 데이터를 컴퓨터 시스템 메모리 또는 레지스터 또는 다른 그러한 정보 저장체나 전송체 또는 표시 디바이스 내부의 물리량으로 표현되는 유사한 다른 데이터로 조작하고 변환시키는 컴퓨터 시스템 또는 유사한 전자 컴퓨팅 디바이스의 동작 및 프로세스를 참조한다.

[0016] 상기와 같이, 본 발명의 일 실시예는 컴퓨터 소프트웨어, 즉 하나 이상의 컴퓨터 프로세서/시스템에 의해 실행될 경우, 지정된 동작을 수행하도록 프로세서/시스템에 명령하는 컴퓨터 판독 가능 명령으로 예시된다. 상기 컴퓨터 소프트웨어는 하드 드라이브, CD-롬, DVD-롬, 읽기 전용 메모리(read-only memory), 읽기 쓰기 메모리(read-write memory) 등과 같은 하나 이상의 컴퓨터 판독 가능 매체에서 상주(常駐)할 수 있다. 상기 소프트웨어는 하나 이상의 이들 매체에 분배되거나, 하나 이상의 컴퓨터 네크워크(예를 들어, 인터넷)의 전역에서 다운로드용으로 이용될 수 있다. 포맷과 관계없이, 본 명세서에 기술하고 있는 컴퓨터 프로그래밍 기술, 렌더링 기술, 및 프로세싱 기술은 단지 본 발명의 여러 측면을 구현하는데 사용되는 여러 타입의 컴퓨터 프로그래밍 기술, 렌더링 기술, 및 프로세싱 기술의 실례일 뿐이다. 이들 실례는 본 발명을 한정하는 것은 아니며, 본 상세한 설명 다음의 청구 범위를 참조하면 가장 잘 이해될 것이다.

[0017] 도 1을 참조하면, 본 발명의 일 실시예에 따라 구조된 컴퓨터 기반 네트워크 보안 시스템(10)의 예가 도시되어 있다. 시스템(10)은 에이전트(12), 하나 이상의 관리자(14) 및 하나 이상의 콘솔(16)(브라우저 기반 버전 포함)을 포함한다. 일부 실시예에서, 에이전트, 관리자 및/또는 콘솔은 단일 플랫폼으로 결합될 수 있으나, 둘 또는 셋 이상의 플랫폼(도시된 예와 같은)으로 분배될 수도 있다. 이와 같은 멀티-티어 설계구조의 사용은 컴퓨터 네트워크 또는 시스템이 증가함에 따라 확장성을 지원한다.

[0018] 에이전트(12)는 다양한 네트워크 보안 디바이스 및/또는 애플리케이션으로부터 실시간(또는 거의 실시간) 로컬 이벤트 데이터 캡처와 필터링을 효율적으로 제공하는 소프트 프로그램이다. 보안 이벤트의 주요 소스는 방화벽, 침입 탐지 시스템 및 오퍼레이팅 시스템 로그와 같은 일반적인 네트워크 보안 디바이스이다. 에이전트(12)는 이벤트 로그 또는 메시지를 산출하는 임의의 소스로부터 이벤트를 수집할 수 있고, 본래의 장치에서, 네트워크 내 통합 지역에서 및/또는 간이 망 관리 프로토콜(SNMP) 트랩을 통해서 동작할 수 있다.

[0019] 에이전트(12)는 수동 및 자동의 양쪽 프로세스 및 관련된 구성 파일에 의해 구성가능하다. 각각의 에이전트(12)는 정규화 구성 요소, 시간 보정 구성 요소, 집합 구성 요소, 배치 구성 요소, 해결자 구성 요소, 수송 구성 요소 및/또는 부가 구성 요소를 갖는 하나 이상의 소프트웨어 모듈을 포함한다. 이들 구성 요소는 구성 파일에서의 적절한 명령을 통해 활성화 및/또는 비활성화될 수 있다.

[0020] 관리자(14)는 에이전트로부터 수신되는 이벤트를 더욱 통합, 필터 및 상관시키는 서버 기반 구성 요소일 수 있으며, 규칙 엔진(18) 및 집중화된 이벤트 데이터베이스(20)를 사용한다. 관리자(14)의 한 역할은 보안 활동에 있어서 완전하고, 전사적인(enterprise wide) 화상을 구성하도록(데이터 베이스 관리자(22)를 통해) 모든 실시간 및 히스토릭 이벤트 데이터를 캡처하고 저장하는 것이다. 관리자(14)는 집중화된 경영, 통지(하나 이상의

통지자(24)를 통해) 및 보고 뿐만 아니라 지식 기반(28) 및 케이스 관리 워크플로(workflow)를 제공한다. 관리자(14)는 임의의 컴퓨터 하드웨어 플랫폼에 배치될 수 있고, 일 실시예로는 이벤트 데이터 저장 구성 요소를 구현하도록 Oracle™ 데이터베이스와 같은 관계형 데이터베이스 관리 시스템을 이용한다. 관리자(14) 및 에이전트(12) 사이의 통신은 양방향 지향성(예를 들어, 관리자(14)에게 플랫폼 호스팅 에이전트(12)에게 명령을 전송하도록 허용)이고, 암호화시킬 수 있다. 일부 장치에서는, 관리자(14)가 멀티 에이전트(12)에 대해 집선기로써 동작할 수 있고, 다른 관리자(예를 들어, 기업 본부에 배치된)에게 정보를 포워드할 수 있다.

[0021] 콘솔(16)은 보안 전문가에게 이벤트 모니터링, 규칙 인증, 부수적인 조사, 및 보고와 같은 관리 및 운영 작업을 매일 수행하도록 허용하는 컴퓨터(예를 들어, 워크 스테이션) 기반 애플리케이션이다. 액세스 제어 리스트는 다수의 보안 전문가에게 그들 책임하에 적절한 자신의 견해, 상관 규칙, 경보, 리포트 및 지식 기반을 각각 갖는 동일한 시스템 및 이벤트 데이터를 사용하도록 허용한다. 단일 관리자(14)는 다수의 콘솔(16)을 서포트할 수 있다.

[0022] 일부 실시예에서는, 콘솔(16)의 브라우저 기반 버전은 보안 이벤트, 지식 기반 항목, 리포트, 통지 및 케이스에 대해 액세스할 수 있도록 사용될 수 있다. 즉, 관리자(14)는 개인 컴퓨터 또는 핸드헬드 컴퓨터(콘솔(16)을 대체)에 호스트된 웹 브라우저를 통해 액세스 가능한 웹 서버 구성 요소를 포함하여, 콘솔(16) 기능의 일부 또는 모두를 제공한다. 브라우저 액세스는 콘솔(16)로부터 떨어져 있는 보안 전문가와 파트타임 유저에게 특히 유용하다. 콘솔(16)과 관리자(12) 사이의 통신은 양방향 지향성이고 암호화될 수 있다.

[0023] 상술한 구조를 통해, 본 발명은 중앙 집중 또는 분산 처리 환경을 서포트할 수 있다. 이것은 기관이 시스템(10)의 단일 인스턴스를 구현하고 액세스 제어 리스트를 사용하여 유저를 파티션할 수 있기 때문에 유용하다. 이와 달리, 기관은 다수의 그룹의 각각에 대해 개별 시스템(10)을 배치하도록 선택하여, "마스터" 레벨에서 그 결과를 통합할 수 있다. 이와 같은 배치는 지리적으로 분산된 피어 그룹(peer group)이 표준 근무 시간 동안 현재 작업중인 그룹에 대해 주요 감독 의무를 진행함으로써 서로 협력하는 "폴로 더 선(Follow-The-Sun)" 구성을 또한 이룰 수 있다. 시스템(10)은 비지니스 부서가 개별적으로 작업하고 중앙 관리 기능으로 롤업(rollup) 기능을 서포트하는 기업의 위계제도에 또한 배치될 수 있다.

[0024] 도 1에 도시된 예인 네트워크 보안 시스템은 본 명세서의 참조로써, 전체 내용이 포함되어 2002년 12월 2일에 출원되고 현재는 미국 특허 제7,376,969로 등록되어 있는 "멀티플 네트워크 보안 디바이스로부터 이벤트의 실시간 모니터링 및 분석"이라 발명의 명칭이 부여된 미국 특허 출원 10/308415호에 더 상세하게 기술되어 있다.

[0025] 본 발명의 일 실시예에서, 네트워크 보안 시스템(10)은 패턴 인식 성능(pattern recognition capability)을 포함한다. 일 실시예에서, 이는 도 1에 나타낸 패턴 발견 모듈(30)을 통해 관리자(14)에서 구현된다. 도 1에 나타낸 바와 같이, 일 실시예에서, 이벤트 데이터베이스(20)에 저장된 한 세트의 보안 이벤트는 패턴 발견용 데이터베이스 관리자(22)를 통해 패턴 발견 모듈(30)에 제공될 수 있다.

[0026] 일 실시예에서는, 유저는 이전에 저장된 보안 이벤트의 시간 주기를 재연하도록 특정함으로써, 콘솔 인터페이스(16)를 사용하여 이벤트 데이터베이스(20)로부터 이벤트 재연을 시작할 수 있다. 이벤트 재연을 위한 몇몇 실시예는 본 명세서의 참조로써 전체 내용이 포함되어 2002년 12월 2일에 출원된 "네트워크 보안 시스템용 활동 및 디버깅용 방법 및 장치"라 발명의 명칭이 부여된 미국 특허 출원 10/308416호에 더 상세하게 기술되어 있다.

[0027] 일 실시예에서, 상기의 재연된 이벤트 스트림은 패턴 발견 모듈에 제공된다. 다른 실시예에서는, 에이전트(12)로부터 수집된 보안 이벤트는 가끔 수집된 이벤트의 스냅샷을 취하는 패턴 발견 모듈(30)에 직접 제공된다. 일실시예에서는, 패턴 발견 모듈(30)은 제공된 이벤트 스트림에서 관심 패턴을 찾도록 - 즉, 발견(discover), 탐지(detect)하도록 - 구성된다.

[0028] 일 실시에에서는, 패턴 발견 모듈(30)에 의해 탐지된 패턴과 규칙 엔진(18)에 의해 식별된 패턴 사이의 차이는 패턴 발견 모듈이 보안 이벤트에서 이전에 알려지지 않은 패턴을 찾고, 규칙 엔진(18)이 수집된 보안 이벤트에서 알려진 패턴을 찾는다는 점이다. 즉, 규칙 엔진(18)은 이벤트 스트림에 적용되는 상관 규칙을 갖고, 이러한 각 규칙은 특정한 패턴을 찾는다. 하지만, 일 실시예에서는, 패턴 발견 모듈(30)은 사전에 정의된 규칙 또는 필터와 같이 암호화된 도메인 지식에 의지하지 않고, 이전에 인지되지 않은 패턴을 찾는다. 일 실시예에서는, 패턴 발견 모듈에 의해 발견된 패턴은 규칙 엔진(18)에 의해 사용되기 위해 상관 규칙으로 변환된 이후, 보안 이벤트를 상관시킨다.

[0029] 일 실시예에서는, 규칙 엔진(18)의 동작과 패턴 발견 모듈(30)의 동작은 규칙 엔진이 이전에 알려진 패턴의 특정한 인스턴스를 찾아낸다는 점에 있어서 서로 다르다. 하지만, 일 실시예에서는, 패턴 발견 모듈(30)은 단지

패턴을 식별 가능하도록 구성된다. 이벤트 스트림에서 패턴을 찾아내기 위해, 새롭게 탐지된 패턴 기반 규칙이 규칙 엔진(18)에 제공될 수 있다.

[0030] 패턴 발견 모듈(30)의 일 실시예는 도 2를 참조하여 더욱 상세히 기술된다. 상술한 바에 따르면, 일 실시예에서 패턴 발견 모듈(30)의 입력은 데이터베이스 관리자(22)에 의해 이벤트 데이터베이스(20)로부터 제공된 이벤트 스트림(보안 이벤트의 수집물)이다. 일 실시예에서는, 패턴 발견 모듈(30)은 각 이벤트에서 선택된 이벤트 필드만을 관리하는 이벤트 필드 필터(32)를 포함한다. 예를 들어, 이벤트 필드는 패턴 발견을 위해서는 유용하지 않기 때문에, 각 이벤트에 대해 유일하게 패턴을 발견할 수 있는 이벤트 필드(예를 들어, 유일한 이벤트 식별자)는 메모리를 보호하고 처리 속도를 증가시키도록 이벤트 필드 필터(32)에 의해 필터링될 수 있다.

[0031] 이벤트 필드 필터(event field filter)는 트랜잭션 필터(34)에 의해 사용된 최소한의 필드를 관리하도록 구성된다. 일 실시예에서는, 이벤트 필드 필터(32)는 최소한의 이벤트 네임 필드와 소스 및 목적지 필드를 관리한다. 다른 실시예에서는, 센서 벤더, 프로덕트, 카테고리, 소스 및 목적지 포트 등의 부가적인 이벤트 필드가 관리된다. 일 실시예에서는, 각 트랜잭션 필터(34)를 위한 하나의 다중 이벤트 필드 필터(32)가 있다. 그러한 실시예에서는, 각 이벤트 필드 필터(32)는 각각의 트랜잭션 필터(34)에 의해 사용된 최소한의 이벤트 필드를 관리할 것이다. 이벤트 필드 필터(32)는 임의의 개개의 이벤트 필드 또는 임의의 이벤트 필드의 서브셋을 필터링 또는 유지하도록 구성될 수 있고, 상기 리스트화된 이벤트 필드의 임의의 결합을 포함한다.

[0032] 일 실시예에서는, (유지되는 이벤트 필드로 표현되는)보안 이벤트는 트랜잭션을 발생시킨다. 일 실시예에서는, 트랜잭션은 일부 트랜잭션 파라미터에 의해 관련되는 보안 이벤트를 표현하는 항목의 그룹으로 규정된다. 예를 들어, 일부 시간프레임(예를 들어, 5분, 30분, 하루, 일주일, 한달 ... 등) 내에 수신된 모든 이벤트는 하나의 트랜잭션이 될 수 있다. 일 실시예에서, 일부 시간의 주기 동안 수신된 모든 이벤트인 이벤트 스트림은 임의의 개수의 시간 기반 트랜잭션으로 분할될 수 있다.

[0033] 트랜잭션은 또한 이벤트 비율을 사용하여 규정될 수 있다. 일 실시예에서는, 트랜잭션은 보안 이벤트의 버스트(burst)로 규정된다. 즉, 다수의 트랜잭션은 이벤트 비율에 기초하여 소수의 트랜잭션으로 더 분할될 수 있다. 은닉 마르코프 모델과 같은 이벤트 스트림 또는 다수의 트랜잭션에 대한 일부 통계적 분석은 이벤트 스트림이 버스티(bursty)이고, 각 버스트는 트랜잭션으로 필터링될 수 있다는 점을 나타낸다.

[0034] 트랜잭션은 또한 이벤트 필드로 규정될 수 있다. 일 실시예에서는, 동일한 소스 어드레스(예를 들어, 소스 IP 어드레스)로부터 시작하는 이벤트는 트랜잭션으로 규정된다. 따라서, 규정된 각 트랜잭션은 하나의 IP 어드레스의 모든 이벤트를 이벤트 스트림의 시간 프레임 내에서 포함한다.

[0035] 트랜잭션 파라미터는 소스 어드레스 및 목적지 어드레스 양쪽을 포함할 수 있다. 따라서, 일부 트랜잭션은 동일한 소스 및 목적지 IP 어드레스(또는 IP 포트 또는 둘 모두)를 갖는 시간프레임 내의 모든 이벤트로 규정될 수 있다. 또한, 일부 트랜잭션은 관련하는 두 개의 소스/목적지 어드레스(또는 포트) 쌍 사이에 교환되는 이벤트로 규정될 수 있다. 그러한 실시예에서는, 어드레스 A로부터 어드레스 B까지 및 어드레스 B로부터 어드레스 A까지의 보안 이벤트는 하나의 트랜잭션에 포함될 수 있다.

[0036] 일 실시예에서는, 트랜잭션은 실제 보안 이벤트가 아니라 보안 이벤트를 표현하는 항목이다. 하나의 항목으로 다수의 동일한 보안 이벤트를 표현할 수 있다. 이러한 방식에 의해 중요한 데이터가 저장될 수 있다. 일 실시예에서는, 각 항목은 각 보안 이벤트가 몇 번 표현되는 가를 나타낸다. 다른 실시예에서는, 각 보안 이벤트는 트랜잭션에서 단지 계수되지 않는 단일 항목으로 표현된다.

[0037] 일 실시예에서, 트랜잭션의 규정 사항들은 상호 간에 익스클루시브(exclusive)하지 않다. 즉, 상술되고 추가로 하술된 트랜잭션의 규정 사항 및 기술되지 않은 그 밖의 규정 사항들은 동일한 세트의 보안 이벤트에서 구현될 수 있다.

[0038] 일 실시형태에서, 트랜잭션은 복수의 트랜잭션 필터(34)로 규정된다. 도 2의 생략 부호(ellipsis)는 보안 이벤트가 임의의 개수의 트랜잭션으로 필터링될 수 있다는 것을 나타내며, 도 2에 나타낸 4개 트랜잭션 필터(34)로 제한하지 않는다. 트랜잭션 필터(34)는 상술된 임의의 트랜잭션을 생산 가능하도록 구성될 수 있다. 예를 들어, 트랜잭션 필터(34a)는 시간 기반 트랜잭션을 생산할 수 있는 한편, 트랜잭션 필터(34b)는 이벤트 필드 기반 트랜잭션을 생산할 수 있다.

[0039] 일 실시예에서는, 트랜잭션 필터는 또한 보안 이벤트를 항목으로 변환시킨다. 상술한 바와 같이, 일 실시형태에서, 항목은 이벤트 타입의 디스크립션(description)으로 규정된다. 트랜잭션 필터(34)가 트랜잭션을 채울 경

우, 상술한 바와 같이, 중복되는 보안 이벤트가 폐기되거나 항목 수가 증가될 수 있다.

[0040] 일 실시예에서, 트랜잭션은 또한 새로운 트랜잭션을 발생시키도록 "클러스터화"될 수 있다. 예를 들어, 트랜잭션 필터(34)는 두 개의 중복된 트랜잭션을 클러스터화하여 구성될 수 있다. 따라서, 예를 들어, 하나의 트랜잭션이 소스(A)에서 목적지(B)까지 항목을 수집하고, 다른 트랜잭션이 소스(B)에서 목적지(C)까지 항목을 수집하는 경우, 세 번째 트랜잭션은 A에서 B를 거쳐 C까지 가로지르는 항목을 수집할 수 있다. 다른 트랜잭션도 또한 중복할 수 있다. 그와 같이 클러스터화된 트랜잭션은 웜(worm)의 유포 및 목적지 어드레스가 공격용 소스 어드레스가 되는 다른 공격들을 나타내는 트랜잭션의 탐지에 유용할 수 있다.

[0041] 일 실시예에서, 트랜잭션 필터(34)에 의해 생산된 트랜잭션은 패턴 식별자(36)에 제공된다. 패턴 식별자(36)는 트랜잭션을 분석하여 이벤트 스트림에서의 패턴을 찾는다. 패턴 식별자(36)의 일 실시예는 도 3을 참조하여 더욱 상세하게 기술된다.

[0042] 일 실시예에서, 패턴은 일부의 방식으로 함께 발생한, 즉 관련된 보안 이벤트 그룹으로 규정될 수 있다. 예를 들어, 보안 이벤트 A, B, 및 C가 서로 가까이에서 종종 발생한다면, 이벤트 그룹 ABC는 패턴화될 수 있다. 한편, A, B, 및 C가 시간적으로 서로 가까이에서 발생하지 않고, 시간이 지나서도 유사한 빈도로 발생할 경우, 이벤트 그룹 ABC는 또한 패턴이 될 수 있다. 알려진 패턴의 예는 성공한 로그인에 뒤이은 일련의 실패한 로그인이다. 이러한 패턴은 사전단어공격(dictionary attack)이 사용될 경우에 발생한다. 느리면서 저급한 공격은 시간적으로 가까이 있지는 않지만 이벤트 빈도에 관련되는 이벤트를 가질 것이다.

[0043] 패턴 발견 모듈(30)에 제공된 보안 이벤트에서 이벤트 패턴을 식별하기 위해, 일 실시형태에서, 패턴 식별자(36)는 먼저 트랜잭션 필터(34)에 의해서 제공된 트랜잭션으로부터 트리 구축자(38)를 사용하여 트랜잭션 트리를 구축한다. 일 실시예에서 트리 구축자(38)는 각 트랜잭션에서 항목 서포트(item support)에 의해 항목을 먼저 배열(order)한다.

[0044] 일 실시예에서, 이는 각 항목이 발생한 트랜잭션의 수를 계수함으로써 이루어진다. 이 수는 종종 항목의 "서포트"라고 한다. 일 실시예에서, 항목은 트랜잭션에 머물기 위해 최소수의 서포트를 필요로 하며, 그렇지 않은 경우 항목은 트랜잭션의 크기를 감소시키기 위해 삭제된다.

[0045] 일 실시예에서, 트리 구축자(38)는 트랜잭션을 각 보안 이벤트의 서포트에 기반된 트리 구조로 편성한다. 트랜잭션 트리를 구축하기 위한 하나의 공정은 도 4를 참조하여 기술된다. 도 4에 도시된 공정은 동시에 하나의 트랜잭션을 반복적으로 동작시킨다. 일 실시예에서, 트랜잭션은 서포트에 의해 최상으로 서포트된 항목으로부터 최하로 서포트된 항목까지 편성된다.

[0046] 블록(402)에서는, 트랜잭션이 선택된다. 일 실시예에서, 트리는 빈 루트 노드(empty root node)를 갖는다. 그 밖의 모든 노드는 항목을 나타낸다. 블록(404)에서는, 루트 노드가 선택되고, 블록(406)에서는, 선택된 트랜잭션에서 초기 반복의 제 1(즉, 가장 높은 서포트) 항목인 다음의 항목이 선택된다.

[0047] 블록(408)에서, 선택된 노드(초기 반복에서 루트 노드)의 임의의 차일드가 선택된 항목과 동일한가의 여부로 결정이 이루어진다. 대답이 '예'이면, 이 노드는 블록(410)에서 선택된 노드가 되고, 선택된 항목의 서포트가 노드의 현 서포트에 부가된다. 하지만, 대답이 '아니오'라면, 선택된 노드의 새로운 차일드 노드는 블록(414)에서 선택된 항목용으로 생성되며, 이 새로운 노드는 블록(416)에서 선택된 노드가 된다.

[0048] 다음, 블록(412)에서 결정은 선택된 트랜잭션이 완전히 처리되었는가, 즉 그 이상의 항목을 포함하는가의 여부로 이루어진다. 트랜잭션이 완료되지 않는 경우(트랜잭션에서 항목이 더 있음), 선택된 트랜잭션의 처리는 서포트 순으로 다음 항목의 선택에 의해 블록(406)에서 계속된다. 그러나, 선택된 트랜잭션이 완전하게 처리(추가 항목이 없음) 되는 경우, 처리될 다음 트랜잭션이 블록(402)에서 선택되고, 모든 트랜잭션이 트리 구축 처리에서 소모될 때까지 도 4의 순서도에 따라 처리를 지속한다.

[0049] 다시 도 3을 참조하면, 일 실시예에서, 트리 구축자(38)는 트랜잭션 트리로부터 관심 패턴을 추출하는 트리 파서(40)에 트랜잭션 트리를 제공한다. 일 실시예에서, 트리 파서(40)는 서포트(support)의 주목할만한 하락(drop)을 찾도록 트리 브랜치를 가로지른다(traverse). 예를 들면, 항목 명칭(서포트)으로 나타낸 트리의 하나의 브랜치가 A(200), B(190), C(50), D(45)인 경우, 항목(B)과 항목(C) 사이의 서포트의 주목할만한 하락이 있다. 이 경우, 트리 파서(40)는 패턴으로서 그룹 AB 및 패턴으로서 그룹 CD를 식별할 수 있다. 트리 파서(40)는 마찬가지로 패턴으로서 ABCD를 또한 식별할 수 있다. 일 실시예에서, 패턴인 ABCD는 이들 보안 이벤트의 시간적 순서가 패턴 내에 보존된다는 것을 의미하지 않는다.

[0050] 일 실시예에서는, 서포트의 하락은 감소의 비율로 규정된다. 예를 들어, 트리 파서(40)는 브랜치에서 한 항목으로부터 다음 항목까지 30 퍼센트 이상 서포트가 하락할 때는 항상 서포트의 하락으로 규정한다. 다른 비율도 또한 사용될 수 있다. 다른 실시예에서는, 서포트의 하락은 10과 같이 특정한 수에 의한 서포트의 하락으로 규정된다.

[0051] 또, 다른 실시예에서는, 서포트의 하락은 트리의 브랜치에서 다음 항목의 부가가 지금까지 고려된 서포트의 표준 편차를 임계값(threshold number) 이상이 되는 경우에 탐지된다. 예를 들어, 제 1 트리 브랜치 항목의 서포트의 표준 편차는 0.03 이하이지만, 제 4 항목의 부가에 의해 표준 편차가 0.03 이상이 되는 경우, 트리 파서(40)는 브랜치에서 제 3 및 제 4 항목 사이에서 서포트에서의 하락을 탐지한다. 수 0.03은 단지 예일 뿐이며, 다른 임계값이 사용될 수 있다. 게다가, 분산과 같은 변화에 대한 다른 통계적인 척도가 표준 편차를 대신 할 수 있다.

[0052] 일 실시예에서는, 트리 파서(40)는 트랜잭션 트리를 가로지르면서, 트리 내에 관련된 요인을 또한 찾는다. 예를 들어, 이벤트 그룹이 트랜잭션 트리의 일부분보다 많은 부분에서 발생하는 경우, 일 실시예에서, 트리 파서(40)는 이벤트 그룹이 패턴임을 선언한다. 일 실시예에서는, 이는 트리에서의 모든 발생을 새롭게 식별된 패턴의 표현으로 대신하고, 그 대신한 결과로 생성된 트리에서 패턴 탐지를 수행함으로써 행한다.

[0053] 일 실시예에서는, 패턴 식별자(36)는 발견된 패턴을 패턴 발견 모듈(30)에 제공한다. 일 실시예에서는, 패턴 발견 모듈(30)은 새롭게 발견된 패턴을 콘솔 인터페이스(16)를 통해 유저에게 표시한다. 유저는 그 목적에 따라 다수의 방식으로 패턴을 사용할 수 있다. 예를 들어, 유저는 이벤트 패턴을 조사하여 그것이 악성 공격이나 양성 실행을 나타내는가의 여부를 결정할 수 있다.

[0054] 일 실시예에서는, 유저 인터페이스는 규칙 발생 툴을 유저에게 제공하고, 그 규칙 발생 툴은 유저가 발견된 임의의 패턴을 규칙 엔진(18)에 의해 사용되는 상관 규칙으로 직접 변환시키는 것이 가능하다. 다음, 규칙 엔진(18)은 제공되는 모든 금후(또는 재연된) 이벤트 스트림에서 패턴의 특정 인스턴스를 탐지하고 찾아낸다. 이 방식으로, 새롭게 발견된 패턴은 "이전에 알려진" 패턴이 될 수 있다.

[0055] 일 실시예에서는, 변환은 단일-클릭 인터페이스를 사용하여 제공될 수 있다. 예를 들어, 각 발견된 패턴의 옆에서, 콘솔 인터페이스(16)는 규칙 엔진(18)에서 상관 규칙으로서 패턴을 구현하도록 기능하는 "규칙으로 변환"이라 라벨된 버튼을 제공할 수 있다.

[0056] 상술한 설명에서는, 패턴 발견 모듈(30)은 보안 이벤트의 취득이 입력으로, 보안 이벤트에서의 패턴 생산이 출력으로 기술된다. 하지만, 패턴 발견 모듈은 패턴 내에서 패턴을 탐지하도록 또한 구성될 수 있다. 그러한 실시예에서는, 패턴 발견 모듈은 각각의 단일 패턴을 단일 요소로서 다루며, 이벤트의 패턴을 찾는 것과 유사한 방식으로 패턴 발견을 수행한다.

[0057] 이와 같이, 패턴 발견 가능성을 갖는 네트워크 보안 시스템이 기술되었다. 상술한 설명에서, 다양한 특정한 값 및 데이터 구조가 "보안 이벤트", 트랜잭션", 트랜잭션 트리", 및 "항목" 등의 명칭으로 주어졌다. 게다가, "패턴 발견 모듈" 및 "패턴 식별자"와 같은 다양한 특정 모듈이 기술되었다. 하지만, 이들 명칭은 본 발명의 다양한 면을 단지 기술 및 도시할 뿐이고, 본 발명의 범위를 한정하지 않는다. 또한, 도 3의 트리 구축자(38) 및 트리 파서(40)와 같은 다양한 모듈은 소프트웨어나 하드웨어 모듈 또는 그들 기능을 모듈로 분할하지 않고도 구현될 수 있다. 상술여부에 관계없이, 본 발명은 소프트웨어 또는 하드웨어 중 어느 하나에서 임의의 모듈 구조로 한정되지 않는다.

도면의 간단한 설명

[0009] 본 발명은 첨부한 도면을 예로써 도시하지만, 그 도면에 의해 한정되지는 않는다.

[0010] 도 1은 본 발명의 일 실시예에 따른 네트워크 보안 시스템의 블록도.

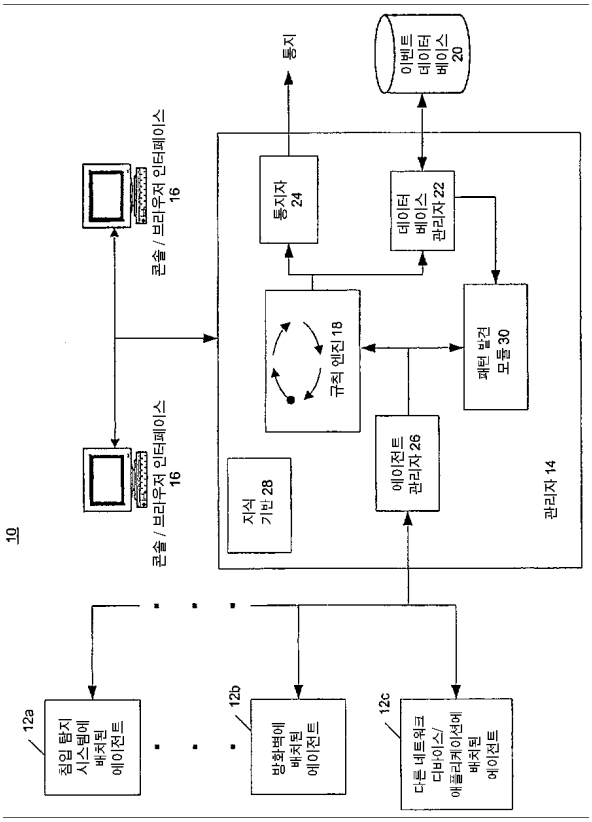
[0011] 도 2는 본 발명의 일 실시예에 따른 패턴 발견 모듈의 블록도.

[0012] 도 3은 본 발명의 일 실시예에 따른 패턴 식별자의 블록도.

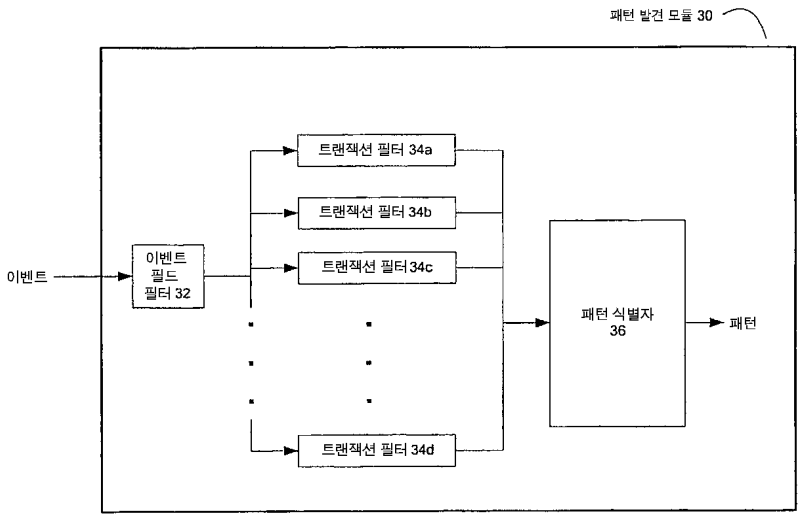
[0013] 도 4는 본 발명의 일 실시예에 따른 트랜잭션 트리 구축용 프로세스를 도시하는 플로 차트.

도면

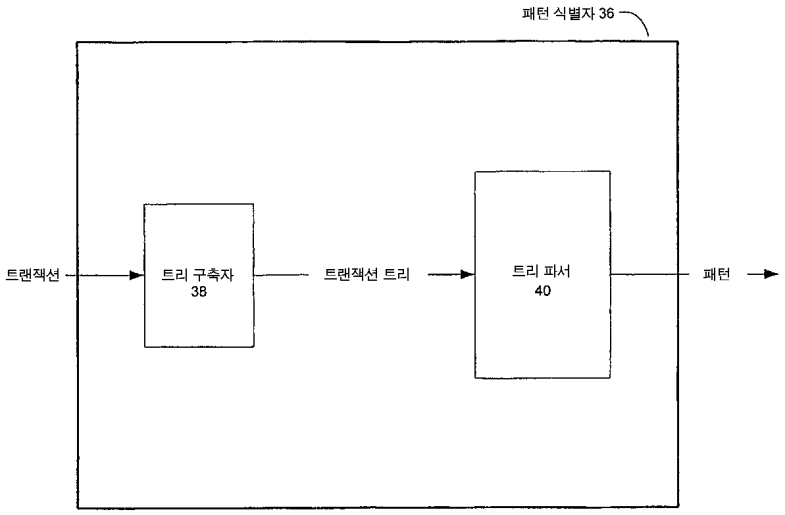
도면1


10
콘솔 / 브라우저 인터페이스 16
콘솔 / 브라우저 인터페이스 16
통지
이벤트 데이터 베이스 20
통지자 24
데이터 베이스 관리자 22
규칙 엔진 18
패턴 발견 모듈 30
에이전트 관리자 26
지식 기반 28
관리자 14
12a
침입 탐지 시스템에 배치된 에이전트
12b
방화벽에 배치된 에이전트
12c
다른 네트워크 디바이스/애플리케이션에 배치된 에이전트

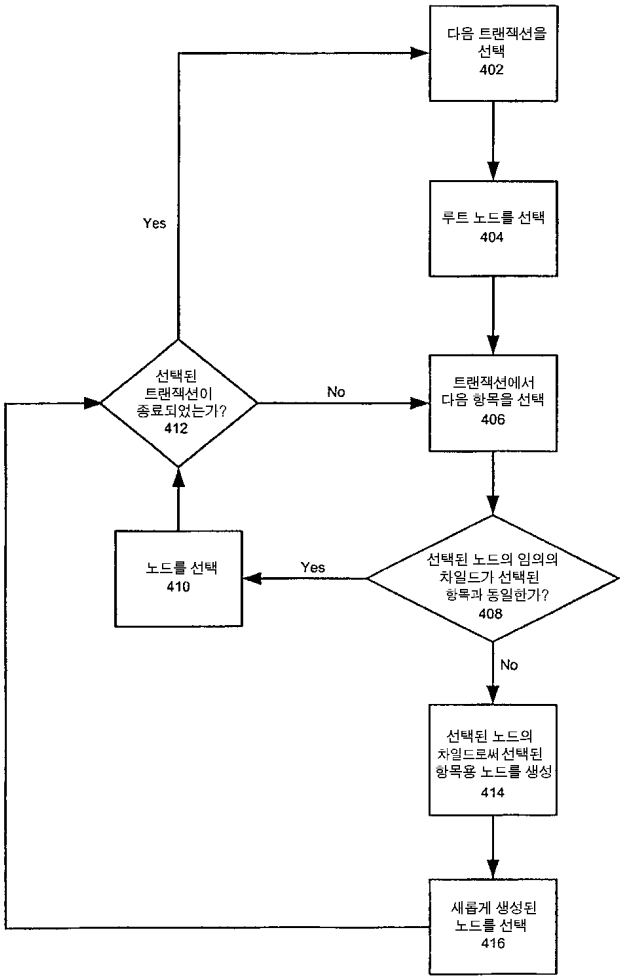
도면2


패턴 발견 모듈 30
이벤트
이벤트 필드 필터 32
트랜잭션 필터 34a
트랜잭션 필터 34b
트랜잭션 필터 34c
트랜잭션 필터 34d
패턴 식별자 36
패턴

도면3


패턴 식별자 36
트랜잭션
트리 구축자
38
트랜잭션 트리
트리 파서
40
패턴

도면4


다음 트랜잭션을 선택
402
루트 노드를 선택
404
트랜잭션에서 다음 항목을 선택
406
선택된 노드의 임의의 차일드가 선택된 항목과 동일한가?
408
No
선택된 노드의 차일드로써 선택된 항목용 노드를 생성
414
새롭게 생성된 노드를 선택
416
Yes
노드를 선택
410
선택된 트랜잭션이 종료되었는가?
412
No
Yes