



(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2022-0068877
(43) 공개일자 2022년05월26일

(51) 국제특허분류(Int. Cl.)
G06F 21/57 (2013.01) G06F 21/55 (2013.01)
G06F 8/10 (2018.01)
(52) CPC특허분류
G06F 21/57 (2013.01)
G06F 21/55 (2013.01)
(21) 출원번호 10-2021-0020251
(22) 출원일자 2021년02월16일
심사청구일자 2021년02월16일
(30) 우선권주장
1020200155581 2020년11월19일 대한민국(KR)

(71) 출원인
고려대학교 산학협력단
서울특별시 성북구 안암로 145, 고려대학교 (안암동5가)
(72) 발명자
김승주
서울특별시 강남구 도곡로57길 12, 103동 503호(역삼동, 역삼2차아이파크)
강수영
서울특별시 성북구 월곡로14길 26, 108동 1302호(하월곡동, 월곡래미안루나밸리)
김예준
서울특별시 성북구 고려대로12길 32, 303호(안암동3가)
(74) 대리인
이대호, 박건홍

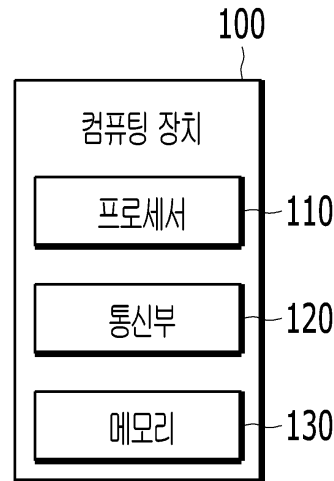
전체 청구항 수 : 총 7 항

(54) 발명의 명칭 보안 시스템 설계 및 검증을 위한 모델 기반 보안 프레임워크

(57) 요약

본 개시의 몇몇 실시예에 따라 컴퓨팅 장치의 프로세서에 의해 수행되는 모델 기반 보안 프레임워크를 이용하여 보안 시스템 설계 및 검증을 수행하는 방법이 개시된다. 상기 방법은: 설계 대상 시스템의 보안 요구사항을 도출하기 위한 위협 모델링 단계; 상기 보안 요구사항에 기초하여 설계 모델을 생성하고, 상기 설계 모델을 검증하기 위한 설계 모델링 단계; 및 상기 설계 모델에 기초하여 구현 모델을 생성하고, 상기 구현 모델을 검증하기 위한 구현 모델링 단계;를 포함할 수 있다.

대표도 - 도2



(52) CPC특허분류
G06F 8/10 (2013.01)

이 발명을 지원한 국가연구개발사업

과제고유번호	1711103082
과제번호	2015-0-00403-006
부처명	과학기술정보통신부
과제관리(전문)기관명	정보통신기획평가원
연구사업명	정보통신방송혁신인재양성(R&D)
연구과제명	IoT 보안기술연구
기 여 율	1/1
과제수행기관명	순천향대학교산학협력단
연구기간	2015.06.01 ~ 2021.03.31

명세서

청구범위

청구항 1

컴퓨팅 장치의 프로세서에 의해 수행되는 모델 기반 보안 프레임워크를 이용하여 보안 시스템 설계 및 검증을 수행하는 방법에 있어서,

설계 대상 시스템의 보안 요구사항을 도출하기 위한 위협 모델링 단계;

상기 보안 요구사항에 기초하여 설계 모델을 생성하고, 상기 설계 모델을 검증하기 위한 설계 모델링 단계; 및

상기 설계 모델에 기초하여 구현 모델을 생성하고, 상기 구현 모델을 검증하기 위한 구현 모델링 단계;

를 포함하는,

보안 시스템 설계 및 검증을 수행하는 방법.

청구항 2

제 1 항에 있어서,

상기 위협 모델링 단계는,

상기 설계 대상 시스템의 기능 요구사항에 기초하여, 보안이 요구되는 중요 자산을 인식하는 단계;

상기 설계 대상 시스템의 구조에 기초하여, 주요 구성요소 기반의 데이터 흐름도를 생성하는 단계;

복수의 자료에 기초하여, 상기 설계 대상 시스템의 취약점 및 상기 설계 대상 시스템에 대한 공격 방법과 관련된 공격 라이브러리를 생성하는 단계;

사전 설정된 보안 속성에 기초하여, 상기 설계 대상 시스템에서 발생될 수 있는 예상 위협을 인식하는 단계;

상기 데이터 흐름도에 상기 예상 위협을 적용함에 따라 발생하는 식별 위협 및 상기 공격 라이브러리에 기초하여, 공격 트리를 생성하는 단계; 및

상기 데이터 흐름도 및 상기 공격 라이브러리를 통해 도출된 위협 목록과 상기 공격 트리를 이용하여 보안 체크리스트를 생성하고, 상기 보안 체크리스트에 대응하는 상기 보안 요구사항을 도출하는 단계;

를 포함하는,

보안 시스템 설계 및 검증을 수행하는 방법.

청구항 3

제 1 항에 있어서,

상기 설계 모델링 단계는,

표준화된 모델링 언어를 이용하여, 보안 요구사항에 대응하는 설계 모델을 생성하는 단계; 및

형식 언어를 이용하여, 상기 설계 모델과 상기 보안 요구사항 간 모순이 없는지에 대한 정확성을 검증하는 단계;

를 포함하는,

보안 시스템 설계 및 검증을 수행하는 방법.

청구항 4

제 1 항에 있어서,
 상기 구현 모델링 단계는,
 상기 설계 모델에 기초하여, 소스 코드를 생성하는 단계;
 상기 소스 코드를 검증 모델링 언어로 변환하여, 구현 모델을 생성하는 단계; 및
 모델 체커를 이용하여 상기 구현 모델에 대한 정형 검증을 수행하는 단계;
 를 포함하는,
 보안 시스템 설계 및 검증을 수행하는 방법.

청구항 5

제 2 항에 있어서,
 상기 중요 자산은,
 상기 설계 대상 시스템의 특성 및 기능에 대응하여 메모리에 사전 저장되어 있거나 또는, 사용자 입력에 기초하여 결정되는,
 보안 시스템 설계 및 검증을 수행하는 방법.

청구항 6

제 2 항에 있어서,
 상기 데이터 흐름도는,
 상기 설계 대상 시스템에서 제공되는 복수의 기능들 각각에 대한 정보 및 상기 복수의 기능들 사이에서 송수신되는 데이터에 대한 정보를 포함하는,
 보안 시스템 설계 및 검증을 수행하는 방법.

청구항 7

제 2 항에 있어서,
 상기 공격 트리는,
 최종 공격 목표에 대한 정보를 포함하고,
 상기 공격 트리의 최상위 노드에 상기 최종 공격 목표가 배치되고,
 상기 최상위 노드와 연관된 적어도 하나의 하위 노드에 상기 최종 공격 목표와 관련된 적어도 하나의 위협이 배치되는,
 보안 시스템 설계 및 검증을 수행하는 방법.

발명의 설명

기술 분야

본 개시는 보안 시스템 개발 시 보안요구사항에 따라 제대로 설계되고 구현되었는지 확인하기 위한 방법에 관한 것으로, 보안 시스템에 대한 모델기반으로 위협모델링, 설계 모델링, 구현 모델링을 결합하여 검증할 수 있는

[0001]

프레임워크에 관한 것이다. 구체적으로, 본 개시는 정형 명세 및 검증에 대한 모델기반 보안 프레임워크를 통해 보안 시스템의 설계 및 구현 모델을 검증하여 신뢰할 수 있는 시스템을 개발하기 위한 방법에 관한 것이다.

배경 기술

[0002] IT 시스템의 발전과 더불어 정보 시스템의 보안문제로 인해 심각한 재정적 손실이나 인명 피해가 발생할 수 있기에 보안에 대한 관심이 점점 더 많아지고 있다. 이러한 보안 문제의 일부는 잘못된 설계로 인해 발생하고 있다. 대부분의 시스템은 보안 기능이 제대로 설계되지 않고 개발 시 시큐어 코딩이나 후반의 모의해킹에 의존하여 구현되고 있다. 하지만, 도 1의 IBM 시스템 과학 연구소(IBM System Science Institute)의 단계별 결함 수정에 발생하는 비용 연구 결과에 의하면 보안 시스템 개발 생명주기 초기에 문제를 식별하여 수정하는 경우에는 개발 완료에 수정할 때 보다 약 100배의 비용을 절감할 수 있기 때문에 개발 초기 단계인 요구사항 분석 및 설계 단계부터 보안 기능을 고려하여야 한다고 한다.

[0003] 설계 단계에서 보안 기능을 고려한 뒤, 설계된 보안 기능이 제대로 구현되고 동작하는지 확인하기 위하여 설계에서부터 구현 단계까지의 정확성을 만족해야 하며, 이를 위해 보안 시스템에 대한 설계 기법과 정형 명세 및 검증 기법 연구가 지속적으로 진행되고 있다.

선행기술문헌

특허문헌

[0004] (특허문헌 0001) 대한민국 등록특허 공보 제10-1203879호

발명의 내용

해결하려는 과제

[0005] 본 개시는 전술한 배경기술에 대응하여 안출된 것으로, 보안 시스템 설계 및 검증을 위한 모델 기반 보안 프레임워크를 제공하고자 한다. 본 개시는 제품의 수준을 증대시키기 위해 시스템에서 발생할 수 있는 다양한 보안 위협을 개발 초기 단계부터 고려할 수 있다.

[0006] 구체적으로, 시스템은 설계된 모델을 통해 구현이 이루어지기 때문에 보안 기능 요구사항, 설계 및 구현 모델에 대한 검증을 통해 설계 모델과 구현물 간의 정확성이 검증되어야 한다. 또한, 보안 기능에 대한 설계가 제대로 수행되었더라도 이후 단계에서 설계 사항이 제대로 반영되지 못한다면 구현 단계에서 잠재적 위협이 발생할 수 있고 결함을 수정하기 위해 발생하는 비용이 커지게 된다.

[0007] 따라서, 요구사항에 대한 설계 모델 생성 및 검증부터 구현 모델에 대한 검증까지의 정확성 검증을 통해 결함 수정에 대한 비용을 줄이고, 보안성을 향상시킬 수 있는 보안 시스템 설계 및 검증을 위한 모델 기반의 보안 프레임워크 제공을 목적으로 한다.

[0008] 본 개시의 기술적 과제들은 이상에서 언급한 기술적 과제로 제한되지 않으며, 언급되지 않은 또 다른 기술적 과제들은 아래의 기재로부터 당업자에게 명확하게 이해될 수 있을 것이다.

과제의 해결 수단

[0009] 전술한 바와 같은 과제를 해결하기 위한 본 개시의 몇몇 실시예에 따라, 보안 시스템 개발 시 보안요구사항에 따라 제대로 설계되고 구현되었는지 확인하기 위한 방법을 제공할 수 있다.

[0010] 구체적으로, 본 개시는 보안 시스템에 대한 모델기반으로 위협모델링, 설계 모델링, 구현 모델링을 결합하여 검증할 수 있는 프레임워크를 제공할 수 있다.

[0011] 또한, 본 개시는 정형 명세 및 검증에 대한 모델기반 보안 프레임워크를 통해 보안 시스템의 설계 및 구현 모델을 검증하여 신뢰할 수 있는 시스템을 개발하기 위한 방법을 제공할 수 있다.

[0012] 전술한 바와 같은 과제를 해결하기 위한 본 개시의 몇몇 실시예에 따라, 컴퓨팅 장치의 프로세서에 의해 수행되는 모델 기반 보안 프레임워크를 이용하여 보안 시스템 설계 및 검증을 수행하는 방법이 개시된다. 상기 방법은: 설계 대상 시스템의 보안 요구사항을 도출하기 위한 위협 모델링 단계; 상기 보안 요구사항에 기초하여

설계 모델을 생성하고, 상기 설계 모델을 검증하기 위한 설계 모델링 단계; 및 상기 설계 모델에 기초하여 구현 모델을 생성하고, 상기 구현 모델을 검증하기 위한 구현 모델링 단계;를 포함할 수 있다.

[0013] 또한, 상기 위협 모델링 단계는, 상기 설계 대상 시스템의 기능 요구사항에 기초하여, 보안이 요구되는 중요 자산을 인식하는 단계; 상기 설계 대상 시스템의 구조에 기초하여, 주요 구성요소 기반의 데이터 흐름도를 생성하는 단계; 복수의 자료에 기초하여, 상기 설계 대상 시스템의 취약점 및 상기 설계 대상 시스템에 대한 공격 방법과 관련된 공격 라이브러리를 생성하는 단계; 사전 설정된 보안 속성에 기초하여, 상기 설계 대상 시스템에서 발생될 수 있는 예상 위협을 인식하는 단계; 상기 데이터 흐름도에 상기 예상 위협을 적용함에 따라 발생하는 식별 위협 및 상기 공격 라이브러리에 기초하여, 공격 트리를 생성하는 단계; 및 상기 데이터 흐름도 및 상기 공격 라이브러리를 통해 도출된 위협 목록과 상기 공격 트리를 이용하여 보안 체크리스트를 생성하고, 상기 보안 체크리스트에 대응하는 상기 보안 요구사항을 도출하는 단계;를 포함할 수 있다.

[0014] 또한, 상기 설계 모델링 단계는, 표준화된 모델링 언어를 이용하여, 보안 요구사항에 대응하는 설계 모델을 생성하는 단계; 및 형식 언어를 이용하여, 상기 설계 모델과 상기 보안 요구사항 간 모순이 없는지에 대한 정확성을 검증하는 단계;를 포함할 수 있다.

[0015] 또한, 상기 구현 모델링 단계는, 상기 설계 모델에 기초하여, 소스 코드를 생성하는 단계; 상기 소스 코드를 검증 모델링 언어로 변환하여, 구현 모델을 생성하는 단계; 및 모델 체커를 이용하여 상기 구현 모델에 대한 정형 검증을 수행하는 단계;를 포함할 수 있다.

[0016] 또한, 상기 중요 자산은, 상기 설계 대상 시스템의 특성 및 기능에 대응하여 메모리에 사전 저장되어 있거나 또는, 사용자 입력에 기초하여 결정될 수 있다.

[0017] 또한, 상기 데이터 흐름도는, 상기 설계 대상 시스템에서 제공되는 복수의 기능들 각각에 대한 정보 및 상기 복수의 기능들 사이에서 송수신 되는 데이터에 대한 정보를 포함할 수 있다.

[0018] 또한, 상기 공격 트리는, 최종 공격 목표에 대한 정보를 포함하고, 상기 공격 트리의 최상위 노드에 상기 최종 공격 목표가 배치되고, 상기 최상위 노드와 연관된 적어도 하나의 하위 노드에 상기 최종 공격 목표와 관련된 적어도 하나의 위협이 배치될 수 있다.

[0019] 본 개시에서 얻을 수 있는 기술적 해결 수단은 이상에서 언급한 해결 수단들로 제한되지 않으며, 언급하지 않은 또 다른 해결 수단들은 아래의 기재로부터 본 개시가 속하는 기술분야에서 통상의 지식을 가진 자에게 명확하게 이해될 수 있을 것이다.

발명의 효과

[0020] 본 개시는 설계 및 구현 단계에서 UML과 OCL, Promela, SPIN 모델체커와 같은 공개된 도구를 사용하여 프레임워크 사용자에게 편의성을 제공할 수 있다.

[0021] 또한, 본 개시는 위협모델링과 정형 명세 및 검증 기법을 사용한 모델 기반의 보안 프레임워크를 통해 체계적으로 보안 시스템에 대한 설계 및 검증을 통해 개발 초기에 보안 기능을 고려함으로써 제품에 대한 보안성을 향상시키고 개발 시 발생하는 비용을 절감시킬 수 있다.

[0022] 또한, 본 개시는 정형 명세 및 검증 기법에 대한 체계적인 프레임워크를 통해 신뢰할 수 있는 보안 시스템 개발에 도움을 줄 수 있다.

[0023] 본 개시에서 얻을 수 있는 효과는 이상에서 언급한 효과로 제한되지 않으며, 언급하지 않은 또 다른 효과들은 아래의 기재로부터 본 개시가 속하는 기술분야에서 통상의 지식을 가진 자에게 명확하게 이해될 수 있을 것이다.

도면의 간단한 설명

[0024] 다양한 양상들이 이제 도면들을 참조로 기재되며, 여기서 유사한 참조 번호들은 총괄적으로 유사한 구성요소들을 지칭하는데 이용된다. 이하의 실시예에서, 설명 목적을 위해, 다수의 특정 세부사항들이 하나 이상의 양상들의 총체적 이해를 제공하기 위해 제시된다. 그러나, 그러한 양상(들)이 이러한 특정 세부사항들 없이 실시될 수 있음은 명백할 것이다. 다른 예시들에서, 공지의 구조들 및 장치들이 하나 이상의 양상들의 기재를 용이하게 하기 위해 블록도 형태로 도시된다.

도 1은 단계별 결합 수정에 발생하는 비용 연구 결과에 대한 그래프이다.

도 2는 본 개시의 보안 시스템 설계 및 검증을 위한 모델 기반 보안 프레임워크와 관련된 컴퓨팅 장치의 블록 구성도이다.

도 3은 본 개시의 몇몇 실시예에 따른 보안 시스템 개발 시 보안요구사항에 따라 제대로 설계되고 구현되었는지 확인하기 위한 방법을 설명하기 위한 도면이다.

도 4는 본 개시의 몇몇 실시예에 따른 데이터 흐름도의 일례를 설명하기 위한 도면이다.

도 5는 본 개시의 몇몇 실시예에 따른 공격 트리의 일례를 설명하기 위한 도면이다.

도 6은 본 개시내용의 실시예들이 구현될 수 있는 예시적인 컴퓨팅 환경에 대한 일반적인 개략도를 도시한다.

발명을 실시하기 위한 구체적인 내용

- [0025] 다양한 실시예들 및/또는 양상들이 이제 도면들을 참조하여 개시된다. 하기 설명에서는 설명을 목적으로, 하나 이상의 양상들의 전반적 이해를 돕기 위해 다수의 구체적인 세부사항들이 개시된다. 그러나, 이러한 양상(들)은 이러한 구체적인 세부사항들 없이도 실행될 수 있다는 점 또한 본 개시의 기술 분야에서 통상의 지식을 가진 자에게 감지될 수 있을 것이다. 이후의 기재 및 첨부된 도면들은 하나 이상의 양상들의 특정한 예시적인 양상들을 상세하게 기술한다. 하지만, 이러한 양상들은 예시적인 것이고 다양한 양상들의 원리들에서의 다양한 방법들 중 일부가 이용될 수 있으며, 기술되는 설명들은 그러한 양상들 및 그들의 균등물들을 모두 포함하고자 하는 의도이다. 구체적으로, 본 명세서에서 사용되는 "실시예", "예", "양상", "예시" 등은 기술되는 임의의 양상 또는 설계가 다른 양상 또는 설계들보다 양호하다거나, 이점이 있는 것으로 해석되지 않을 수도 있다.
- [0026] 이하, 도면 부호에 관계없이 동일하거나 유사한 구성 요소는 동일한 참조 번호를 부여하고 이에 대한 중복되는 설명은 생략한다. 또한, 본 명세서에 개시된 실시예를 설명함에 있어서 관련된 공지 기술에 대한 구체적인 설명이 본 명세서에 개시된 실시예의 요지를 흐릴 수 있다고 판단되는 경우 그 상세한 설명을 생략한다. 또한, 첨부된 도면은 본 명세서에 개시된 실시예를 쉽게 이해할 수 있도록 하기 위한 것일 뿐, 첨부된 도면에 의해 본 명세서에 개시된 기술적 사상이 제한되지 않는다.
- [0027] 비록 제 1, 제 2 등이 다양한 소자나 구성요소들을 서술하기 위해서 사용되나, 이들 소자나 구성요소들은 이들 용어에 의해 제한되지 않음은 물론이다. 이들 용어들은 단지 하나의 소자나 구성요소를 다른 소자나 구성요소와 구별하기 위하여 사용하는 것이다. 따라서, 이하에서 언급되는 제 1 소자나 구성요소는 본 발명의 기술적 사상 내에서 제 2 소자나 구성요소 일 수도 있음은 물론이다.
- [0028] 다른 정의가 없다면, 본 명세서에서 사용되는 모든 용어(기술 및 과학적 용어를 포함)는 본 발명이 속하는 기술 분야에서 통상의 지식을 가진 자에게 공통적으로 이해될 수 있는 의미로 사용될 수 있을 것이다. 또 일반적으로 사용되는 사전에 정의되어 있는 용어들은 명백하게 특별히 정의되어 있지 않는 한 이상적으로 또는 과도하게 해석되지 않는다.
- [0029] 더불어, 용어 "또는"은 배타적 "또는"이 아니라 내포적 "또는"을 의미하는 것으로 의도된다. 즉, 달리 특정되지 않거나 문맥상 명확하지 않은 경우에, "X는 A 또는 B를 이용한다"는 자연적인 내포적 치환 중 하나를 의미하는 것으로 의도된다. 즉, X가 A를 이용하거나; X가 B를 이용하거나; 또는 X가 A 및 B 모두를 이용하는 경우, "X는 A 또는 B를 이용한다"가 이들 경우들 어느 것으로도 적용될 수 있다. 또한, 본 명세서에 사용된 "및/또는"이라는 용어는 열거된 관련 아이템들 중 하나 이상의 아이템의 가능한 모든 조합을 지칭하고 포함하는 것으로 이해되어야 한다.
- [0030] 또한, "포함한다" 및/또는 "포함하는"이라는 용어는, 해당 특징 및/또는 구성요소가 존재함을 의미하지만, 하나 이상의 다른 특징, 구성요소 및/또는 이들의 그룹의 존재 또는 추가를 배제하지 않는 것으로 이해되어야 한다. 또한, 달리 특정되지 않거나 단수 형태를 지시하는 것으로 문맥상 명확하지 않은 경우에, 본 명세서와 청구범위에서 단수는 일반적으로 "하나 또는 그 이상"을 의미하는 것으로 해석되어야 한다.
- [0031] 더불어, 본 명세서에서 사용되는 용어 "정보" 및 "데이터"는 종종 서로 상호교환 가능하도록 사용될 수 있다.
- [0032] 어떤 구성 요소가 다른 구성 요소에 "연결되어" 있다거나 "접속되어" 있다고 언급된 때에는, 그 다른 구성 요소에 직접적으로 연결되어 있거나 또는 접속되어 있을 수도 있지만, 중간에 다른 구성 요소가 존재할 수도 있다고 이해되어야 할 것이다. 반면에, 어떤 구성 요소가 다른 구성 요소에 "직접 연결되어" 있다거나 "직접

접속되어” 있다고 언급된 때에는, 중간에 다른 구성 요소가 존재하지 않는 것으로 이해되어야 할 것이다.

- [0033] 이하의 설명에서 사용되는 구성 요소에 대한 접미사 “모듈” 및 “부”는 명세서 작성의 용이함만이 고려되어 부여되거나 혼용되는 것으로서 그 자체로 서로 구별되는 의미 또는 역할을 갖는 것은 아니다.
- [0034] 본 개시의 목적 및 효과, 그리고 그것들을 달성하기 위한 기술적 구성들은 첨부되는 도면과 함께 상세하게 후술되어 있는 실시예들을 참조하면 명확해질 것이다. 본 개시를 설명하는데 있어서 공지 기능 또는 구성에 대한 구체적인 설명이 본 개시의 요지를 불필요하게 흐릴 수 있다고 판단되는 경우에는 그 상세한 설명을 생략할 것이다. 그리고 후술되는 용어들은 본 개시에서의 기능을 고려하여 정의된 용어들로써 이는 사용자, 운용자의 의도 또는 관례 등에 따라 달라질 수 있다.
- [0035] 그러나 본 개시는 이하에서 개시되는 실시예들에 한정되는 것이 아니라 서로 다른 다양한 형태로 구현될 수 있다. 단지 본 실시예들은 본 개시가 완전하도록 하고, 본 개시가 속하는 기술분야에서 통상의 지식을 가진 자에게 개시의 범주를 완전하게 알려주기 위해 제공되는 것이며, 본 개시는 청구항의 범주에 의해 정의될 뿐이다. 그러므로 그 정의는 본 명세서 전반에 걸친 내용을 토대로 내려져야 할 것이다.
- [0037] 도 2는 본 개시의 보안 시스템 설계 및 검증을 위한 모델 기반 보안 프레임워크와 관련된 서버의 블록 구성도이다.
- [0038] 도 2를 참조하면, 컴퓨팅 장치(100)는 프로세서(110), 통신부(120) 및 메모리(130)를 포함할 수 있다. 다만, 상술한 구성 요소들은 컴퓨팅 장치(100)를 구현하는데 있어서 필수적인 것은 아니어서, 컴퓨팅 장치(100)는 위에서 열거된 구성요소들 보다 많거나, 또는 적은 구성요소들을 가질 수 있다.
- [0039] 컴퓨팅 장치(100)는 예를 들어, 마이크로프로세서, 메인프레임 컴퓨터, 디지털 프로세서, 휴대용 디바이스 및 디바이스 제어기 등과 같은 임의의 타입의 컴퓨터 시스템 또는 컴퓨터 디바이스를 포함할 수 있다. 다만, 이에 한정되는 것은 아니다.
- [0040] 컴퓨팅 장치(100)의 프로세서(110)는 통상적으로 컴퓨팅 장치(100)의 전반적인 동작을 제어한다. 프로세서(110)는 컴퓨팅 장치(100)에 포함된 구성요소들을 통해 입력 또는 출력되는 신호, 데이터, 정보 등을 처리하거나 메모리(130)에 저장된 응용 프로그램을 구동함으로써, 사용자에게 적절한 정보 또는 기능을 제공 또는 처리할 수 있다.
- [0041] 또한, 프로세서(110)는 메모리(130)에 저장된 응용 프로그램을 구동하기 위하여, 컴퓨팅 장치(100)의 구성요소들 중 적어도 일부를 제어할 수 있다. 나아가, 프로세서(110)는 상기 응용 프로그램의 구동을 위하여, 컴퓨팅 장치(100)에 포함된 구성요소들 중 적어도 둘 이상을 서로 조합하여 동작시킬 수 있다.
- [0042] 컴퓨팅 장치(100)의 통신부(120)는, 컴퓨팅 장치(100)와 사용자 단말 사이 및 컴퓨팅 장치(100)와 외부 서버들 사이의 통신을 가능하게 하는 하나 이상의 모듈을 포함할 수 있다. 또한, 상기 통신부(120)는, 컴퓨팅 장치(100)를 하나 이상의 네트워크에 연결하는 하나 이상의 모듈을 포함할 수 있다.
- [0043] 컴퓨팅 장치(100)와 사용자 단말 사이 및 컴퓨팅 장치(100)와 외부 서버들 사이의 통신을 연결하는 네트워크는 공중전화 교환망(PSTN:Public Switched Telephone Network), xDSL(x Digital Subscriber Line), RADSL(Rate Adaptive DSL), MDSL(Multi Rate DSL), VDSL(Very High Speed DSL), UADSL(Universal Asymmetric DSL), HDSL(High Bit Rate DSL) 및 근거리 통신망(LAN) 등과 같은 다양한 유선 통신 시스템들을 사용할 수 있다.
- [0044] 또한, 여기서 제시되는 네트워크(600)는 CDMA(Code Division Multi Access), TDMA(Time Division Multi Access), FDMA(Frequency Division Multi Access), OFDMA(Orthogonal Frequency Division Multi Access), SC-FDMA(Single Carrier-FDMA) 및 다른 시스템들과 같은 다양한 무선 통신 시스템들을 사용할 수 있다.
- [0045] 본 개시의 실시예들에 따른 네트워크는 유선 및 무선 등과 같은 그 통신 양태를 가리지 않고 구성될 수 있으며, 단거리 통신망(LAN: Local Area Network), 원거리 통신망(WAN: Wide Area Network) 등 다양한 통신망으로 구성될 수 있다. 또한, 상기 네트워크는 공지의 월드와이드웹(WWW:World Wide Web)일 수 있으며, 적외선(IrDA:Infrared Data Association) 또는 블루투스(Bluetooth)와 같이 단거리 통신에 이용되는 무선 전송 기술을 이용할 수도 있다.
- [0046] 본 명세서에서 설명된 기술들은 위에서 언급된 네트워크들뿐만 아니라, 다른 네트워크들에서도 사용될 수 있다.

- [0047] 컴퓨팅 장치(100)의 메모리(130)는 프로세서(110)의 동작을 위한 프로그램을 저장할 수 있고, 입/출력되는 데이터들을 임시 또는 영구 저장할 수도 있다. 메모리(130)는 플래시 메모리 타입(flash memory type), 하드디스크 타입(hard disk type), 멀티미디어 카드 마이크로 타입(multimedia card micro type), 카드 타입의 메모리(예를 들어 SD 또는 XD 메모리 등), 램(Random Access Memory, RAM), SRAM(Static Random Access Memory), 롬(Read-Only Memory, ROM), EEPROM(Electrically Erasable Programmable Read-Only Memory), PROM(Programmable Read-Only Memory), 자기 메모리, 자기 디스크, 광디스크 중 적어도 하나의 타입의 저장매체를 포함할 수 있다. 이러한 메모리(130)는 프로세서(110)에 제어에 의하여 동작 될 수 있다.
- [0048] 소프트웨어적인 구현에 의하면, 본 명세서에서 설명되는 절차 및 기능과 같은 실시예들은 별도의 소프트웨어 모듈들로 구현될 수 있다. 상기 소프트웨어 모듈들 각각은 본 명세서에서 설명되는 하나 이상의 기능 및 작동을 수행할 수 있다. 적절한 프로그램 언어로 쓰여진 소프트웨어 어플리케이션으로 소프트웨어 코드가 구현될 수 있다. 상기 소프트웨어 코드는 컴퓨팅 장치(100)의 메모리(130)에 저장되고, 컴퓨팅 장치(100)의 프로세서(110)에 의해 실행될 수 있다.
- [0049] 본 개시의 몇몇 실시예에 따른 컴퓨팅 장치(100)는 보안 시스템 설계 및 검증을 위한 모델 기반 보안 프레임워크에 기반하여, 보안 시스템 개발 시 보안요구사항에 따라 제대로 설계되고 구현되었는지 확인하기 위한 방법과 관련된 동작들을 수행할 수 있다.
- [0050] 구체적으로, 본 개시의 컴퓨팅 장치(100)는 요구사항 분석 단계, 설계 단계 및 구현 단계로 이루어진 모델기반 보안 프레임워크를 통해 보안 시스템의 설계 및 구현 모델을 검증할 수 있다.
- [0051] 좀더 구체적으로, 본 개시의 몇몇 실시예에 따르면, 보안 시스템 설계자(컴퓨팅 장치(100) 또는, 단말(미도시))는 설계 모델링에 필요한 보안 기능 요구사항을 도출하기 위하여 보안 시스템에 대한 위협모델링을 수행한다. 설계자가 위협모델링을 수행하기 위해서는 시스템에 대한 기능 요구사항이 필요하다. 설계자는 기능 요구사항을 기반으로 시스템 내에서 악의적인 사용자에게 의해 변조되거나 유출되지 않아야 할 자산에 대하여 식별한다. 여기서, 자산 식별은 분석 대상 내 보호하고자 하는 데이터의 중요 자산에 대하여 파악하는 단계이다.
- [0052] 설계자는 자산 식별이 완료된 이후, 보안 위협의 구체화 및 식별을 수행하기 위해 분석 대상 시스템에서 발견된 취약점을 확보하여 공격 라이브러리를 구축한다. 여기서, 공격 라이브러리 수집은 위협모델링의 초기 프로세스에서 수행하지만, 수집 이후 새로운 위협이 발생하거나 미처 수집하지 못한 위협이 존재할 수 있으므로 전 프로세스에 걸쳐 지속해서 수행한다.
- [0053] 설계자는 자산 식별 및 공격 라이브러리 수집 과정에서 파악한 분석 대상 시스템의 중요 구성요소를 기반으로 이를 도식화하여 데이터 흐름도를 작성한다.
- [0054] 설계자는 데이터 흐름도 작성 과정에서 분석 대상 시스템의 구조에 대하여 명확하게 파악하는 것이 가능하다. 설계자는 작성한 데이터 흐름도와 수집한 공격 라이브러리를 이용하여 STRIDE 보안속성에 따라 시스템의 구성요소 자체에서 발생 가능한 위협과 구성요소 간 상호작용하는 과정에서 발생 가능한 위협을 도출한다. 설계자는 공격 트리 작성 단계에서, 최종적으로 수행될 수 있는 공격 목표를 최상위 노드로 설정한다. 설계자는 설정한 최종 공격 목표를 이루기 위한 과정 및 수단으로써 위협 도출단계에서 도출한 위협을 하위 트리의 노드로 배치한다. 설계자는 공격 트리를 통해 도출된 위협들이 시스템의 중요 자산에 대해 최종 공격을 수행하는 공격 시나리오를 파악할 수 있다. 설계자는 작성된 공격트리의 최하위 공격 노드에 대한 킬체인을 구성하기 위한 완화 기법을 식별하고 이를 이루기 위한 체크리스트를 생성하여 최종적으로 보안 기능 요구사항을 도출한다.
- [0055] 설계자는 위협모델링을 이용하여 도출된 보안 기능 요구사항은 이후 설계 검증을 위해 사용된다. 설계자는 보안 기능 요구사항을 구체화하여 UML을 통해 설계 모델을 생성하고 OCL을 이용하여 설계 모델과 보안 기능 요구사항 간 모순이 없는지에 대한 정확성 검증을 수행한다.
- [0056] 설계자는 정확성 검증이 완료된 설계 모델에 대한 소스코드를 생성하고 이를 정형 명세 및 검증하는 구현 검증 단계를 수행한다. 설계자는 모델체커인 SPIN을 통해 검증하기 위하여 Promela를 통해 정형 명세를 수행한다. 이후 설계자는 SPIN을 이용하여 정형 명세가 완료된 소스코드인 구현 모델과 보안 기능 요구사항에 따라 맞게 동작하는지 구현 모델 검증을 수행하여 최종적으로 모델링 기반의 설계 모델 및 구현 모델에 대한 검증 결과를 도출할 수 있다.
- [0057] 상술한 바와 같이, 본 개시의 몇몇 실시예에 따른 모델 기반 보안 프레임워크는 UML을 이용한 설계 모델 생성, OCL을 이용한 설계 검증과 Promela를 이용한 구현 모델 생성, SPIN 모델체커를 이용한 구현 모델 검증과 같이,

공개된 도구를 사용하여 프레임워크 사용자에게 편의성을 제공할 수 있다. 그리고, 본 개시의 모델 기반 보안 프레임워크는 보안 설계 및 검증의 각 방법에 대한 산출물의 추적성이 보장될 수 있다.

- [0059] 도 3은 본 개시의 몇몇 실시예에 따른 보안 시스템 개발 시 보안요구사항에 따라 제대로 설계되고 구현되었는지 확인하기 위한 방법을 설명하기 위한 도면이다. 도 4는 본 개시의 몇몇 실시예에 따른 데이터 흐름도의 일례를 설명하기 위한 도면이다. 도 5는 본 개시의 몇몇 실시예에 따른 공격 트리의 일례를 설명하기 위한 도면이다.
- [0060] 도 3을 참조하면, 본 개시의 컴퓨팅 장치(100)의 프로세서(110)는 위협 모델링 단계(S100), 설계 모델링 단계(S200) 및 구현 모델링 단계(S300)로 이루어진 모델기반 보안 프레임워크를 통해 보안 시스템의 설계 및 구현 모델을 검증할 수 있다.
- [0061] 본 개시의 몇몇 실시예에 따르면, 위협 모델링 단계(S100)는 설계 대상 시스템의 보안 요구사항을 도출 도출하기 위한 단계일 수 있다.
- [0062] 구체적으로, 위협 모델링 단계(S100)는 자산식별 단계(S101), 데이터흐름도 작성 단계(S102), 공격라이브러리 수집 단계(S103), 사전 설정된 보안 속성 적용 단계(S104), 공격 트리 작성 단계(S105) 및 체크리스트 도출 단계(S106)를 포함할 수 있다. 다만, 상술한 세부 단계들은 위협 모델링 단계(S100)를 구현하는데 있어서 필수적인 것은 아니어서, 위협 모델링 단계(S100)는 위에서 열거된 세부 단계들 보다 많거나, 또는 적은 세부 단계들을 가질 수 있다.
- [0063] 보안 기능 요구사항을 도출하기 위해서는 일반적으로 위협 모델링 기법이 사용된다. 위협 모델링은 분석 대상의 자산, 발생 가능한 취약점, 잠재적 위협을 식별하고 열거하며 완화 조치의 우선순위를 지정하는 체계적인 보안성 분석 방법이다. 보안 기능을 개발 초기부터 고려하기 위하여 소프트웨어 개발 생명주기를 적용하는 기업에서는 체계적인 설계 및 검증을 위하여 필요한 언어, 프로세스, 도구가 필요하다. 설계 단계에서 주로 사용되고 있는 위협모델링을 통해 시스템 설계부터 잠재적인 보안 위협들을 식별하여 이를 완화할 수 있도록 보안 기능 요구사항을 도출할 수 있다. 위협모델링은 시스템을 모델링하고, 발생 가능한 위협을 식별하고 공격 시나리오를 작성하여 공격 표면과 수단에 대한 완화방안을 도출하여 요구사항 분석 단계에서 보안 기능에 대한 요구사항 도출이 가능하다.
- [0064] 본 개시의 몇몇 실시예에 따르면, 보안 시스템 개발 생명주기의 위협 모델링 단계(S100)에서 사용되고 있는 기법인 위협 모델링의 사용 단계는 상술한 바와 같이 적어도 하나의 세부 단계들을 포함할 수 있다.
- [0065] 구체적으로, 컴퓨팅 장치(100)의 프로세서(110)는 위협 모델링 단계(S100)의 자산식별 단계(S101)에서 설계 대상(또는 분석 대상)의 오동작을 일으킬 수 있는 데이터, 관련 장치(하드웨어/소프트웨어), 전송 매체, 시스템 등의 자산을 식별할 수 있다. 즉, 프로세서(110)는 설계 대상 시스템의 기능 요구사항에 기초하여, 보안이 요구되는 중요 자산을 인식할 수 있다. 여기서, 중요 자산은 설계 대상 시스템의 특성 및 기능에 대응하여 메모리(130)에 사전 저장되어 있거나 또는, 사용자 입력에 기초하여 결정될 수 있다. 다만, 이에 한정되는 것은 아니다.
- [0066] 또한, 컴퓨팅 장치(100)의 프로세서(110)는 위협 모델링 단계(S100)의 데이터흐름도 작성 단계(S102)에서 시스템의 구조에 대한 분석을 통해 주요 구성요소 기반 데이터 흐름도를 작성할 수 있다. 즉, 프로세서(110)는 설계 대상 시스템의 구조에 기초하여, 주요 구성요소 기반의 데이터 흐름도를 생성할 수 있다.
- [0067] 여기서, 데이터 흐름도는 설계 대상 시스템에서 제공되는 복수의 기능들 각각에 대한 정보 및 복수의 기능들 사이에서 송수신 되는 데이터에 대한 정보를 포함할 수 있다. 다만, 이에 한정되는 것은 아니다.
- [0068] 구체적으로, 도 4를 참조하면, 컴퓨팅 장치(100)의 프로세서(110)가 작성한 데이터 흐름도의 일례가 도시되어 있다.
- [0069] 데이터 흐름도(400)는 설계 대상 시스템(401)을 중심으로, 복수의 기능들에 대한 정보(402) 및 복수의 기능들 사이에서 송수신 되는 데이터에 대한 정보(403)를 포함할 수 있다.
- [0070] 예를 들어, 설계 대상 시스템(401)이 '5G 보안기능 인터페이스'인 경우, 복수의 기능들에 대한 정보(402)는 '키 관리'에 대한 정보, '인증'에 대한 정보, '암호화/복호화'에 대한 정보를 포함할 수 있다. 또한, 복수의 기능들 사이에서 송수신 되는 데이터에 대한 정보(403)는 '키 관리'에 대한 정보와 '5G 보안기능 인터페이스'간 송수신 되는 데이터인 '요청된 키'에 대한 정보 및 '네트워크 키'에 대한 정보를 포함할 수 있다. 다만, 이에 한정되는

것은 아니다.

- [0071] 다시 도 3을 참조하면, 컴퓨팅 장치(100)의 프로세서(110)는 위협 모델링 단계(S100)의 공격라이브러리 수집 단계(S103)에서 다양한 자료(예를 들어, 논문, 컨퍼런스, 기술문서, CVE(Common Vulnerabilities and Exposures) 및 CWE(Common Weakness Enumeration) 등)를 통해 분석 대상 시스템에 대해 알려진 취약점 및 공격 방법을 수집하여 공격 라이브러리를 구축할 수 있다. 즉, 프로세서(110)는 복수의 자료에 기초하여, 설계 대상 시스템의 취약점 및 설계 대상 시스템에 대한 공격 방법과 관련된 공격 라이브러리를 생성할 수 있다.
- [0072] 또한, 컴퓨팅 장치(100)의 프로세서(110)는 위협 모델링 단계(S100)의 사전 설정된 보안 속성 적용 단계(S104)에서, 사전 설정된 보안 속성에 따라 시스템에서 발생할 수 있는 위협을 식별 및 분류할 수 있다. 즉, 프로세서(110)는 사전 설정된 보안 속성에 기초하여, 설계 대상 시스템에서 발생될 수 있는 예상 위협을 인식할 수 있다.
- [0073] 여기서, 사전 설정된 보안 속성은 STRIDE(Spoofing, Tampering, Repudiation, Information disclosure, Denial of service, Elevation of privilege)를 포함할 수 있다. 다만, 이에 한정되는 것은 아니다.
- [0074] 또한, 컴퓨팅 장치(100)의 프로세서(110)는 위협 모델링 단계(S100)의 공격 트리 작성 단계(S105)에서 데이터 흐름도에 사전 설정된 보안 속성을 적용하여 식별된 위협과 공격 라이브러리를 기반으로 공격 트리를 작성할 수 있다. 여기서, 데이터 흐름도는 단계(S102)에서 생성된 데이터 흐름도이고, 공격 라이브러리는 단계(S103)에서 생성된 공격 라이브러리일 수 있다.
- [0075] 여기서, 공격 트리는 최종 공격 목표에 대한 정보를 포함할 수 있다. 또한, 공격 트리의 최상위 노드에 최종 공격 목표가 배치되고, 최상위 노드와 연관된 적어도 하나의 하위 노드에 상기 최종 공격 목표와 관련된 적어도 하나의 위협이 배치될 수 있다. 다만, 이에 한정되는 것은 아니다.
- [0076] 구체적으로, 도 5를 참조하면, 컴퓨팅 장치(100)의 프로세서(110)가 작성한 공격 트리의 일례가 도시되어 있다.
- [0077] 공격 트리(500)는 최상위 노드(501)에 최종 공격 목표가 배치될 수 있다. 예를 들어, 최상위 노드(501)에 포함된 최종 공격 목표는 System Destruction일 수 있다. 다만, 이에 한정되는 것은 아니다.
- [0078] 또한, 공격 트리(500)는 하위 노드(502)에 최종 공격 목표와 관련된 적어도 하나의 위협이 배치될 수 있다. 예를 들어, 하위 노드(502)에 포함된 적어도 하나의 위협은 System Access 및 Rootkits Installation이 포함될 수 있다. 다만, 이에 한정되는 것은 아니다.
- [0079] 다시 도 3을 참조하면, 컴퓨팅 장치(100)의 프로세서(110)는 위협 모델링 단계(S100)의 체크리스트 도출 단계(S106)에서 데이터 흐름도 및 공격 라이브러리를 사용하여 도출된 위협 목록과 작성된 공격 트리를 사용하여 체크리스트를 도출할 수 있다. 또한, 프로세서(110)는 체크리스트를 도출한 후 이를 만족할 수 있는 보안 기능 요구사항을 도출할 수 있다.
- [0080] 본 개시의 몇몇 실시예에 따르면, 설계 모델링 단계(S200)는 보안 요구사항에 기초하여 설계 모델을 생성하고, 설계 모델을 검증하기 위한 단계일 수 있다.
- [0081] 구체적으로, 설계 모델링 단계(S200)는 요구사항 구체화를 통한 설계 모델 생성 단계(S201) 및 설계 모델에 대한 정확성 검증 단계(S202)를 포함할 수 있다. 다만, 상술한 세부 단계들은 설계 모델링 단계(S200)를 구현하는데 있어서 필수적인 것은 아니어서, 설계 모델링 단계(S200)는 위에서 열거된 세부 단계들 보다 많거나, 또는 적은 세부 단계들을 가질 수 있다.
- [0082] 보안 기능에 대한 설계 이후에는 설계 모델링을 통해 도출된 보안 기능 요구사항을 통해 보안 시스템을 개발하고 테스트하는 단계를 수행해야 한다.
- [0083] 본 개시의 몇몇 실시예에 따르면, 보안 시스템 개발 생명주기의 설계 모델링 단계(S200)에서 모델에 대한 정형 명세 및 검증에 대한 프레임워크 사용 단계는 상술한 바와 같이 적어도 하나의 세부 단계들을 포함할 수 있다.
- [0084] 구체적으로, 컴퓨팅 장치(100)의 프로세서(110)는 설계 모델링 단계(S200)의 요구사항 구체화를 통한 설계 모델 생성 단계(S201)에서 표준화된 모델링 언어를 이용하여, 보안 요구사항에 대응하는 설계 모델을 생성할 수 있다. 여기서, 표준화된 모델링 언어는 UML(Unified Modeling Language)일 수 있다.
- [0085] 즉, 프로세서(110)는 보안 기능 요구사항을 이용하여 UML 다이어그램을 통해 보안 시스템 구축을 위한 설계 모델을 생성할 수 있다.

- [0086] 다음으로, 컴퓨팅 장치(100)의 프로세서(110)는 설계 모델링 단계(S200)의 설계 모델에 대한 정확성 검증 단계(S202)에서 형식 언어를 이용하여 생성된 설계 모델과 보안 요구사항 간 모순이 없는지에 대한 정확성을 검증할 수 있다. 여기서, 형식언어는 OCL(Object Constraint Language)일 수 있다.
- [0087] 즉, 프로세서(110)는 설계 모델을 대상으로 OCL을 통해 설계 모델에서의 위협 식별, 일관성 검사, 보안 정책에 대한 검증을 수행할 수 있다.
- [0088] 본 개시의 몇몇 실시예에 따르면, 구현 모델링 단계(S300)는 설계 모델에 기초하여 구현 모델을 생성하고, 상기 구현 모델을 검증하기 위한 단계일 수 있다.
- [0089] 구체적으로, 구현 모델링 단계(S300)는 설계 모델에 대한 소스코드 생성 단계(S301), 소스코드 정형 명세 단계(S302) 및 모델 체커를 통한 구현 모델 정형 검증 단계(S303)를 포함할 수 있다. 다만, 상술한 세부 단계들은 구현 모델링 단계(S300)를 구현하는데 있어서 필수적인 것은 아니어서, 구현 모델링 단계(S300)는 위에서 열거된 세부 단계들 보다 많거나, 또는 적은 세부 단계들을 가질 수 있다.
- [0090] 정확한 설계를 통해 보안 기능을 설계하였더라도 구현 단계에서 정확하게 반영하지 못한다면 예상하지 못한 결함이 발생할 수 있다. 설계에 대한 검증 이후 검증된 설계 모델에 따라 시스템이 제대로 구현되었는지 명세하고 검증하는 것도 중요하다. 이때 설계 모델을 이용하여 소스코드를 생성하고 Promela 언어로 변환한 뒤 SPIN 모델 체커를 통해 구현 모델에 대한 검증을 수행한다.
- [0091] 본 개시의 몇몇 실시예에 따르면, 보안 시스템 개발 생명주기의 구현 모델링 단계(S300)에서 모델에 대한 정형 명세 및 검증에 대한 프레임워크 사용 단계는 상술한 바와 같이, 적어도 하나의 세부 단계들을 포함할 수 있다.
- [0092] 구체적으로, 컴퓨팅 장치(100)의 프로세서(110)는 구현 모델링 단계(S300)의 설계 모델에 대한 소스코드 생성 단계(S301)에서 설계 모델에 기초하여, 소스 코드를 생성할 수 있다.
- [0093] 또한, 컴퓨팅 장치(100)의 프로세서(110)는 구현 모델링 단계(S300)의 소스코드 정형 명세 단계(S302)에서 소스 코드를 검증 모델링 언어로 변환하여, 구현 모델을 생성할 수 있다. 여기서, 검증 모델링 언어는 Promela(Process or Protocol Meta Language) 언어일 수 있다. 다만, 이에 한정되는 것은 아니다.
- [0094] 마지막으로, 컴퓨팅 장치(100)의 프로세서(110)는 구현 모델링 단계(S300)의 모델 체커를 통한 구현 모델 정형 검증 단계(S303)에서 모델 체커를 이용하여 구현 모델에 대한 정형 검증을 수행할 수 있다. 여기서, 모델 체커는 SPIN(Simple Promela Interpreter) 모델 체커일 수 있다. 다만, 이에 한정되는 것은 아니다.
- [0095] 본 개시의 몇몇 실시예에 따르면, 상술한 설계 모델링 단계(S200) 및 구현 모델링 단계(S300)에서 UML과 OCL, Promela, SPIN 모델체커와 같은 공개된 도구를 사용하여 프레임워크 사용자에게 편의성을 제공할 수 있다.
- [0096] 또한, 본 개시는 위협 모델링과 정형 명세 및 검증 기법을 사용한 모델 기반의 보안 프레임워크를 통해 체계적으로 보안 시스템에 대한 설계 및 검증을 통해 개발 초기에 보안 기능을 고려할 수 있다. 이 경우, 제품에 대한 보안성이 향상되고, 개발 비용을 절감할 수 있다. 구체적으로, 도 3에 도시된 단계별 결함 수정에 발생하는 비용 연구 결과에 대한 그래프를 참조하면, 보안 시스템 개발 생명주기 초기에 문제를 식별하여 수정하는 경우에는 개발 완료 후에 수정할 때 보다 약 100배의 비용을 절감할 수 있다.
- [0097] 또한, 본 개시는 정형 명세 및 검증 기법에 대한 체계적인 프레임워크를 통해 신뢰할 수 있는 보안 시스템 개발에 도움을 줄 수 있다.
- [0099] 도 6은 본 개시내용의 실시예들이 구현될 수 있는 예시적인 컴퓨팅 환경에 대한 일반적인 개략도를 도시한다.
- [0100] 본 개시내용이 일반적으로 하나 이상의 컴퓨터 상에서 실행될 수 있는 컴퓨터 실행가능 명령어와 관련하여 진술되었지만, 당업자라면 본 개시내용 기타 프로그램 모듈들과 결합되어 및/또는 하드웨어와 소프트웨어의 조합으로서 구현될 수 있다는 것을 잘 알 것이다.
- [0101] 일반적으로, 본 명세서에서의 모듈은 특정의 태스크를 수행하거나 특정의 추상 데이터 유형을 구현하는 루틴, 프로시저, 프로그램, 컴포넌트, 데이터 구조, 기타 등등을 포함한다. 또한, 당업자라면 본 개시의 방법이 단일-프로세서 또는 멀티프로세서 컴퓨터 시스템, 미니컴퓨터, 메인프레임 컴퓨터는 물론 퍼스널 컴퓨터, 핸드헬드 컴퓨팅 장치, 마이크로프로세서-기반 또는 프로그램가능 가전 제품, 기타 등등(이들 각각은 하나 이상의 연관된 장치와 연결되어 동작할 수 있음)을 비롯한 다른 컴퓨터 시스템 구성으로 실시될 수 있다는 것을 잘 알 것이다.

- [0102] 본 개시의 설명된 실시예들은 또한 어떤 태스크들이 통신 네트워크를 통해 연결되어 있는 원격 처리 장치들에 의해 수행되는 분산 컴퓨팅 환경에서 실시될 수 있다. 분산 컴퓨팅 환경에서, 프로그램 모듈은 로컬 및 원격 메모리 저장 장치 둘다에 위치할 수 있다.
- [0103] 컴퓨터는 통상적으로 다양한 컴퓨터 판독가능 매체를 포함한다. 컴퓨터에 의해 액세스 가능한 매체로서, 휘발성 및 비휘발성 매체, 일시적(transitory) 및 비일시적(non-transitory) 매체, 이동식 및 비-이동식 매체를 포함한다. 제한이 아닌 예로서, 컴퓨터 판독가능 매체는 컴퓨터 판독가능 저장 매체 및 컴퓨터 판독가능 전송 매체를 포함할 수 있다.
- [0104] 컴퓨터 판독가능 저장 매체는 컴퓨터 판독가능 명령어, 데이터 구조, 프로그램 모듈 또는 기타 데이터와 같은 정보를 저장하는 임의의 방법 또는 기술로 구현되는 휘발성 및 비휘발성 매체, 일시적 및 비-일시적 매체, 이동식 및 비이동식 매체를 포함한다. 컴퓨터 판독가능 저장 매체는 RAM, ROM, EEPROM, 플래시 메모리 또는 기타 메모리 기술, CD-ROM, DVD(digital video disk) 또는 기타 광 디스크 저장 장치, 자기 카세트, 자기 테이프, 자기 디스크 저장 장치 또는 기타 자기 저장 장치, 또는 컴퓨터에 의해 액세스될 수 있고 원하는 정보를 저장하는데 사용될 수 있는 임의의 기타 매체를 포함하지만, 이에 한정되지 않는다.
- [0105] 컴퓨터 판독가능 전송 매체는 통상적으로 반송파(carrier wave) 또는 기타 전송 메커니즘(transport mechanism)과 같은 피변조 데이터 신호(modulated data signal)에 컴퓨터 판독가능 명령어, 데이터 구조, 프로그램 모듈 또는 기타 데이터들을 구현하고 모든 정보 전달 매체를 포함한다. 피변조 데이터 신호라는 용어는 신호 내에 정보를 인코딩하도록 그 신호의 특성들 중 하나 이상을 설정 또는 변경시킨 신호를 의미한다. 제한이 아닌 예로서, 컴퓨터 판독가능 전송 매체는 유선 네트워크 또는 직접 배선 접속(direct-wired connection)과 같은 유선 매체, 그리고 음향, RF, 적외선, 기타 무선 매체와 같은 무선 매체를 포함한다. 상술된 매체들 중 임의의 것의 조합도 역시 컴퓨터 판독가능 전송 매체의 범위 안에 포함되는 것으로 한다.
- [0106] 컴퓨터(1102)를 포함하는 본 개시의 여러가지 측면들을 구현하는 예시적인 환경(1100)이 나타내어져 있으며, 컴퓨터(1102)는 처리 장치(1104), 시스템 메모리(1106) 및 시스템 버스(1108)를 포함한다. 시스템 버스(1108)는 시스템 메모리(1106)(이에 한정되지 않음)를 비롯한 시스템 컴포넌트들을 처리 장치(1104)에 연결시킨다. 처리 장치(1104)는 다양한 상용 프로세서들 중 임의의 프로세서일 수 있다. 듀얼 프로세서 및 기타 멀티프로세서 아키텍처도 역시 처리 장치(1104)로서 이용될 수 있다.
- [0107] 시스템 버스(1108)는 메모리 버스, 주변장치 버스, 및 다양한 상용 버스 아키텍처 중 임의의 것을 사용하는 로컬 버스에 추가적으로 상호 연결될 수 있는 몇 가지 유형의 버스 구조 중 임의의 것일 수 있다. 시스템 메모리(1106)는 판독 전용 메모리(ROM)(1110) 및 랜덤 액세스 메모리(RAM)(1112)를 포함한다. 기본 입/출력 시스템(BIOS)은 ROM, EPROM, EEPROM 등의 비휘발성 메모리(1110)에 저장되며, 이 BIOS는 시동 중과 같은 때에 컴퓨터(1102) 내의 구성요소들 간에 정보를 전송하는 일을 돕는 기본적인 루틴을 포함한다. RAM(1112)은 또한 데이터를 캐싱하기 위한 정적 RAM 등의 고속 RAM을 포함할 수 있다.
- [0108] 컴퓨터(1102)는 또한 내장형 하드 디스크 드라이브(HDD)(1114)(예를 들어, EIDE, SATA)–이 내장형 하드 디스크 드라이브(1114)는 또한 적당한 새시(도시 생략) 내에서 외장형 용도로 구성될 수 있음–, 자기 플로피 디스크 드라이브(FDD)(1116)(예를 들어, 이동식 디스켓(1118)으로부터 판독을 하거나 그에 기록을 하기 위한 것임), 및 광 디스크 드라이브(1120)(예를 들어, CD-ROM 디스크(1122)를 판독하거나 DVD 등의 기타 고용량 광 매체로부터 판독을 하거나 그에 기록을 하기 위한 것임)를 포함한다. 하드 디스크 드라이브(1114), 자기 디스크 드라이브(1116) 및 광 디스크 드라이브(1120)는 각각 하드 디스크 드라이브 인터페이스(1124), 자기 디스크 드라이브 인터페이스(1126) 및 광 드라이브 인터페이스(1128)에 의해 시스템 버스(1108)에 연결될 수 있다. 외장형 드라이브 구현을 위한 인터페이스(1124)는 예를 들어, USB(Universal Serial Bus) 및 IEEE 1394 인터페이스 기술 중 적어도 하나 또는 그 둘 다를 포함한다.
- [0109] 이들 드라이브 및 그와 연관된 컴퓨터 판독가능 매체는 데이터, 데이터 구조, 컴퓨터 실행가능 명령어, 기타 등등의 비휘발성 저장을 제공한다. 컴퓨터(1102)의 경우, 드라이브 및 매체는 임의의 데이터를 적당한 디지털 형식으로 저장하는 것에 대응한다. 상기에서의 컴퓨터 판독가능 저장 매체에 대한 설명이 HDD, 이동식 자기 디스크, 및 CD 또는 DVD 등의 이동식 광 매체를 언급하고 있지만, 당업자라면 zip 드라이브(zip drive), 자기 카세트, 플래시 메모리 카드, 카트리지, 기타 등등의 컴퓨터에 의해 판독가능한 다른 유형의 저장 매체도 역시 예시적인 운영 환경에서 사용될 수 있으며 또 임의의 이러한 매체가 본 개시의 방법들을 수행하기 위한 컴퓨터 실행가능 명령어를 포함할 수 있다는 것을 잘 알 것이다.

- [0110] 운영 체제(1130), 하나 이상의 애플리케이션 프로그램(1132), 기타 프로그램 모듈(1134) 및 프로그램 데이터(1136)를 비롯한 다수의 프로그램 모듈이 드라이브 및 RAM(1112)에 저장될 수 있다. 운영 체제, 애플리케이션, 모듈 및/또는 데이터의 전부 또는 그 일부가 또한 RAM(1112)에 캐싱될 수 있다. 본 개시가 여러가지 상업적으로 이용가능한 운영 체제 또는 운영 체제들의 조합에서 구현될 수 있다는 것을 잘 알 것이다.
- [0111] 사용자는 하나 이상의 유선/무선 입력 장치, 예를 들어, 키보드(1138) 및 마우스(1140) 등의 포인팅 장치를 통해 컴퓨터(1102)에 명령 및 정보를 입력할 수 있다. 기타 입력 장치(도시 생략)로는 마이크, IR 리모콘, 조이스틱, 게임 패드, 스타일러스 펜, 터치 스크린, 기타 등등이 있을 수 있다. 이들 및 기타 입력 장치가 종종 시스템 버스(1108)에 연결되어 있는 입력 장치 인터페이스(1142)를 통해 처리 장치(1104)에 연결되지만, 병렬 포트, IEEE 1394 직렬 포트, 게임 포트, USB 포트, IR 인터페이스, 기타 등등의 기타 인터페이스에 의해 연결될 수 있다.
- [0112] 모니터(1144) 또는 다른 유형의 디스플레이 장치도 역시 비디오 어댑터(1146) 등의 인터페이스를 통해 시스템 버스(1108)에 연결된다. 모니터(1144)에 부가하여, 컴퓨터는 일반적으로 스피커, 프린터, 기타 등등의 기타 주변 출력 장치(도시 생략)를 포함한다.
- [0113] 컴퓨터(1102)는 유선 및/또는 무선 통신을 통한 원격 컴퓨터(들)(1148) 등의 하나 이상의 원격 컴퓨터로의 논리적 연결을 사용하여 네트워크화된 환경에서 동작할 수 있다. 원격 컴퓨터(들)(1148)는 워크스테이션, 서버 컴퓨터, 라우터, 퍼스널 컴퓨터, 휴대용 컴퓨터, 마이크로프로세서-기반 오락 기기, 피어 장치 또는 기타 통상의 네트워크 노드일 수 있으며, 일반적으로 컴퓨터(1102)에 대해 기술된 구성요소들 중 다수 또는 그 전부를 포함하지만, 간략함을 위해, 메모리 저장 장치(1150)만이 도시되어 있다. 도시되어 있는 논리적 연결은 근거리 통신망(LAN)(1152) 및/또는 더 큰 네트워크, 예를 들어, 원격 통신망(WAN)(1154)에의 유선/무선 연결을 포함한다. 이러한 LAN 및 WAN 네트워크 환경은 사무실 및 회사에서 일반적인 것이며, 인트라넷 등의 전사적 컴퓨터 네트워크(enterprise-wide computer network)를 용이하게 해주며, 이들 모두는 전세계 컴퓨터 네트워크, 예를 들어, 인터넷에 연결될 수 있다.
- [0114] LAN 네트워킹 환경에서 사용될 때, 컴퓨터(1102)는 유선 및/또는 무선 통신 네트워크 인터페이스 또는 어댑터(1156)를 통해 로컬 네트워크(1152)에 연결된다. 어댑터(1156)는 LAN(1152)에의 유선 또는 무선 통신을 용이하게 해줄 수 있으며, 이 LAN(1152)은 또한 무선 어댑터(1156)와 통신하기 위해 그에 설치되어 있는 무선 액세스 포인트를 포함하고 있다. WAN 네트워킹 환경에서 사용될 때, 컴퓨터(1102)는 모뎀(1158)을 포함할 수 있거나, WAN(1154) 상의 통신 서버에 연결되거나, 또는 인터넷을 통하는 등, WAN(1154)을 통해 통신을 설정하는 기타 수단을 갖는다. 내장형 또는 외장형 및 유선 또는 무선 장치일 수 있는 모뎀(1158)은 직렬 포트 인터페이스(1142)를 통해 시스템 버스(1108)에 연결된다. 네트워크화된 환경에서, 컴퓨터(1102)에 대해 설명된 프로그램 모듈들 또는 그의 일부가 원격 메모리/저장 장치(1150)에 저장될 수 있다. 도시된 네트워크 연결이 예시적인 것이며 컴퓨터들 사이에 통신 링크를 설정하는 기타 수단이 사용될 수 있다는 것을 잘 알 것이다.
- [0115] 컴퓨터(1102)는 무선 통신으로 배치되어 동작하는 임의의 무선 장치 또는 개체, 예를 들어, 프린터, 스캐너, 데스크톱 및/또는 휴대용 컴퓨터, PDA(portable data assistant), 통신 위성, 무선 검출가능 태그와 연관된 임의의 장비 또는 장소, 및 전화와 통신을 하는 동작을 한다. 이것은 적어도 Wi-Fi 및 블루투스 무선 기술을 포함한다. 따라서, 통신은 종래의 네트워크에서와 같이 미리 정의된 구조이거나 단순하게 적어도 2개의 장치 사이의 애드혹 통신(ad hoc communication)일 수 있다.
- [0116] Wi-Fi(Wireless Fidelity)는 유선 없이도 인터넷 등으로의 연결을 가능하게 해준다. Wi-Fi는 이러한 장치, 예를 들어, 컴퓨터가 실내에서 및 실외에서, 즉 기지국의 통화권 내의 아무 곳에서나 데이터를 전송 및 수신할 수 있게 해주는 셀 전화와 같은 무선 기술이다. Wi-Fi 네트워크는 안전하고 신뢰성 있으며 고속인 무선 연결을 제공하기 위해 IEEE 802.11(a,b,g, 기타)이라고 하는 무선 기술을 사용한다. 컴퓨터를 서로에, 인터넷에 및 유선 네트워크(IEEE 802.3 또는 이더넷을 사용함)에 연결시키기 위해 Wi-Fi가 사용될 수 있다. Wi-Fi 네트워크는 비인가 2.4 및 5 GHz 무선 대역에서, 예를 들어, 11Mbps(802.11a) 또는 54 Mbps(802.11b) 데이터 레이트로 동작하거나, 양 대역(듀얼 대역)을 포함하는 제품에서 동작할 수 있다.
- [0117] 본 개시의 기술 분야에서 통상의 지식을 가진 자는 여기에 개시된 실시예들과 관련하여 설명된 다양한 예시적인 논리 블록들, 모듈들, 프로세서들, 수단들, 회로들 및 알고리즘 단계들이 전자 하드웨어, (편의를 위해, 여기에서 "소프트웨어"로 지칭되는) 다양한 형태들의 프로그램 또는 설계 코드 또는 이들 모두의 결합에 의해 구현될 수 있다는 것을 이해할 것이다. 하드웨어 및 소프트웨어의 이러한 상호 호환성을 명확하게 설명하기 위해, 다양한 예시적인 컴포넌트들, 블록들, 모듈들, 회로들 및 단계들이 이들의 기능과 관련하여 위에서 일반적으로 설명

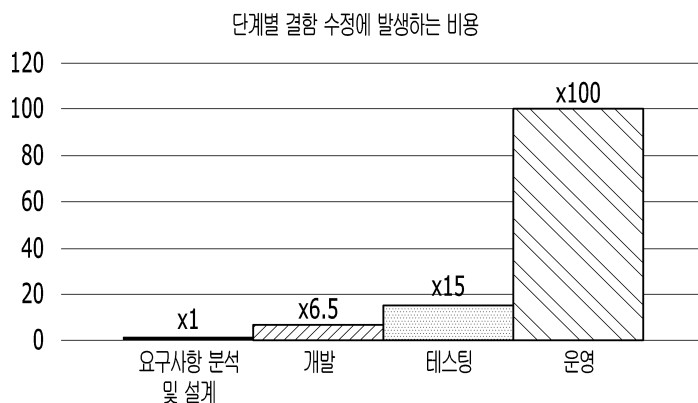
되었다. 이러한 기능이 하드웨어 또는 소프트웨어로서 구현되는지 여부는 특정한 애플리케이션 및 전체 시스템에 대하여 부과되는 설계 제약들에 따라 좌우된다. 본 개시의 기술 분야에서 통상의 지식을 가진 자는 각각의 특정한 애플리케이션에 대하여 다양한 방식으로 설명된 기능을 구현할 수 있으나, 이러한 구현 결정들은 본 개시의 범위를 벗어나는 것으로 해석되어서는 안 될 것이다.

[0118] 여기서 제시된 다양한 실시예들은 방법, 장치, 또는 표준 프로그래밍 및/또는 엔지니어링 기술을 사용한 제조 물품(article)으로 구현될 수 있다. 용어 "제조 물품"은 임의의 컴퓨터-판독가능 장치로부터 액세스 가능한 컴퓨터 프로그램 또는 매체(media)를 포함한다. 예를 들어, 컴퓨터-판독가능 저장 매체는 자기 저장 장치(예를 들면, 하드 디스크, 플로피 디스크, 자기 스트림, 등), 광학 디스크(예를 들면, CD, DVD, 등), 스마트 카드, 및 플래쉬 메모리 장치(예를 들면, EEPROM, 카드, 스틱, 키 드라이브, 등)를 포함하지만, 이들로 제한되는 것은 아니다. 용어 "기계-판독가능 매체"는 명령(들) 및/또는 데이터를 저장, 보유, 및/또는 전달할 수 있는 무선 채널 및 다양한 다른 매체를 포함하지만, 이들로 제한되는 것은 아니다.

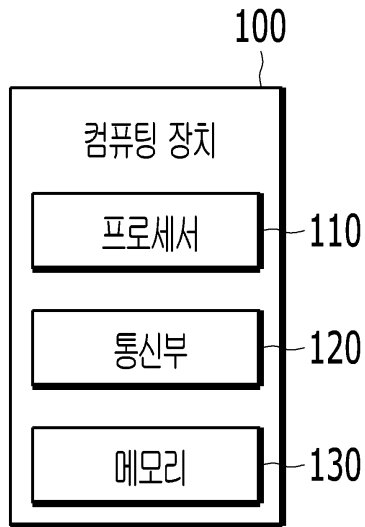
[0120] 제시된 실시예들에 대한 설명은 임의의 본 개시의 기술 분야에서 통상의 지식을 가진 자가 본 개시를 이용하거나 또는 실시할 수 있도록 제공된다. 이러한 실시예들에 대한 다양한 변형들은 본 개시의 기술 분야에서 통상의 지식을 가진 자에게 명백할 것이며, 여기에 정의된 일반적인 원리들은 본 개시의 범위를 벗어남이 없이 다른 실시예들에 적용될 수 있다. 그리하여, 본 개시는 여기에 제시된 실시예들로 한정되는 것이 아니라, 여기에 제시된 원리들 및 신규한 특징들과 일관되는 최광의의 범위에서 해석되어야 할 것이다.

도면

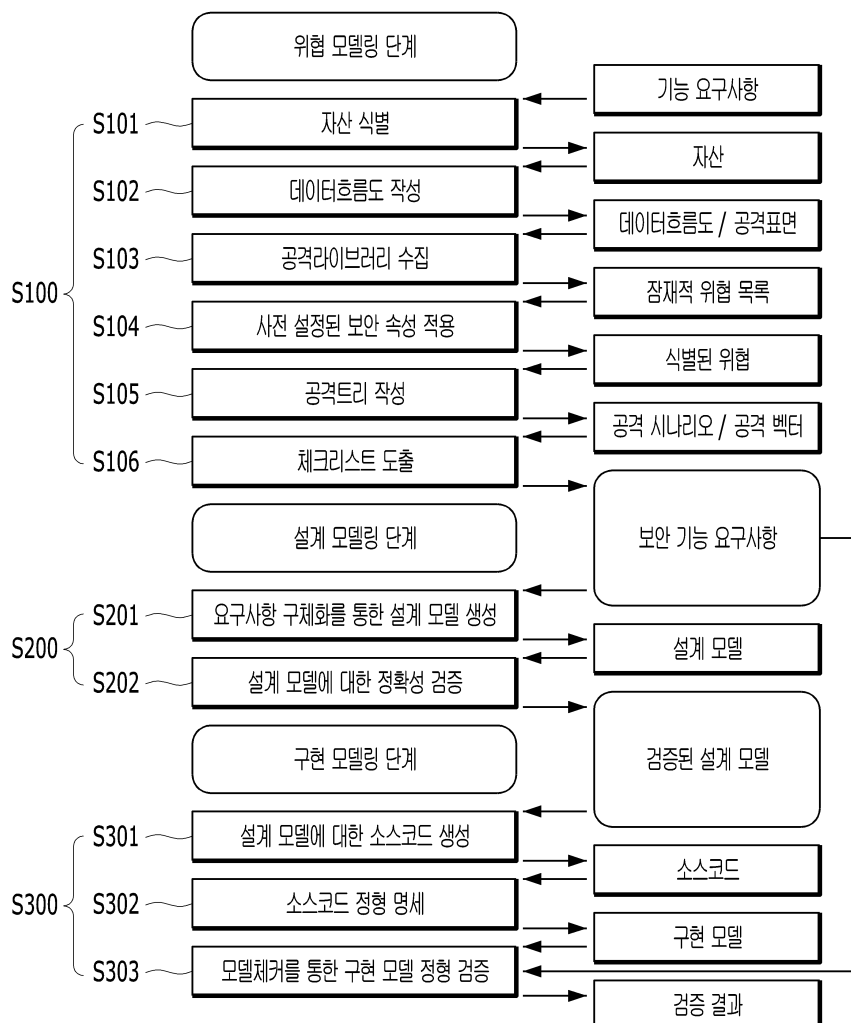
도면1



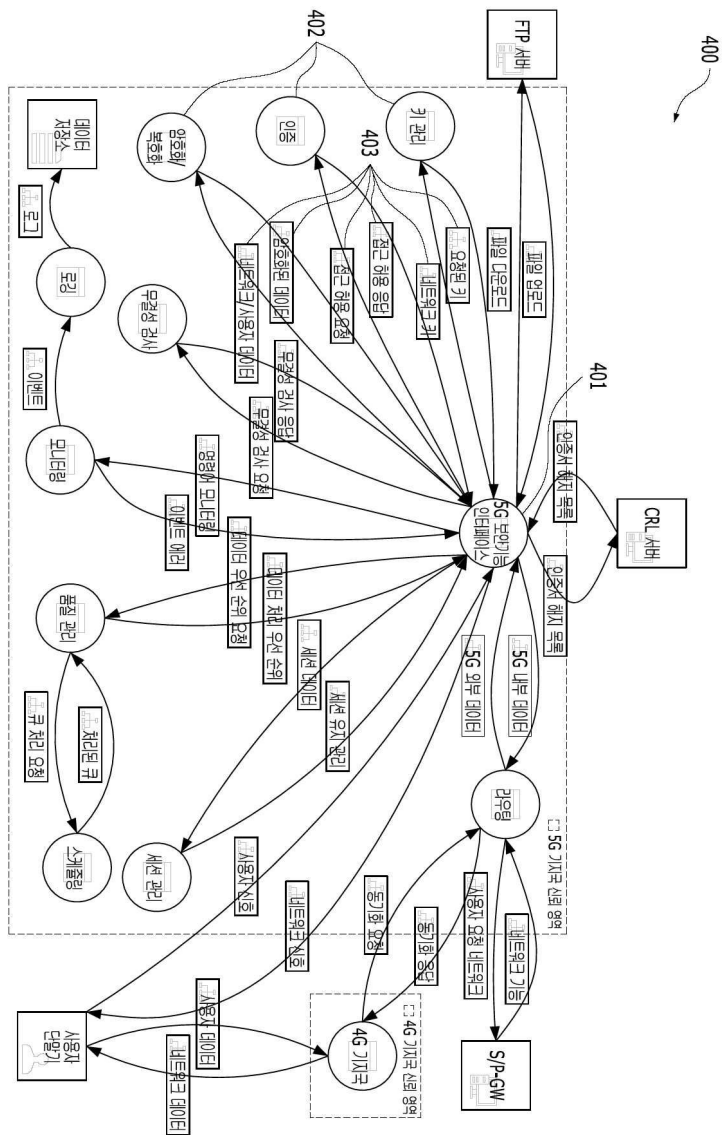
도면2



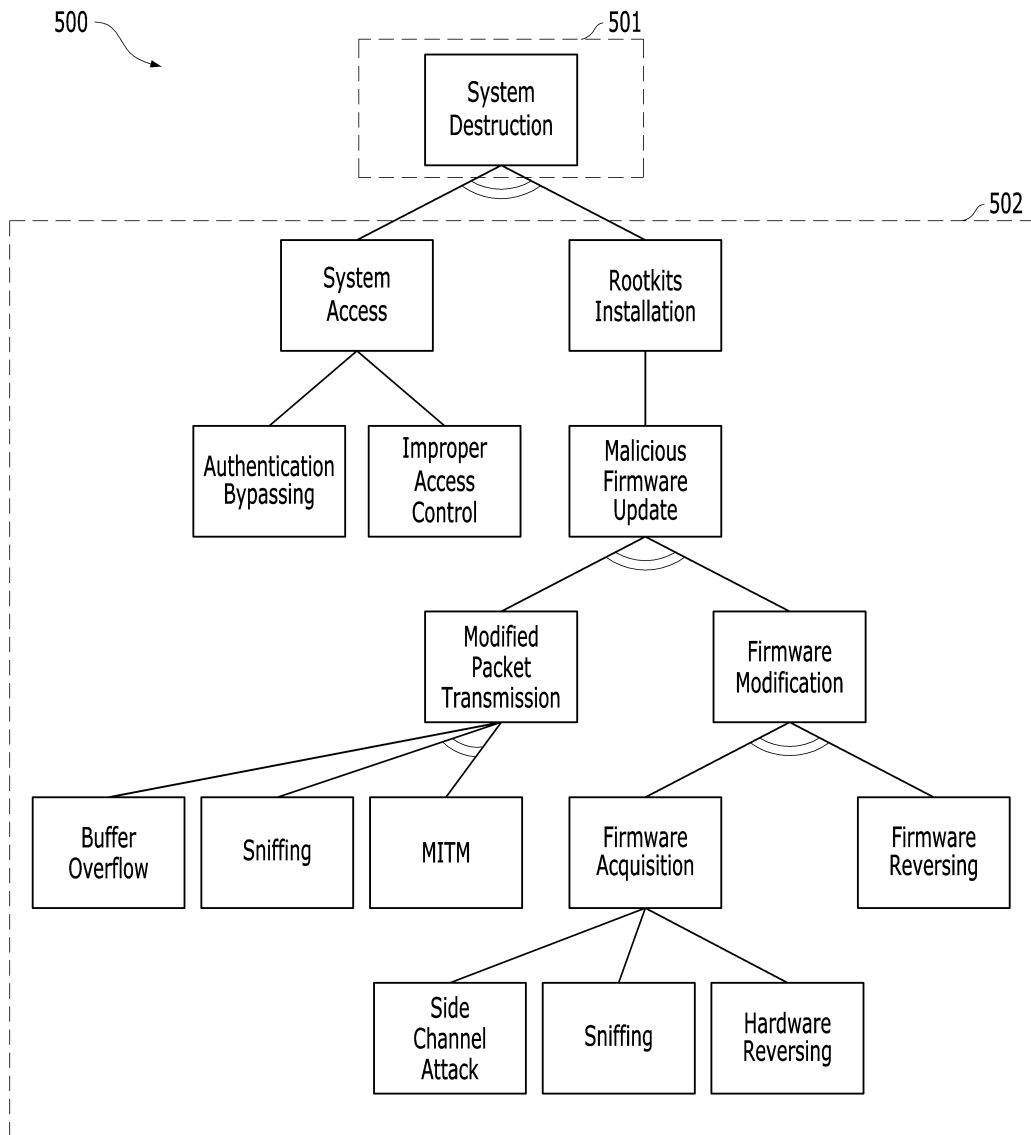
도면3



도면4



도면5



도면6

