

명세서

발명의 명칭: 하드웨어 기반의 커널 코드 삽입 공격 탐지 장치 및 그 방법

기술분야

- [1] 본 발명은 커널 코드에 대한 악성 코드 삽입 시도를 탐지하는 장치 및 그 방법에 관한 것으로서, 보다 상세하게는 공격 탐지 대상 시스템에 독립적인 하드웨어로 구현되어 대상 시스템의 커널 코드에 대한 쓰기 시도나 시스템이 커널 모드로 동작하는 동안에 발생하는 커널 코드 밖으로부터의 실행시도를 감지할 수 있는 장치 및 그 방법에 관한 것이다.

배경기술

- [2] 운영 체제 커널의 무결성을 보호하기 위하여 많은 보안 전문가들은 보안 모니터를 호스트 시스템에 대해 독립적으로 만들기 위해 노력하고 있다. 커널 무결성 모니터링에 대한 노력은 크게 두 가지 접근 방식으로 분류할 수 있다. 하나는 하드웨어 기반의 접근 방식이며 나머지 하나는 하이퍼바이저(hypervisor) 기반의 접근 방식이다.
- [3] 하이퍼바이저 기반의 방식이 최근 인기를 얻고 있으나, 하이퍼바이저가 복잡해질수록 더욱 많은 소프트웨어 취약점을 가질 수 있다. 하이퍼바이저의 무결성을 보호하기 위해 추가적인 소프트웨어 레이어가 삽입될 수 있으나, 추가적인 소프트웨어 레이어 또한 새로운 취약점들을 발생시킬 수 있으므로 근본적인 해결책을 제공하지는 못한다.
- [4] 커널 무결성 모니터링(kernel integrity monitoring)을 위한 기존 솔루션의 대부분은 커널이 사용하는 메모리에 대한 악의적인 쓰기를 탐지하는 방향으로 발전하여 왔다. 그러나 이에 못지않게 중요한 것은 커널에 대한 악성 코드 삽입 공격을 막는 것이다. 커널에 대한 많은 공격들이 공격자가 설계한 코드가 커널의 일부로 동작되도록 하는 방법으로 만들어지는데, 이와 같은 공격을 실현하는 가장 쉬운 방법이 코드 삽입이기 때문이다. 코드 삽입을 탐지하기 위한 방법으로 하이퍼바이저 기반의 접근 방법이나, 특정 프로세서가 제공하는 프로세서의 특수기능을 이용한 코드삽입탐지기술에 대한 연구들이 진행되고 있다. 그러나 이러한 접근 방법은 여전히 성능 문제를 크게 일으키거나, 앞서 언급한 보안 문제점이 있다.

발명의 상세한 설명

기술적 과제

- [5] 본 발명이 이루고자 하는 기술적 과제는, 커널 코드 영역에 대한 쓰기 시도와 시스템이 커널모드로 동작하는 동안 발생하는 커널 코드 영역 밖으로부터 실행시도를 감지하여 커널 코드에 대한 악성 코드 삽입 시도를 탐지해 낼 수 있는 하드웨어 기반의 공격 탐지 장치 및 그 방법을 제공하는 데 있다.

과제 해결 수단

- [6] 상기의 기술적 과제를 달성하기 위한, 본 발명에 따른 공격 탐지 장치의 일 예는, 호스트 시스템의 프로그램 트레이스 인터페이스(PTI)로부터 런타임 정보를 포함하는 패킷을 수신하고, 상기 패킷을 파싱하여 간접 분기 목적지 주소를 파악하는 트레이스 관리부; 호스트 시스템의 부트시 커널의 코드 블록들 중 주소변환 베이스 레지스터(ATBR) 변경, 모드 변환 또는 PTI 제어 레지스터 변경을 포함하는 이벤트와 관련 있는 코드 블록을 트램폴린에 대한 점프 명령으로 대체한 상태에서, 상기 이벤트 발생시 상기 트램폴린의 수행을 통해 이벤트를 보고받는 보고 관리부; 및 버스의 트래픽을 스누핑하여 메모리의 물리적 코드 영역에 대한 쓰기 시도를 검출하는 트래픽 관리부;를 포함한다.
- [7] 상기의 기술적 과제를 달성하기 위한, 본 발명에 따른 공격 탐지 방법의 일 예는, 호스트 시스템의 프로그램 트레이스 인터페이스(PTI)로부터 런타임 정보를 포함하는 패킷을 수신하고, 상기 패킷을 파싱하여 간접 분기 목적지 주소를 파악하는 단계; 호스트 시스템의 부트시 커널의 코드 블록들 중 주소변환 베이스 레지스터(ATBR) 변경, 모드 변환 또는 PTI 제어 레지스터 변경을 포함하는 이벤트와 관련 있는 코드 블록을 트램폴린에 대한 점프 명령으로 대체한 상태에서, 상기 이벤트 발생시 상기 트램폴린의 수행을 통해 이벤트를 보고받는 단계; 및 버스의 트래픽을 스누핑하여 메모리의 물리적 코드 영역에 대한 쓰기 시도를 검출하는 단계;를 포함한다.

발명의 효과

- [8] 본 발명에 따르면, 타겟 시스템의 운영체제 커널보다 높은 권한을 가지는 소프트웨어 계층의 도움을 받지 않아도 되고, 기 존재하는 하드웨어 아키텍처를 새롭게 디자인하지 않고서도 커널 공격을 탐지할 수 있다.

도면의 간단한 설명

- [9] 도 1은 본 발명에 따른 하드웨어 기반의 커널 코드 삽입 공격 탐지 장치의 일 실시 예의 구성을 도시한 도면,
- [10] 도 2는 본 발명에 따른 모드 변환 추적 방법의 일 예를 도시한 도면,
- [11] 도 3은 본 발명에 따른 ATBR 변경 등의 이벤트를 보고하는 방법의 일 예를 도시한 도면,
- [12] 도 4는 본 발명에 따른 트램폴린의 일 예를 도시한 도면,
- [13] 도 5는 본 발명에 따른 하드웨어 기반의 커널 코드 삽입 공격 탐지 장치를 이용하여 재생 공격을 방어하는 방법의 일 예를 도시한 도면,
- [14] 도 6은 본 발명에 따른 하드웨어 기반의 커널 코드 삽입 공격 탐지 장치의 각 구성의 상호 의존성의 일 예를 도시한 도면,
- [15] 도 7은 본 발명에 따른 이벤트 보고 방법의 다른 실시 예를 도시한 도면,
- [16] 도 8은 본 발명에 따른 하드웨어 기반의 커널 코드 삽입 공격 탐지 장치의 다른 실시 예를 도시한 도면, 그리고,

- [17] 도 9는 본 발명에 따른 하드웨어 기반의 커널 코드 삽입 공격 탐지 방법의 일 실시 예의 흐름을 도시한 도면이다.

발명의 실시를 위한 형태

- [18] 이하에서, 첨부된 도면들을 참조하여 본 발명에 따른 하드웨어 기반의 커널 코드 삽입 공격 탐지 장치 및 그 방법에 대해 상세히 설명한다.

- [19] 도 1은 본 발명에 따른 하드웨어 기반의 커널 코드 삽입 공격 탐지 장치의 일 실시 예의 구성을 도시한 도면이다.

- [20] 본 실시 예에서, 하드웨어 기반의 커널 코드 삽입 공격 탐지 장치(이하, '공격 탐지 장치'라 함)(100)는 호스트 시스템(135)의 주소변환 베이스 레지스터(ATBR, Address Translation Base Register)의 올바른 값들을 미리 알고 있다고 가정한다. ATBR은 제1 레벨 페이지 테이블(또는 페이지 글로벌 디렉토리)의 기본 주소를 포함한다. 호스트 시스템(135)은 프로세서(140), 메모리(150) 및 주변기기(160) 등을 포함하는 일종의 컴퓨팅 장치로서 공격자가 커널 코드를 삽입하여 공격하려는 대상이다.

- [21] 또한 호스트 시스템(135)의 커널 페이지 테이블들은 초기 부트(boot)시 가상 코드 영역과 물리 코드 영역을 올바르게 맵핑하고 있다고 가정한다. 여기서, 가상 코드 영역은 커널 코드를 저장하고 있는 가상 메모리 영역을 의미하고, 물리 코드 영역은 커널 코드를 저장하고 있는 물리 메모리 영역을 의미한다.

- [22] 또한 공격 탐지 장치(100)는 호스트 시스템(135)이 부팅될 때, 특권 모드(privileged-mode)와 사용자 모드(user-mode) 사이의 모드 변환을 일으키는 커널의 코드 블록, 사용자 모드에서 특권 모드로 진입할 때 프로세서에 의해 수행되는 코드 블록(즉, 게이트웨이 코드 블록), ATBR 변경과 관련된 코드 블록, 게이트웨이 코드 블록의 주소를 변경하는 코드 블록, 프로그램 트레이스 인터페이스(PTI, Program Trace Interface)를 재구성하는 코드 블록 등의 위치를 파악한다. 공격 탐지 장치(100)는 파악한 각 코드 블록을 적당한 코드 스니펫(snippet)(이하, '트램폴린(trampolines)'이라 함)에 대한 분기(jump) 명령으로 대체한다. 여기서 각 코드블록에 대한 트램폴린은 대응되는 원래의 코드 블록을 수행하고 원래의 코드 블록의 수행에 따른 모드 변환이나 ATBR 주소 변경, PTI 제어 레지스터의 변경 등의 이벤트를 공격 탐지 장치(100)에게 보고하는 기능을 포함한다. 이벤트를 보고하는 트램폴린의 일 예는 도 4에 도시되어 있다.

- [23] 각 코드 블록의 위치는 종래의 다양한 방법을 통해 파악될 수 있다. 예를 들어, 호스트 시스템(135)이 사용자 모드에서 특권 모드로 진입할 때, 프로세서(140)는 게이트웨이 코드 블록을 가리키는 지정 주소(designated address)를 파악한다. 대부분의 프로세서는 지정 주소를 계산하기 위하여 특별 레지스터(special register)를 포함한다. 따라서 공격 탐지 장치는 호스트 시스템(135)의 부트시, 특별 레지스터의 값들을 이용하여 게이트웨이 코드 블록의 위치를 파악할 수

있다.

- [24] 도 1을 참조하면, 공격 탐지 장치(100)는 호스트 시스템(135)의 각 하드웨어와 연결되어 있으며, 트레이스 관리부(110), 보고 관리부(120), 트래픽 관리부(130)를 포함한다.
- [25] 트레이스 관리부(110)는 프로세서(140)의 PTI(142)와 연결되며, PTI(142)로부터 다양한 런타임 정보를 포함하는 데이터 패킷을 수신한다. 여기서 PTI(142)는 디버깅 등의 목적을 위하여 프로세서에 구현된 외부 인터페이스로서, 디버깅 인터페이스라고도 한다. PTI(142)는 프로세서의 종류에 따라 다양한 이름으로 불려질 수 있으며, 일 예로 '인텔 x86 프로세서'에서는 '프로세서 트레이스(processor tracer)'로 불리며, 'ARM 프로세서'의 경우 'Program Trace Macrocell(PTM)'로 불린다. 따라서 본 실시 예에서 PTI(142)는 프로세서 내 다양한 정보를 외부로 출력할 수 있는 모든 종류의 인터페이스를 나타내는 것으로 정의한다.
- [26] 트레이스 관리부(110)는 악의적인 삽입 코드가 특권 모드(privileged-mode)에서 수행되는지를 파악하기 위하여, 수신한 패킷을 파싱하여 간접 분기 목적지 주소(indirect branch target addressess)를 파악한다. 아래에서 살펴보겠지만, 보고 관리부(120)는 ATBR 변경이나 모드 변환 등의 이벤트가 발생하는 경우 트랩폴린을 통해 ATBR 변경이나 모드 변환 등의 이벤트를 보고받는다. 따라서 트레이스 관리부(110)는 PTI(142)를 통해 파악한 간접 분기 목적지 주소와 보고 관리부(120)를 통해 파악된 ATBR 변경이나 모드 변환 등의 이벤트를 기초로, 공격자가 가상 코드 영역 밖에서 프로세서(140)를 강제로 특권 모드로 변환하려는 시도를 검출할 수 있다.
- [27] 예를 들어, 트레이스 관리부(110)는 간접 분기 목적지 주소 리스트와 리스트 내 각 주소가 어떤 모드가 관련되어 있는지 호스트 시스템(135)의 초기 부팅시 알고 있다고 할 때, PTI를 통해 파악된 간접 분기 목적지 주소와 관련된 모드가 올바른지 보고 관리부(120)를 통해 보고받은 모드를 통해 파악하여 공격 여부를 탐지하고 알람을 발생한다.
- [28] 보고 관리부(120)는 호스트 시스템(135)의 주변기기 버스(180)와 연결되며, 키잉(keying) 레지스터(122), 보고(notification) 레지스터(124), Ack 레지스터(126) 등을 포함한다. 보고 관리부(120)는 주변기기 버스와 두 개의 연결 관계(192,194)를 가질 수 있다. 제1 연결관계(194)를 통해, 프로세서(140)는 트랩폴린이 수행되는 동안 ATBR 변경이나 모드 변환, PTI 제어 레지스터 변경 등의 이벤트를 보고 관리부(120)에 보고한다. 제2 연결관계(192)를 통해, 보고 관리부(120)는 호스트 시스템(135)의 메인 메모리(150)에 접근하고, 프로세서(140)의 보고를 검증하기 위하여 메모리(150)의 콘텐츠를 분석한다.
- [29] 보다 구체적으로, 프로세서(140)가 커널 명령을 수행하다가 트랩폴린에 대한 분기(jump) 명령을 만나면 트랩폴린으로 분기하며, 트랩폴린 수행을 통해 모드 변환이나 ATBR의 주소 변경, PTI 제어 레지스터 변경 등의 이벤트를 보고

관리부(120)에게 보고한다. 보고 관리부(120)가 각 레지스터(122,124,126)를 통해 프로세서(140)로부터 이벤트를 보고받는 방법에 다양한 실시 예는 도 3 및 도 7에 도시되어 있다. 예를 들어, ATBR의 주소 변경 등은 도 3의 프로토콜을 이용하여 보고하고, 모드 변환 등은 도 7의 프로토콜을 이용하여 보고 할 수 있다.

- [30] 트래픽 관리부(130)는 호스트 시스템(140)의 메모리 버스(170) 및 주변기기 버스(180)와 연결(196,198,199)되며, 각 버스(170,180)의 트래픽을 분석하여 메인 메모리(150)에 대한 쓰기(write operation), PTI(142)의 제어 레지스터에 대한 쓰기 동작 등을 감시한다. 보다 구체적으로, 트래픽 관리부(130)는 메인 메모리(150)와 연결된 각 버스(170,180)의 트래픽을 스누핑(snooping)하여 프로세서(140) 또는 주변 기기(160)에 의한 메인 메모리(150)의 쓰기 동작을 파악함으로써 ATBR이 가리키고 있는 제1 레벨 페이지 테이블(페이지 글로벌 디렉토리) 등을 파악하거나, 주변기기 버스(180)의 트래픽을 스누핑하여 메모리 맵 제어 레지스터(memory-mapped control register)를 통해 PTI(142)를 제어하려는 모든 접근을 파악한다.
- [31] 예를 들어, 공격자가 물리 코드 영역에 코드를 삽입하려고 시도한다고 가정하자. 트래픽 관리부(130)는 주변기기 버스(180)와의 연결 관계(198,199)를 통해 메인 메모리(150)에 대한 모든 쓰기 트래픽을 스누핑(snooping)하고, 물리 코드 영역에 쓰기 시도가 있으면 알람을 생성한다. 트래픽 관리부(130)는 프로세서(140)나 주변기기(160)로부터의 쓰기 트래픽을 계속하여 스누핑하므로, 직접 메모리 접근(Direct Memory Access)를 통한 쓰기 시도 등을 검출할 수 있다.
- [32] 또 다른 예로, 트래픽 관리부(130)는 커널 불변 영역의 주소를 미리 가지고 있으며, 버스(170,180)의 트래픽을 스누핑하여 커널 불변 영역에 대한 쓰기 시도를 감지하면 알람을 생성할 수 있다. 여기서 커널 불변 영역이라고 함은 커널 코드 영역, 시스템 콜 테이블, 인터럽트 설명자 테이블(IDT, Interrupt Descriptor table) 등을 가리킨다. 커널 불변 영역에 대한 주소는 호스트 시스템(135)의 부트 시에 다양한 방법을 통해 파악될 수 있다.
- [33] 또 다른 예로, 보고관리부(120)에 의해 ATBR 변경이 파악된 경우에, 트래픽 관리부(130)는 전체 ATBR의 페이지 테이블에 대한 패킷을 스누핑하는 것이 아니라, 변경된 ATBR을 기초로 현재 사용 중인 제1 레벨 페이지 테이블(또는 페이지 글로벌 디렉토리)을 파악하고 파악된 부분에 대한 패킷을 스누핑하여 감시한다.
- [34] 도 2는 본 발명에 따른 모드 변환을 추적하는 방법의 일 예를 도시한 도면이다.
- [35] 도 1 및 도 2를 함께 참조하면, 프로세서(140)의 PTI(142)는 직접 또는 간접 분기 목적지 주소를 순차적으로 트레이스 관리부(110)에 제공하고 있으며, 트레이스 관리부(110)는 PTI(142)로부터 각 분기 목적지 주소 정보를 수신하여 파악할 수 있다. 또한 모드 변환시, 프로세서(140)는 트램폴린으로 분기되어 트램폴린을 수행하므로, 보고 관리부(120)는 프로세서(140)로부터 모드 변환을 보고받을 수 있다.

- [36] 예를 들어, 시간 (a)(200)의 분기 목적지 주소(200)와 연관된 모드는 사용자 모드이고, 그 다음 분기 목적지 주소와 연관된 모드는 특권 모드로서 모드 변환이 발생하므로, 프로세서(140)는 특권 모드에서 트램폴린의 수행을 통해 모드가 사용자 모드에서 특권 모드로 변환되었음을 보고 관리부(120)에 보고한다.
- [37] 또 다른 예로, 시간 (b)의 분기 목적지 주소(210)와 연관된 모드는 특권 모드이고, 그 다음 분기 목적지 주소와 연관된 모드는 사용자 모드로서 모드 변환이 발생하므로, 프로세서(140)는 특권 모드에서 트램폴린의 수행을 통해 모드가 특권 모드에서 사용자 모드로 변환되었음을 보고 관리부(120)에 보고한다.
- [38] 도 3은 본 발명에 따른 ATBR 변경 등의 이벤트를 보고하는 방법의 일 예를 도시한 도면이다.
- [39] 도 1 및 도 3을 함께 참조하면, 호스트 시스템(135)은 ATBR의 변경이나 모드 변환 등이 발생함에 따라 트램폴린으로 분기하면, PTI를 통해 트램폴린 시작을 공격 탐지 장치(100)에게 알려준다(S320). 공격 탐지 장치(100)는 PTI(142)를 통해 호스트 시스템(135)으로부터 트램폴린 시작을 인지하면, 키잉 레지스터(122)에 임의의 키(0이 아닌 값으로 구성)를 생성하여 저장하고, 호스트 시스템(135)은 트램폴린의 수행을 통해 키잉 레지스터(122)를 읽어 키를 획득한다(S310). 키잉 레지스터(122)는 2회 이상 키를 읽어가는 것을 방지하기 위하여 호스트 시스템이 키잉 레지스터(122)를 한 번 읽으면, 리셋한다.
- [40] 호스트 시스템(135)은 키잉 레지스터(122)로부터 키를 얻으면, 공격 탐지 장치(100)의 보고 레지스터(124)에 이벤트 내용(예를 들어, 모드 변환, ATBR 변환, PTI 제어 레지스터 변경 등)을 기록한다(S330). 공격 탐지 장치는 보고 레지스터(124)의 내용을 읽어 호스트 시스템(135)에서 발생한 이벤트를 확인할 수 있다.
- [41] 공격자의 우회(bypass) 공격이나 재생(replay) 공격을 차단하기 위하여, 호스트시스템(135)은 이벤트 내용과 함께 이전 획득한 키를 함께 전송할 수 있다(S320). 즉 호스트 시스템(135)은 키와 이벤트 내용을 순차적으로 공격 탐지 장치(100)에 전송한다(S320, S330).
- [42] 공격 탐지 장치(100)는 보고 레지스터(124)에 새롭게 저장된 보고 내용을 읽은 후, 검증이 필요한 경우 메인 메모리(150) 등에 접근하여 보고 내용을 검증한다. 예를 들어, ATBR 변경 보고인 경우에, 공격 탐지 장치(100)는 새로운 ATBR이 가리키고 있는 메모리 내용으로부터 해당 변경의 악성 여부를 판단한다. PTI 제어 레지스터의 변경 보고인 경우에, 공격 탐지 장치(100)는 메인 메모리에 저장된 PTI 제어 레지스터를 읽고, 호스트 시스템의 부트시 확보한 PTI 제어 레지스터와 비교하여 변경 여부를 검증한다. 모드 변환 보고인 경우에는 별도의 검증이 필요하지 않을 수 있다. 보고 내용이 검증된 후, 공격 탐지 장치(100)는 Ack 레지스터(126)에 확인 값을 기록하고, 호스트 시스템(135)이 Ack

레지스터(126)를 읽어 확인 값을 확인하면(S340), 이벤트 보고를 위한 세션이 종료된다.

[43] 도 4는 본 발명에 따른 주소 변경 이벤트를 보고하기 위한 트램폴린의 일 예를 도시한 도면이다.

[44] 도 4를 참조하면, ARM 시스템의 VBAR(vector base address register)과 같은 특별 레지스터를 변경하는 이벤트를 공격 탐지 장치에게 보고하기 위한 트램폴린의 일 예가 도시되어 있다. VBAR의 값은 ARM 시스템에서 게이트웨이 코드 블록의 위치를 파악하는데 사용된다.

[45] 트램폴린은 먼저 예외상황(exception)이나 인터럽트(interrupt)를 비활성화상태로 변경한다(라인 2). 트램폴린은 간접 분기(indirect branch) 명령을 포함하지 않으며, 대신에 공격 탐지 장치로부터 키와 확인(ack)을 기다리기 위한 내부 루프를 구현하기 위하여 두 개의 직접 분기(direct branche)를 포함한다. 따라서 트램폴린의 10~11 라인의 명령어 사이에 어떤 다른 명령어가 실행되는 것을 막을 수 있다.

[46] 공격 탐지 장치의 트레이스 관리부(110)는 PTI(142)로부터 수신한 분기 목적지 주소(jump target address)(즉, 11라인)와 공격 탐지 장치(100)가 시작할 때 생성된 올바른 분기 목적지 주소(즉, 트램폴린의 시작 주소, 라인 1)들의 리스트를 비교함으로써 공격을 알아차릴 수 있다.

[47] 도 5는 본 발명에 따른 공격 탐지 장치를 이용하여 재생 공격을 방어하는 방법의 일 예를 도시한 도면이다.

[48] 도 1 및 도 5를 참조하면, 프로세서(140)가 트램폴린으로 점핑하여 공격 탐지 장치(100)에게 모드 변환 등의 이벤트 보고를 시작하려고 할 때, 공격자는 가짜 이벤트 보고(S500)를 트램폴린 시작(S510)과 진짜 이벤트 보고(S540) 사이에 공격 탐지 장치(100)로 전송하려고 시도할 수 있다.

[49] 이 경우, 공격 탐지 장치(100)는 트램폴린 시작 보고(S510) 후 이벤트 보고가 키를 수반하였는지 여부를 기초로 진짜 이벤트 보고(S540)와 가짜 이벤트 보고(S500)를 구분할 수 있다. 진짜 보고 메시지가 도달하고 확인 단계(S550)에 의해 세션이 종료되면 공격자는 가짜 보고 메시지를 보고 레지스터(124)에 기록할 수 없다.

[50] 도 6은 본 발명에 따른 공격 탐지 장치의 각 구성의 상호 의존성의 일 예를 도시한 도면이다.

[51] 도 6을 참조하면, 공격 탐지 장치의 대부분의 구성은 다른 구성 요소 또는 호스트 시스템의 상태 등에 의존한다. 그러나 두 개의 구성 요소, 즉 쓰기 모니터링(600)과 보안 부트(secure boot)(610)는 의존성이 없다.

[52] 점선으로 표시된 부분과 같이, 보안 부트(610)는 호스트 시스템이 초기에 삽입-프리(injection-free), ATRA(Address Translation Redirection Attack)-프리 상태로 구현되는 것을 보장한다. 실선은 공격자가 공격 탐지 장치의 특정 구성이나 상태를 무효화시키려면 극복하여야 하는 부분을 나타낸다. 예를 들어,

공격자가 ATBR이나 커널의 페이지 테이블을 변경하여 ATRA를 성공적으로 수행하지 않는 한, 대상 시스템은 ATRA-프리 상태이다. 다시 말해, 공격자는 ATBR을 변경하기 위하여 ATBR 트래킹을 무효화시키거나 페이지 테이블 콘텐츠를 변경하기 위하여 쓰기 모니터링을 회피하여야 하므로, 공격 탐지 장치는 공격자의 공격을 용이하게 탐지할 수 있다.

[53] 도 7은 본 발명에 따른 모드 변환 등의 이벤트를 보고하는 방법의 다른 실시 예를 도시한 도면이다.

[54] 도 7을 참조하면, 도 3과 같이 호스트 시스템(135)은 모드 변환 등에 따라 트램폴린으로 분기하면 트램폴린 시작을 공격 탐지 장치(100)에게 알리면서(S710) 이벤트 보고 과정을 시작한다. 그러나 호스트 시스템(135)은 트램폴린의 수행 동안 키를 획득하는 것이 아니라 이전 트램폴린의 수행 과정에서 획득한 키(S700)를 이용하여 이벤트를 보고한다(S720, S730). 다시 말해, 호스트 시스템(135)은 이전 트램폴린 수행 과정에서 획득한 키를 현재의 이벤트 보고를 위한 키로 활용하여 공격 탐지 장치에게 전송한다(S720).

[55] 호스트 시스템(135)은 키와 함께 모드 변환 등의 이벤트를 보고 레지스터에 기록할 때(S720, S730), 다음 이벤트 보고에 사용할 키를 미리 수신하고(S740), 별도의 확인(ack) 과정 없이 보고 과정을 종료한다(S750).

[56] 도 8은 본 발명에 따른 공격 탐지 장치의 다른 일 예를 도시한 도면이다.

[57] 도 8을 참조하면, 공격 탐지 장치는 트래픽 관리부(800), 보고 관리부(810), 트래이스 관리부(830), 시스템 관리부(850)를 포함한다. 트래픽 관리부(800)는 스누퍼(802), 제1 검증부(804), 제1 알람 생성부(806)를 포함하고, 보고 관리부(810)는 보고 전달부(812), 키 제공부(814), 응답 제공부(816), 보고 수신부(818), 보고 관리 FSM(Finite State Machine)(820), 키 생성부(822), 제2 알람 생성부(824), 메모리 검증부(826), 메모리 접근부(828)를 포함하고, 트래이스 관리부(830)는 트래이스 수신부(832), 트래이스 분석부(834), 트래이스 검증부(836), 트래이스 관리 FSM(838), 알람 임시 저장부(840), 알람 생성부(842)를 포함한다.

[58] 먼저 시스템 관리부(850)는 자신의 프로세서와 메모리를 포함하며 트래픽 관리부(800), 보고 관리부(810), 트래이스 관리부(830) 등을 초기화하고 제어한다. 시스템 관리부(850)는 트래픽 관리부(800), 보고 관리부(810), 트래이스 관리부(830)의 각 알람 생성부(806, 824, 842)를 체크하여 커널 코드 삽입 공격 여부를 파악한다. 알람 생성부들(806, 824, 842) 중 어느 하나가 알람을 생성한 경우에, 시스템 관리부(850)는 시스템 관리자에게 알람이 발생했음을 알린다.

[59] 다음으로 트래픽 관리부(800)에 대해 살펴본다.

[60] 스누퍼(802)는 도 1과 같이 메모리 버스와 주변기기 버스의 트래픽을 스누핑하여 검증부(804)로 전달한다. 검증부(804)는 스누퍼(802)로부터 전달받은 트래픽이 커널 코드 변조와 같은 코드 삽입 공격의 징후인지를

판별하고 그 결과를 알람 생성부(806)로 전달한다. 검증부(804)는 트램폴린을 포함하는 물리 코드 영역, 가상 코드 영역을 맵핑하는 메모리 맵 레지스터와 페이지 테이블에 대한 쓰기(write) 시도를 공격 징후로 판단할 수 있다. 알람 생성부(806)는 검증부(804)에 의해 코드 삽입 공격 징후로 판단되면, 알람을 생성하여 출력한다.

[61] 다음으로 트레이스 관리부(830)에 대해 살펴본다.

[62] 트레이스 수신부(832)는 호스트 시스템의 PTI로부터 데이터 패킷을 수신한다. 트레이스 분석부(834)는 전달받은 패킷을 의미있는 정보로 가공한 후 트레이스 검증부(836)로 보낸다. 예를 들어, 트레이스 분석부(834)는 트레이스 수신부를 통해 수신한 패킷을 파싱하여 간접 분기 목적지 주소를 파악한다.

[63] 트레이스 검증부(836)는 트레이스 분석부(834)로부터 전달받은 정보를 이용하여 커널 코드 삽입 공격 시도 여부를 판별하고, 호스트 시스템의 상태에 따라 알람 임시 저장부(840) 또는 알람 생성부(842)에게 판별 결과를 전달한다.

[64] 예를 들어, 트레이스 검증부(836)는 트레이스 분석부(832)에 의해 파악된 간접 분기 목적지 주소들 중 사용자 모드에서 발생한 간접분기의 주소들을 제거한다. 모드 정보는 트레이스 관리 FSM(838)이 보고 관리부(810)로부터 수신하여 트레이스 검증부(836)에 제공할 수 있다. 트레이스 검증부(836)는 특권 모드와 관련 있는 주소 또는 어떤 모드와도 관련없는 주소가 있다면, 이를 공격 탐지 장치의 초기 구동시 파악한 호스트 시스템의 가상 코드 영역에 대한 주소 정보 등과 비교한다. 그리고 트레이스 검증부(836)는 주소가 가상 코드 영역에 있는 것이 아니라면 알람 임시 저장부(840)로 전달하고, 그 외에는 알람 생성부(842)로 전달한다.

[65] 트레이스 관리 FSM(838)은 보고 관리부로부터 모드 변환 등의 정보를 수신하면, 알람 임시 저장부(840)에 저장된 정보를 분석하여 알람 생성부(842)로 전달한다. 트레이스 검증(836)부는 데이터베이스에 저장되어 있는 트램폴린의 시작 및 끝 주소와 트레이스 분석부(834)에서 분석된 주소를 비교하여, 트램폴린의 중간에서 점핑을 통해 트램폴린 보고를 우회하는 시도를 검출한다. 트레이스 검증부(836)는 우회 시도를 나타내는 주소를 수신하면, 알람 생성부(842)를 통해 알람을 발생한다.

[66] 마지막으로 보고 관리부(810)에 대해 살펴본다.

[67] 보고 관리부(810)는 호스트 시스템의 현재 상태를 파악하기 위하여 호스트 시스템이 따르도록 정의된 프로토콜을 구현하고 있다. 키 생성부(822)는 트레이스 관리부(830)를 통해 트램폴린 시작이 통보되면, 키잉 레지스터(814)가 키를 폐치할 수 있도록 임의의 키를 준비한다. 메모리 검증부(826)는 호스트 시스템이 보고 레지스터를 통해 보고한 이벤트 내용을 검증(예를 들어, ATBR의 변경 여부)하기 위하여 호스트 시스템의 메인 메모리를 메모리 접근부(828)를 통해 접근하여 읽는다. 보고 관리 FSM(820)은 다른 모듈들을 제어하고 가짜 보고 메시지와 같은 악의적 행위가 검출되면 알람 생성부(824)를 통해 알람을

생성한다.

- [68] 보고 관리부(810)는 도 1과 같은 레지스터들을 포함할 수 있다. 키잉 레지스터(814)는 모든 쓰기 접근을 거부하고 오직 읽기 접근만을 허락한다. 예를 들어, 키잉 레지스터(814)는 각 트램폴린을 위한 키를 저장하는 값 필드, 키가 이용가능한지 여부를 나타내는 검증 필드를 포함한다. 보고 관리 FSM(820)이 호스트 시스템의 키 사용이 가능하도록 검증 필드를 설정하면, 키잉 레지스터(814)는 즉시 키 생성부(822)로부터 새로운 키를 패치하여 값 필드에 키를 저장한다. 이용 가능한 키가 호스트 시스템에 의해 패치되면, 키잉 레지스터(814)는 키가 이용 불가능함을 나타내기 위하여 검증 필드를 삭제(clear)한다. Ack 레지스터(816)는 확인(ack)을 나타내는 고정 값이 값 필드에 존재한다는 점을 제외하면, 키잉 레지스터(814)와 거의 동일하게 작동한다. 보고 레지스터(818)는 어떠한 읽기 접근도 허락하지 않으며, 쓰기 접근이 있으면 이를 보고 관리 FSM(820)으로 전달한다.
- [69] 예를 들어, ATBR 변경 보고를 처리하기 위하여, 보고 관리부(810)는 트램폴린의 시작을 기다린다. 트레이스 관리부(830)가 PTI로부터 트램폴린 시작을 보고받으면, 보고 관리 FSM(820)은 키가 이용 가능하도록 키잉 레지스터(814)를 구성하고, 키 생성부(822)로부터 키를 획득한 후 보고 레지스터(818)를 통해 수신되는 보고를 기다린다. 이벤트 보고가 키를 수반하여 수신되면, 보고 관리 FSM(820)은 메모리 검증부(826)가 페이지 테이블 콘텐츠를 검증하도록 제어하고, 가상 코드 영역의 테이블 콘텐츠가 올바른 경우에 Ack 레지스터(816)를 통해 확인 메시지를 제공한다.
- [70] 도 9는 본 발명에 따른 하드웨어 기반의 커널 코드 삽입 공격 탐지 방법의 일 실시 예의 흐름을 도시한 도면이다.
- [71] 도 9를 참조하면, 공격 탐지 장치는 호스트 시스템의 부트 동안에, 모드 변환, ATBR 주소 변환, PTI 제어 등과 관련된 커널의 코드 블록들의 위치를 파악하고, 파악한 코드 블록들을 트램폴린에 대한 분기 명령으로 대체된다(S900).
- [72] 공격 탐지 장치는 호스트 시스템의 PTI를 통해 런타임 정보를 포함하는 패킷을 수신하면, 수신한 패킷을 파싱하여 간접 분기 목적지 주소 등의 정보를 파악한다(S910). 이와 동시에 공격 탐지 장치는 호스트 시스템의 모드 변환 등이 발생하면 트램폴린을 통해 모드 변환 등을 보고받는다(S920). 모드 변환 등을 보고받는 구체적인 방법의 일 예는 도 3 및 도 7에 도시되어 있다. 또한 공격 탐지 장치는 메모리 버스나 주변기기 버스의 트래픽을 스누핑하여 물리 코드 영역에 대한 악의적 쓰기 시도 등을 검출한다(S930).
- [73] 공격 탐지 장치는 PTI를 통해 수신한 패킷을 분석하여 파악한 간접 분기 목적지 주소와 트램폴린을 통해 보고받은 모드 변환 등을 정보를 이용하여, 공격자가 가상 코드 영역 밖에서 프로세서 등을 특권 모드로 강제 변환하여 코드를 삽입하는 공격 등을 탐지한다(S940). 또한 공격 탐지 장치는 ATBR 주소 변환이나 제어 등의 이벤트가 발생하면 트램폴린을 통해 해당 이벤트의 발생

사실을 통지받아 공격자의 악의적인 주소 변경 등을 통한 공격을 탐지할 수 있다(S940).

- [74] 본 발명은 또한 컴퓨터로 읽을 수 있는 기록매체에 컴퓨터가 읽을 수 있는 코드로서 구현하는 것이 가능하다. 컴퓨터가 읽을 수 있는 기록매체는 컴퓨터 시스템에 의하여 읽혀질 수 있는 데이터가 저장되는 모든 종류의 기록장치를 포함한다. 컴퓨터가 읽을 수 있는 기록매체의 예로는 다양한 형태의 ROM, RAM, CD-ROM, 자기 테이프, 플로피디스크, 광데이터 저장장치 등이 있다. 또한 컴퓨터가 읽을 수 있는 기록매체는 네트워크로 연결된 컴퓨터 시스템에 분산되어 분산방식으로 컴퓨터가 읽을 수 있는 코드가 저장되고 실행될 수 있다.
- [75] 이제까지 본 발명에 대하여 그 바람직한 실시예들을 중심으로 살펴보았다. 본 발명이 속하는 기술 분야에서 통상의 지식을 가진 자는 본 발명이 본 발명의 본질적인 특성에서 벗어나지 않는 범위에서 변형된 형태로 구현될 수 있음을 이해할 수 있을 것이다. 그러므로 개시된 실시예들은 한정적인 관점이 아니라 설명적인 관점에서 고려되어야 한다. 본 발명의 범위는 전술한 설명이 아니라 특허청구범위에 나타나 있으며, 그와 동등한 범위 내에 있는 모든 차이점은 본 발명에 포함된 것으로 해석되어야 할 것이다.

청구범위

- [청구항 1] 호스트 시스템의 프로그램 트레이스 인터페이스(PTI)로부터 런타임 정보를 포함하는 패킷을 수신하고, 상기 패킷을 파싱하여 간접 분기 목적지 주소를 파악하는 트레이스 관리부;
호스트 시스템의 부트시 커널의 코드 블록들 중 주소변환 베이스 레지스터(ATBR) 변경, 모드 변환 또는 PTI 제어 레지스터 변경을 포함하는 이벤트와 관련 있는 코드 블록을 트램폴린에 대한 점프 명령으로 대체한 상태에서, 상기 이벤트 발생시 상기 트램폴린의 수행을 통해 이벤트를 보고받는 보고 관리부; 및
버스의 트래픽을 스누핑하여 메모리의 물리적 코드 영역에 대한 쓰기 시도를 검출하는 트래픽 관리부;를 포함하는 것을 특징으로 하는 하드웨어 기반의 커널 코드 삽입 공격 탐지 장치.
- [청구항 2] 제 1항에 있어서, 상기 트레이스 관리부는,
상기 보고 관리부에서 보고받은 이벤트를 기초로 상기 간접 분기 목적지 주소와 관련된 모드가 올바른지 파악하여 공격 여부를 탐지하는 것을 특징으로 하는 하드웨어 기반의 커널 코드 삽입 공격 탐지 장치.
- [청구항 3] 제 1항에 있어서,
상기 트레이스 관리부는 상기 PTI로부터 트램폴린의 시작을 통보받고, 상기 보고 관리부는 상기 트레이스 관리부에 의해 트램폴린의 시작을 통보받으면 키를 생성하여 상기 호스트 시스템으로 제공하고, 상기 호스트 시스템으로부터 키와 모드 변환 정보 또는 주소변환 베이스 레지스터(ATBR) 변경 정보를 수신하는 것을 특징으로 하는 하드웨어 기반의 커널 코드 삽입 공격 탐지 장치.
- [청구항 4] 제 3항에 있어서,
상기 보고 관리부는, 키잉 레지스터, 보고 레지스터 및 Ack 레지스터를 포함하고,
상기 보고 관리부는 상기 트레이스 관리부에 의해 트램폴린의 시작을 통보받으면 키를 생성하여 상기 키잉 레지스터에 저장하고, 상기 호스트 시스템이 상기 키잉 레지스터를 한 번 읽으면 상기 키잉 레지스터를 리셋하고, 상기 호스트 시스템이 상기 보고 레지스터에 모드 변환 정보 또는 주소변환 베이스 레지스터(ATBR) 변경 정보를 기록하면 상기 보고 레지스터를 읽어 모드 변환 정보를 파악하는 것을 특징으로 하는 하드웨어 기반의 커널 코드 삽입 공격 탐지 장치.
- [청구항 5] 제 1항에 있어서, 상기 트래픽 관리부는,
버스의 트래픽을 스누핑하여 불변코드영역, ATBR의 제1 레벨 페이지 테이블 또는 PTI에 대한 접근을 감시하는 것을 특징으로 하는 하드웨어 기반의 커널 코드 삽입 공격 탐지 장치.

- [청구항 6] 호스트 시스템의 프로그램 트레이스 인터페이스(PTI)로부터 런타임 정보를 포함하는 패킷을 수신하고, 상기 패킷을 파싱하여 간접 분기 목적지 주소를 파악하는 단계;
호스트 시스템의 부트시 커널의 코드 블록들 중 주소변환 베이스 레지스터(ATBR) 변경, 모드 변환 또는 PTI 제어 레지스터 변경을 포함하는 이벤트와 관련 있는 코드 블록을 트램폴린에 대한 점프 명령으로 대체한 상태에서, 상기 이벤트 발생시 상기 트램폴린의 수행을 통해 이벤트를 보고받는 단계; 및
버스의 트래픽을 스누핑하여 메모리의 물리적 코드 영역에 대한 쓰기 시도를 검출하는 단계;를 포함하는 것을 특징으로 하는 하드웨어 기반의 커널 코드 삽입 공격 탐지 방법.
- [청구항 7] 제 6항에 있어서,
상기 보고받은 이벤트를 기초로 상기 간접 분기 목적지 주소와 관련된 모드가 올바른지 파악하여 공격 여부를 탐지하는 단계;를 더 포함하는 것을 특징으로 하는 하드웨어 기반의 커널 코드 삽입 공격 탐지 방법.
- [청구항 8] 제 6항에 있어서,
상기 PTI로부터 트램폴린의 시작을 통보받는 단계;
키를 생성하여 상기 호스트 시스템으로 제공하는 단계;
상기 호스트 시스템으로부터 키와 이벤트 정보를 순차적으로 수신하는 단계;를 더 포함하는 것을 특징으로 하는 하드웨어 기반의 커널 코드 삽입 공격 탐지 방법.
- [청구항 9] 제 6항 내지 제 8항 중 어느 한 항에 기재된 방법을 수행하기 위한 프로그램을 기록한 컴퓨터로 읽을 수 있는 기록매체.