(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2022년06월22일
(11) 등록번호 10-2410552
(24) 등록일자 2022년06월14일

(51) 국제특허분류(Int. Cl.)
H04L 9/40 (2022.01) *H04L 67/06* (2022.01)
(52) CPC특허분류
H04L 63/10 (2013.01)
H04L 63/0236 (2013.01)
(21) 출원번호 10-2022-0011596
(22) 출원일자 2022년01월26일
심사청구일자 2022년01월26일
(56) 선행기술조사문헌
KR102309115 B1
(뒷면에 계속)
전체 청구항 수 : 총 18 항

(73) 특허권자
프라이빗테크놀로지 주식회사
서울특별시 금천구 벚꽃로 298 ,1303호(가산동,대륭포스트타워6차)
(72) 발명자
김영랑
서울특별시 금천구 벚꽃로 298 대륭포스트타워6차 1303호
(74) 대리인
특허법인태평양

심사관 : 이준석

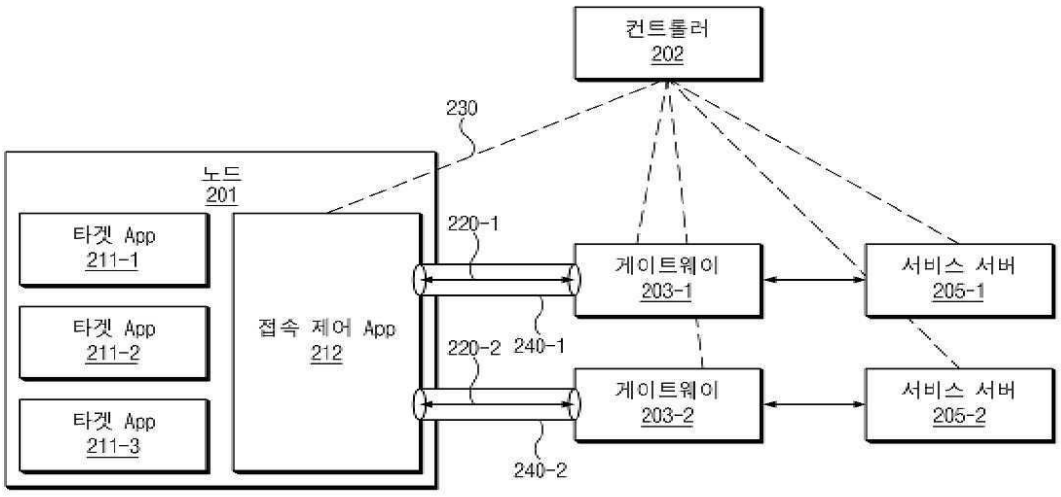
(54) 발명의 명칭 **애플리케이션의 파일 송신 및 수신을 제어하기 위한 시스템 및 그에 관한 방법**

(57) 요 약

본 문서에서 개시되는 일 실시 예에 따른 노드는, 통신 회로, 상기 통신 회로와 작동적으로 연결되는 프로세서, 및 상기 프로세서와 작동적으로 연결되고, 접속 제어 애플리케이션 및 타겟 애플리케이션을 저장하는 메모리를 포함하고, 상기 메모리는, 상기 프로세서에 의해서 실행될 때 상기 노드가, 상기 접속 제어 애플리케이션을

(뒷면에 계속)

대 표 도 - 도1



통해, 외부 서버에게 서비스 서버에 대한 네트워크 접속을 요청하되, 상기 네트워크 접속 요청은 상기 서비스 서버의 네트워크에 접속하려는 상기 타겟 애플리케이션의 식별 정보, 상기 서비스 서버의 IP(Internet Protocol) 및 포트를 포함하고, 상기 서비스 서버의 네트워크에 접속이 가능한 경우, 상기 타겟 애플리케이션의 파일 IO(input output)의 허용 여부를 나타내는 인가된 파일 IO 정보를 포함하는 데이터 플로우를 수신하고, 상기 접속 제어 애플리케이션을 통해, 상기 타겟 애플리케이션과 관련된 파일 IO가 존재하는지를 확인하고, 상기 인가된 파일 IO 정보에 기초하여, 상기 관련된 파일 IO의 동작을 관리하도록 하는 명령어들을 저장할 수 있다.

(52) CPC특허분류
H04L 63/029 (2013.01)
H04L 63/10 (2013.01)
H04L 63/20 (2013.01)
H04L 67/06 (2022.05)

(56) 선행기술조사문헌
KR102250505 B1
KR102204705 B1
KR1020210045917 A
KR1020140006050 A
KR1020070014139 A

명 세 서

청구범위

청구항 1

노드에 있어서,

통신 회로;

상기 통신 회로와 작동적으로 연결되는 프로세서; 및

상기 프로세서와 작동적으로 연결되고, 접속 제어 애플리케이션 및 타겟 애플리케이션을 저장하는 메모리를 포함하고, 상기 메모리는, 상기 프로세서에 의해서 실행될 때 상기 노드가,

상기 접속 제어 애플리케이션을 통해, 외부 서버에게 서비스 서버에 대한 네트워크 접속을 요청하되, 상기 네트워크 접속 요청은 상기 서비스 서버의 네트워크에 접속하려는 상기 타겟 애플리케이션의 식별 정보, 상기 서비스 서버의 IP(Internet Protocol) 및 포트를 포함하고,

상기 서비스 서버의 네트워크에 접속이 가능한 경우, 상기 타겟 애플리케이션의 파일 IO(input output)의 허용 여부를 나타내는 인가된 파일 IO 정보를 포함하는 데이터 플로우를 수신하고,

상기 접속 제어 애플리케이션을 통해, 상기 타겟 애플리케이션과 관련된 파일 IO가 존재하는지를 확인하고, 상기 인가된 파일 IO 정보에 기초하여, 상기 관련된 파일 IO의 동작을 관리하도록 하는 명령어들을 저장하는, 노드.

청구항 2

청구항 1에 있어서, 상기 파일 IO의 동작을 관리하는 경우, 상기 명령어들은 상기 노드가,

상기 접속 제어 애플리케이션을 통해, 상기 인가된 파일 IO 정보에 기초하여 상기 관련된 파일 IO의 허용 여부를 확인하고,

상기 관련된 파일 IO가 허용되지 않는 경우, 상기 접속 제어 애플리케이션을 통해, 상기 관련된 파일 IO를 차단하도록 하는, 노드.

청구항 3

청구항 1에 있어서, 상기 관련된 파일 IO의 동작을 관리하는 경우, 상기 명령어들은 상기 노드가,

상기 접속 제어 애플리케이션을 통해, 상기 관련된 파일 IO의 종류를 식별하고,

상기 관련된 파일 IO의 상기 식별된 종류가 쓰기와 관련된 경우, 상기 접속 제어 애플리케이션을 통해, 상기 타겟 애플리케이션의 파일 쓰기를 제어하고,

상기 관련된 파일 IO의 상기 식별된 종류가 읽기와 관련된 경우, 상기 접속 제어 애플리케이션을 통해, 상기 타겟 애플리케이션의 파일 읽기를 제어하도록 하는, 노드.

청구항 4

청구항 1에 있어서, 상기 명령어들은 상기 노드가,

상기 접속 제어 애플리케이션을 통해, 상기 타겟 애플리케이션의 파일 IO에 대한 접근이 확인되면, 상기 타겟 애플리케이션에 할당된 데이터 플로우가 존재하는지를 확인하고,

상기 타겟 애플리케이션에 할당된 데이터 플로우가 존재하는 경우, 상기 접속 제어 애플리케이션을 통해, 상기 타겟 애플리케이션에 대한 상기 인가된 파일 IO 정보를 확인하고,

상기 타겟 애플리케이션이 접근하는 상기 파일 IO가 허용되지 않는 경우, 상기 접속 제어 애플리케이션을 통해, 상기 타겟 애플리케이션이 접근하는 상기 파일 IO를 차단하도록 하는, 노드.

청구항 5

청구항 4에 있어서, 상기 명령어들은 상기 노드가,

상기 타겟 애플리케이션에 할당된 데이터 플로우가 존재하지 않는 경우, 상기 접속 제어 애플리케이션을 통해, 상기 타겟 애플리케이션이 접근하는 상기 파일 IO와 관련된 파일 정보를 파일 IO 테이블에 저장하도록 하는, 노드.

청구항 6

청구항 1에 있어서, 상기 명령어들은 상기 노드가,

상기 접속 제어 애플리케이션을 통해, 상기 타겟 애플리케이션의 데이터 패킷 전송 요청을 확인하고,

상기 접속 제어 애플리케이션을 통해, 파일 IO 테이블에서 상기 타겟 애플리케이션이 접근하는 파일 IO의 대상 파일을 확인하고,

상기 접속 제어 애플리케이션을 통해, 상기 대상 파일의 파일 정보를 확인하고,

상기 접속 제어 애플리케이션을 통해, 상기 전송 요청된 데이터 패킷이 상기 파일 정보에 포함된 식별 정보를 포함하는지를 확인하고,

상기 데이터 패킷이 상기 식별 정보를 포함하는 경우, 상기 데이터 패킷을 드롭하고,

상기 데이터 패킷이 상기 식별 정보를 포함하지 않는 경우, 상기 데이터 패킷을 상기 서비스 서버로 전송하도록 하는, 노드.

청구항 7

청구항 1에 있어서, 상기 명령어들은 상기 노드가,

상기 접속 제어 애플리케이션을 통해, 상기 외부 서버에게 외부 서버 접속 요청을 송신하고,

상기 접속 제어 애플리케이션을 통해, 상기 외부 서버로부터 화이트 리스트를 수신하고,

상기 타겟 애플리케이션이 상기 화이트 리스트에 포함된 경우, 상기 접속 제어 애플리케이션을 통해, 상기 타겟 애플리케이션의 무결성 및 안정성 여부를 검사하고,

상기 접속 제어 애플리케이션을 통해, 상기 외부 서버에게 상기 타겟 애플리케이션의 검사 결과를 송신하고,

상기 접속 제어 애플리케이션을 통해, 상기 외부 서버로부터 상기 데이터 플로우를 수신하도록 하는, 노드.

청구항 8

청구항 1에 있어서, 상기 명령어들은 상기 노드가,

상기 접속 제어 애플리케이션을 통해, 상기 타겟 애플리케이션의 종료를 확인하고,

상기 접속 제어 애플리케이션을 통해, 데이터 플로우 테이블에서 상기 타겟 애플리케이션의 식별 정보에 대응하는 데이터 플로우를 확인하고,

상기 접속 제어 애플리케이션을 통해, 상기 확인된 데이터 플로우의 데이터 플로우 정보를 삭제하고,

상기 접속 제어 애플리케이션을 통해, 상기 외부 서버에게 상기 확인된 데이터 플로우의 종료를 요청하도록 하는, 노드.

청구항 9

노드의 동작 방법에 있어서,

상기 노드의 접속 제어 애플리케이션을 통해, 외부 서버에게 서비스 서버에 대한 네트워크 접속을 요청하는 동작, 상기 네트워크 접속 요청은 상기 서비스 서버의 네트워크에 접속하려는 상기 노드의 타겟 애플리케이션의 식별 정보, 상기 서비스 서버의 IP(Internet Protocol) 및 포트를 포함하고,

상기 서비스 서버의 네트워크에 접속이 가능한 경우, 상기 타겟 애플리케이션의 파일 IO(input output)의 허용

여부를 나타내는 인가된 파일 I0 정보를 포함하는 데이터 플로우를 수신하는 동작,

상기 접속 제어 애플리케이션을 통해, 상기 타겟 애플리케이션과 관련된 파일 I0가 존재하는지를 확인하는 동작, 및

상기 인가된 파일 I0 정보에 기초하여, 상기 관련된 파일 I0의 동작을 관리하는 동작을 포함하는, 동작 방법.

청구항 10

청구항 9에 있어서, 상기 관련된 파일 I0의 동작을 관리하는 동작은,

상기 접속 제어 애플리케이션을 통해, 상기 인가된 파일 I0 정보에 기초하여 상기 관련된 파일 I0의 허용 여부를 확인하는 동작, 및

상기 관련된 파일 I0가 허용되지 않는 경우, 상기 접속 제어 애플리케이션을 통해, 상기 관련된 파일 I0를 차단하는 동작을 포함하는, 동작 방법.

청구항 11

청구항 9에 있어서, 상기 관련된 파일 I0의 동작을 관리하는 동작은,

상기 접속 제어 애플리케이션을 통해, 상기 관련된 파일 I0의 종류를 식별하는 동작,

상기 관련된 파일 I0의 상기 식별된 종류가 쓰기와 관련된 경우, 상기 접속 제어 애플리케이션을 통해, 상기 타겟 애플리케이션의 파일 쓰기를 제어하는 동작, 및

상기 관련된 파일 I0의 상기 식별된 종류가 읽기와 관련된 경우, 상기 접속 제어 애플리케이션을 통해, 상기 타겟 애플리케이션의 파일 읽기를 제어하는 동작을 포함하는, 동작 방법.

청구항 12

청구항 9에 있어서,

상기 접속 제어 애플리케이션을 통해, 상기 타겟 애플리케이션의 파일 I0에 대한 접근이 확인되면, 상기 타겟 애플리케이션에 할당된 데이터 플로우가 존재하는지를 확인하는 동작,

상기 타겟 애플리케이션에 할당된 데이터 플로우가 존재하는 경우, 상기 접속 제어 애플리케이션을 통해, 상기 타겟 애플리케이션에 대한 상기 인가된 파일 I0 정보를 확인하는 동작, 및

상기 타겟 애플리케이션이 접근하는 상기 파일 I0가 허용되지 않는 경우, 상기 접속 제어 애플리케이션을 통해, 상기 타겟 애플리케이션이 접근하는 상기 파일 I0를 차단하는 동작을 포함하는 동작 방법.

청구항 13

청구항 12에 있어서,

상기 타겟 애플리케이션에 할당된 데이터 플로우가 존재하지 않는 경우, 상기 접속 제어 애플리케이션을 통해, 상기 타겟 애플리케이션이 접근하는 상기 파일 I0와 관련된 파일 정보를 파일 I0 테이블에 저장하는 동작을 포함하는, 동작 방법.

청구항 14

청구항 9에 있어서,

상기 접속 제어 애플리케이션을 통해, 상기 타겟 애플리케이션의 데이터 패킷 전송 요청을 확인하는 동작,

상기 접속 제어 애플리케이션을 통해, 파일 I0 테이블에서 상기 타겟 애플리케이션이 접근하는 파일 I0의 대상 파일을 확인하는 동작,

상기 접속 제어 애플리케이션을 통해, 상기 대상 파일의 파일 정보를 확인하는 동작,

상기 접속 제어 애플리케이션을 통해, 상기 전송 요청된 데이터 패킷이 상기 파일 정보에 포함된 식별 정보를 포함하는지를 확인하는 동작,

상기 데이터 패킷이 상기 식별 정보를 포함하는 경우, 상기 데이터 패킷을 드롭하는 동작, 및

상기 데이터 패킷이 상기 식별 정보를 포함하지 않는 경우, 상기 데이터 패킷을 상기 서비스 서버로 전송하는 동작을 포함하는, 동작 방법.

청구항 15

청구항 9에 있어서,

상기 접속 제어 애플리케이션을 통해, 상기 외부 서버에게 외부 서버 접속 요청을 송신하는 동작,

상기 접속 제어 애플리케이션을 통해, 상기 외부 서버로부터 화이트 리스트를 수신하는 동작,

상기 타겟 애플리케이션이 상기 화이트 리스트에 포함된 경우, 상기 접속 제어 애플리케이션을 통해, 상기 타겟 애플리케이션의 무결성 및 안정성 여부를 검사하는 동작,

상기 접속 제어 애플리케이션을 통해, 상기 외부 서버로부터 상기 타겟 애플리케이션의 검사 결과를 송신하는 동작, 및

상기 접속 제어 애플리케이션을 통해, 상기 외부 서버로부터 상기 데이터 플로우를 수신하는 동작을 포함하는, 동작 방법.

청구항 16

청구항 9에 있어서,

상기 접속 제어 애플리케이션을 통해, 상기 타겟 애플리케이션의 종료를 확인하는 동작,

상기 접속 제어 애플리케이션을 통해, 데이터 플로우 테이블에서 상기 타겟 애플리케이션의 식별 정보에 대응하는 데이터 플로우를 확인하는 동작,

상기 접속 제어 애플리케이션을 통해, 상기 확인된 데이터 플로우의 데이터 플로우 정보를 삭제하는 동작, 및

상기 접속 제어 애플리케이션을 통해, 상기 외부 서버에게 상기 확인된 데이터 플로우의 종료를 요청하는 동작을 포함하는, 동작 방법.

청구항 17

서버에 있어서,

통신 회로;

상기 통신 회로와 작동적으로 연결되는 프로세서; 및

상기 프로세서와 작동적으로 연결되고, 데이터베이스를 저장하는 메모리를 포함하고, 상기 메모리는, 상기 프로세서에 의해서 실행될 때 상기 서버가,

노드의 접속 제어 애플리케이션으로부터 서비스 서버에 대한 네트워크 접속 요청을 수신하되, 상기 네트워크 접속 요청은 상기 서비스 서버의 네트워크에 접속하려는 상기 노드의 타겟 애플리케이션의 식별 정보, 상기 서비스 서버의 IP(Internet Protocol) 및 포트를 포함하고,

상기 타겟 애플리케이션이 상기 서비스 서버의 네트워크에 접속이 가능한 경우, 상기 타겟 애플리케이션의 파일 IO(input output)의 허용 여부를 나타내는 인가된 파일 IO 정보를 포함하는 데이터 플로우를 생성하고,

상기 노드의 상기 접속 제어 애플리케이션에게 상기 데이터 플로우를 송신하도록 하는 명령어들을 저장하는 서버.

청구항 18

서버의 동작 방법에 있어서,

노드의 접속 제어 애플리케이션으로부터 서비스 서버에 대한 네트워크 접속 요청을 수신하는 동작, 상기 네트워크 접속 요청은 상기 서비스 서버의 네트워크에 접속하려는 상기 노드의 타겟 애플리케이션의 식별 정보, 상기 서비스 서버의 IP(Internet Protocol) 및 포트를 포함하고,

상기 타겟 애플리케이션이 상기 서비스 서버의 네트워크에 접속이 가능한 경우, 상기 타겟 애플리케이션의 파일 IO(input output)의 허용 여부를 나타내는 인가된 파일 IO 정보를 포함하는 데이터 플로우를 생성하는 동작, 및

상기 노드의 상기 접속 제어 애플리케이션에게 상기 데이터 플로우를 송신하는 동작을 포함하는 동작 방법.

발명의 설명

기 술 분 야

[0001] 본 문서에서 개시된 실시 예들은 애플리케이션의 파일 송신 및 수신을 제어하기 위한 시스템 및 그에 관한 방법에 관한 것이다.

배 경 기 술

[0002] 단말과 서버 간에는 IP(Internet Protocol) 기반의 TCP(Transmission Control Protocol)를 사용하며 통신이 수행되며, 데이터 패킷 내에 포함된 5 튜플(Tuples) 정보를 이용하여 출발지 IP, 도착지 IP, 및 포트 정보를 식별하여 접속 제어를 수행하는 방화벽 기술이 보편적으로 사용된다.

[0003] 방화벽 기술은 단말 또는 네트워크 단말(예: 게이트웨이, 공유기 등)에 할당된 IP를 식별하여 네트워크 경계 사이에서 인바운드 또는 아웃바운드 데이터 패킷의 접근 제어를 수행함으로써 비인가된 IP가 비인가된 목적지 네트워크로의 접속을 차단하는 역할을 수행한다.

[0004] IP를 기반으로 하는 방화벽과 같은 기술은 IP 할당 및 제어가 어려운 인터넷 대역에 있는 단말 또는 공유기 및 게이트웨이에 의해 서브 네트워크를 구성하여 사설 IP 대역을 만드는 경우 IP 단위로의 통제가 어려운 문제점이 있다. 또한, IP 통신 구조상 IP를 위조, 변조 가능하는 문제점이 있다. 이에 따라, 방화벽은 최소한의 안전 장치로써 사용되고 있다.

[0005] 이러한 문제점을 해결하기 위해 단말과 서버(또는, 게이트웨이) 사이에 전송되는 데이터 패킷을 암호화하거나 위조, 변조되지 않도록 하고 인가된 대상만 유일한 접속을 허용할 수 있도록 하는 터널링 기술(예: IPSec(Internet Protocol Security), GRE(Generic Routing Encapsulation), GTP(GPRS Tunneling Protocol)) 또는 보안 세션(SSL(Secure Sockets Layer), TLS(Transport Layer Security))과 같은 접속성 제어(Connectivity Control) 기술을 사용함으로써 방화벽이 내재하고 있는 문제점을 해결하고 있다.

[0006] 하지만 단말(단말에 부여된 IP 단위) 수준으로 생성되는 터널링 기술 기반의 VPN(Virtual Private Network) 사용 시, 최초 인증 이후 상시 연결된 터널링을 통해서 비허용된 또는 안전하지 않은 애플리케이션이 허용되지 않은 목적지 네트워크에 접속하는 취약점을 내재하고 있기 때문에, 각종 멀웨어 및 랜섬웨어 침투에 의한 보안 사고가 증가하고 있다.

[0007] 이러한 문제점을 해결하기 위해 허용된 애플리케이션만 허용된 네트워크에 접속할 수 있는 애플리케이션 접속성 제어 기술을 적용하여 근본적인 통신 주체인 애플리케이션을 식별하고, 허용되지 않은 애플리케이션의 접속을 사전에 차단함으로써 각종 멀웨어 및 랜섬웨어가 허용되지 않은 목적지 네트워크에 접속하는 것을 방지함과 동시에 현재 발생되고 있는 각종 네트워크 보안 문제를 효과적으로 해결할 수 있다.

발명의 내용

해결하려는 과제

[0008] 최근 산업의 발전에 따라 경계를 초월한 글로벌 업무 환경 및 재택 업무 환경의 증가, 클라우드로 전환이 증가되면서 단말은 언제 어디서든 업무 시스템에 접속할 수 있게 되었으며, 이러한 환경에서의 애플리케이션 접속 제어 기술은 업무 시스템을 보호하기 위한 최적의 보안 요소를 제공한다.

[0009] 하지만, 허용된 애플리케이션이 허용된 네트워크에 접속 가능한 상태에서 애플리케이션이 네트워크를 통해서 파일을 수신하거나 전송하는 행위를 통제할 수 없는 문제가 존재한다.

[0010] 일례로, 인터넷에 연결된 PC를 사용하는 재택 근무 환경에서 허용된 애플리케이션으로 클라우드에 접속해서 중요 파일을 수신하여 외부로 유출하거나, 인터넷으로부터 유입된 악성 코드를 포함하는 파일을 클라우드에 전송

하여 타 단말로 위험이 전파되는 것을 방지할 수 없다.

[0011] 즉, 매크로 또는 프로그래밍 가능한 정보를 포함하고, 애플리케이션 자체에 포함된 런타임을 통해서 실행 가능한 파일을 로딩하는 애플리케이션(예: 문서 편집 도구, 그래픽 편집 도구 등)의 경우, 사용자가 유추하지 못하는 악성 코드를 포함할 수 있으며 이러한 악성 코드는 클라우드에 접속하여 중요 파일(예: 개인 정보, 기밀 정보)을 수신 받아 외부에 유출하거나, 악성 코드가 포함된 파일이 클라우드에 전송되어 클라우드에 접속된 모든 단말에 전파될 수 있는 문제를 내재하고 있다.

[0012] 이러한 문제점을 해결하기 위한 유사 기술로써 DLP(Data Loss Prevention) 기술이 존재하며, DLP 기술은 허용되지 않은 매체(예: USB(Universal Serial Bus), 이동식 저장 장치 등)가 연결되는 것을 방지하거나, 네트워크 상에서 특정 폴더를 공유하는 것을 방지하거나, 특정 애플리케이션이 파일을 전송하기 위해, 파일을 선택하는 경우 파일이 선택되지 않게 차단하는 방식으로 파일이 전송되는 것을 방지하거나, 특정 애플리케이션의 파일 IO(input output)(또는, 파일 시스템 IO)가 발생하는 경우, 파일 IO를 추적해서 해당 사용자가 파일 IO 권한이 없는 경우, 파일 IO의 핸들(Handle)을 제거하는 방식으로의 데이터 유출을 방지한다.

[0013] 이러한 DLP 기술의 문제점은 해당 애플리케이션이 설치된 단말은 일괄적인 보안 정책이 상시 강제 적용되기 때문에 물리적 보안 통제가 불가능한 영역 및 보안 정책을 적용하는 것에 한계가 존재하는 개인 소유의 단말을 사용하는 재택 근무 환경에서 허용되지 않은 매체가 연결되는 것을 방지하는 행위를 강제할 수 없다.

[0014] 나아가, 특정 행위를 추적하는 DLP 기술을 사용하는 경우, 사용자에 의해서 특정 동작을 수행하는 행위(예: 애플리케이션 내의 파일 선택 창을 여는 행위)에 따라 동작하기 때문에 특정 동작을 수행하지 않는 경우, 예를 들어 사용자가 아닌 악성 코드가 파일 선택 창 없이 직접적으로 파일을 적재하고 전송하는 행위를 막을 수 없다.

[0015] 이러한 문제를 해결하기 위해 특정 애플리케이션의 파일 시스템 IO를 추적하는 방식의 DLP 기술이 사용되지만, 해당 기술은 특정 애플리케이션을 대상으로 파일의 IO를 추적하고 일괄적으로 차단하기 때문에 해당 애플리케이션이 접속한 특정 네트워크에서 파일 전송을 허용하기 어려운 문제가 존재하며, 일부 DLP 기술의 경우 이러한 문제를 해결하기 위해 인터넷 브라우저의 접속 주소창을 식별하여 파일 IO 제한을 해제하는 방식을 제공하고 있지만 접속 주소창이 존재하지 않는 애플리케이션의 경우, 네트워크 접속 정보를 추적할 수 없기 때문에 파일 수신 및 전송을 제어하는 데에 한계가 존재한다.

[0016] 본 문서에 개시되는 다양한 실시 예들은 네트워크 환경에서 상술한 문제점을 해결하기 위한 시스템 및 그에 관한 방법을 제공하고자 한다.

과제의 해결 수단

[0017] 본 문서에서 개시되는 일 실시 예에 따른 노드는, 통신 회로, 상기 통신 회로와 작동적으로 연결되는 프로세서, 및 상기 프로세서와 작동적으로 연결되고, 접속 제어 애플리케이션 및 타겟 애플리케이션을 저장하는 메모리를 포함하고, 상기 메모리는, 상기 프로세서에 의해서 실행될 때 상기 노드가, 상기 접속 제어 애플리케이션을 통해, 외부 서버에게 서비스 서버에 대한 네트워크 접속을 요청하되, 상기 네트워크 접속 요청은 상기 서비스 서버의 네트워크에 접속하려는 상기 타겟 애플리케이션의 식별 정보, 상기 서비스 서버의 IP(Internet Protocol) 및 포트를 포함하고, 상기 서비스 서버의 네트워크에 접속이 가능한 경우, 상기 타겟 애플리케이션의 파일 IO(input output)의 허용 여부를 나타내는 인가된 파일 IO 정보를 포함하는 데이터 플로우를 수신하고, 상기 접속 제어 애플리케이션을 통해, 상기 타겟 애플리케이션과 관련된 파일 IO가 존재하는지를 확인하고, 상기 인가된 파일 IO 정보에 기초하여, 상기 관련된 파일 IO의 동작을 관리하도록 하는 명령어들을 저장할 수 있다.

[0018] 본 문서에서 개시되는 일 실시 예에 따른 노드의 동작 방법은 상기 노드의 접속 제어 애플리케이션을 통해, 외부 서버에게 서비스 서버에 대한 네트워크 접속을 요청하는 동작, 상기 네트워크 접속 요청은 상기 서비스 서버의 네트워크에 접속하려는 상기 노드의 타겟 애플리케이션의 식별 정보, 상기 서비스 서버의 IP(Internet Protocol) 및 포트를 포함하고, 상기 서비스 서버의 네트워크에 접속이 가능한 경우, 상기 타겟 애플리케이션의 파일 IO(input output)의 허용 여부를 나타내는 인가된 파일 IO 정보를 포함하는 데이터 플로우를 수신하는 동작, 상기 접속 제어 애플리케이션을 통해, 상기 타겟 애플리케이션과 관련된 파일 IO가 존재하는지를 확인하는 동작, 및 상기 인가된 파일 IO 정보에 기초하여, 상기 관련된 파일 IO의 동작을 관리하는 동작을 포함할 수 있다.

발명의 효과

[0019] 본 문서에 개시되는 다양한 실시 예들은, 애플리케이션 접속성 제어 기술을 활용하여 애플리케이션과 해당 애플리케이션이 접속한 네트워크를 식별한 정보를 기반으로 파일 IO를 통제할 수 있다.

[0020] 본 문서에 개시되는 다양한 실시 예들은 단말의 파일 시스템의 IO를 상시 감시하고, 접속된 또는 접속하고자 하는 애플리케이션과 네트워크 정보를 식별하여 식별된 대상의 파일 IO를 허용할 것인지 여부를 중앙화된 컨트롤러를 통해서 확인하고, 허용된 접속 대상에 한해서 파일 IO를 허용함으로써 애플리케이션 및 네트워크별 파일 수신 및 전송 제어가 가능하다.

[0021] 나아가, 본 문서에 개시되는 다양한 실시 예들은, 네트워크 접속 전 파일 IO를 사용하고 있는 상태인 경우(파일을 전송하기 위한 예비 상태), 네트워크 접속 전 파일 IO를 사용하였다가 해제한 경우(파일을 전송하기 위한 예비 상태), 네트워크 접속 후 파일 IO을 사용하고 있는 상태인 경우(파일을 전송하거나 수신하고 있는 상태)와 같이 네트워크 접속 및 접속 대상(애플리케이션 및 네트워크) 식별 시점과 파일 IO 처리 시점이 상이한 경우에 대응하는 네트워크 및 파일 IO를 제어함으로써 파일 수신 및 전송의 세밀한 제어가 가능하고, 파일이 접속 제어 애플리케이션을 우회하여 전송되거나 수신되지 않게 한다.

[0022] 본 문서에 개시되는 다양한 실시 예들은, 네트워크 접속 제어와 파일 IO 제어가 결합되어, 접속 제어 애플리케이션이 종료되는 경우 대상 네트워크로의 접속이 차단함으로써 파일을 수신하는 행위를 사전에 차단할 수 있다.

[0023] 나아가, 본 문서에 개시되는 다양한 실시 예들은, 파일 시스템의 IO를 상시 추적함으로써 특정 애플리케이션이 파일을 전송하기 위해, 파일을 선택하는 경우 파일이 선택되지 않게 차단하는 방식으로 파일이 전송되는 것을 방지하는 형태의 DLP 기술이 내재한 문제점인 파일 시스템에 직접적으로 접근하여 네트워크로 파일을 전송하는 문제점을 해결할 수 있다.

[0024] 이 외에, 본 문서를 통해 직접적 또는 간접적으로 파악되는 다양한 효과들이 제공될 수 있다.

도면의 간단한 설명

[0025] 도 1은 다양한 실시 예들에 따른 네트워크 환경 내의 아키텍처를 나타낸다.

도 2는 다양한 실시 예들에 따라 컨트롤러에 저장된 데이터베이스를 나타내는 기능적 블록도이다.

도 3은 다양한 실시 예들에 따른 제어 플로우 및 데이터 플로우에 포함된 정보를 나타낸다.

도 4는 다양한 실시 예들에 따른 노드의 기능적 블록도를 나타낸다.

도 5는 다양한 실시 예들에 따라 데이터 패킷의 전송을 제어하는 동작을 설명한다.

도 6은 다양한 실시 예들에 따른 컨트롤러 접속을 위한 신호 흐름도를 나타낸다.

도 7은 다양한 실시 예들에 따른 사용자 인증을 위한 신호 흐름도를 도시한다.

도 8은 다양한 실시 예들에 따른 컨트롤러 접속을 위한 사용자 인터페이스 화면을 도시한다.

도 9는 다양한 실시 예들에 따른 네트워크 접속을 위한 신호 흐름도를 도시한다.

도 10은 다양한 실시 예들에 따른 파일 IO 접근 감시를 위한 노드의 동작 흐름도를 도시한다.

도 11은 다양한 실시 예들에 따른 파일 IO 접근 감시 결과를 알리는 사용자 인터페이스 화면을 도시한다.

도 12는 다양한 실시 예들에 따른 파일 IO 테이블 검사를 위한 노드의 동작 흐름도를 도시한다.

도 13은 다양한 실시 예들에 따른 데이터 패킷 전송을 위한 노드의 동작 흐름도를 도시한다.

도 14는 다양한 실시 예들에 따른 제어 플로우 갱신을 위한 신호 흐름도를 도시한다.

도 15는 다양한 실시 예들에 따른 제어 플로우 제거를 위한 신호 흐름도를 도시한다.

도 16은 다양한 실시 예들에 따른 데이터 플로우 제거를 위한 신호 흐름도를 도시한다.

발명을 실시하기 위한 구체적인 내용

[0026] 이하, 본 발명의 다양한 실시 예가 첨부된 도면을 참조하여 기재된다. 그러나, 이는 본 발명을 특정한 실시 형태에 대해 한정하려는 것이 아니며, 본 발명의 실시 예의 다양한 변경(modification), 균등물(equivalent), 및/또는 대체물(alternative)을 포함하는 것으로 이해되어야 한다.

[0027] 본 문서에서 아이템에 대응하는 명사의 단수 형은 관련된 문맥상 명백하게 다르게 지시하지 않는 한, 상기 아이템 한 개 또는 복수 개를 포함할 수 있다. 본 문서에서, "A 또는 B", "A 및 B 중 적어도 하나","A 또는 B 중 적어도 하나", "A, B 또는 C", "A, B 및 C 중 적어도 하나" 및 "A, B, 또는 C 중 적어도 하나"와 같은 문구들 각각은 그 문구들 중 해당하는 문구에 함께 나열된 항목들 중 어느 하나, 또는 그들의 모든 가능한 조합을 포함할 수 있다. "제 1", "제 2", 또는 "첫째" 또는 "둘째"와 같은 용어들은 단순히 해당 구성요소를 다른 해당 구성요소와 구분하기 위해 사용될 수 있으며, 해당 구성요소들을 다른 측면(예: 중요성 또는 순서)에서 한정하지 않는다. 어떤(예: 제 1) 구성요소가 다른(예: 제 2) 구성요소에, "기능적으로" 또는 "통신적으로"라는 용어와 함께 또는 이런 용어 없이, "커플드" 또는 "커넥티드"라고 언급된 경우, 그것은 상기 어떤 구성요소가 상기 다른 구성요소에 직접적으로(예: 유선으로), 무선으로, 또는 제 3 구성요소를 통하여 연결될 수 있다는 것을 의미한다.

[0028] 본 문서에서 설명되는 구성요소들의 각각의 구성요소(예: 모듈 또는 프로그램)는 단수 또는 복수의 개체를 포함할 수 있다. 다양한 실시 예들에 따르면, 해당 구성요소들 중 하나 이상의 구성요소들 또는 동작들이 생략되거나, 또는 하나 이상의 다른 구성요소들 또는 동작들이 추가될 수 있다. 대체적으로 또는 추가적으로, 복수의 구성요소들(예: 모듈 또는 프로그램)은 하나의 구성요소로 통합될 수 있다. 이런 경우, 통합된 구성요소는 상기 복수의 구성요소들 각각의 구성요소의 하나 이상의 기능들을 상기 통합 이전에 상기 복수의 구성요소들 중 해당 구성요소에 의해 수행되는 것과 동일 또는 유사하게 수행할 수 있다. 다양한 실시 예들에 따르면, 모듈, 프로그램 또는 다른 구성요소에 의해 수행되는 동작들은 순차적으로, 병렬적으로, 반복적으로, 또는 휴리스틱하게 실행되거나, 상기 동작들 중 하나 이상이 다른 순서로 실행되거나, 생략되거나, 또는 하나 이상의 다른 동작들이 추가될 수 있다.

[0029] 본 문서에서 사용되는 용어 "모듈"은 하드웨어, 소프트웨어 또는 펌웨어로 구현된 유닛을 포함할 수 있으며, 예를 들면, 로직, 논리 블록, 부품, 또는 회로와 같은 용어와 상호 호환적으로 사용될 수 있다. 모듈은, 일체로 구성된 부품 또는 하나 또는 그 이상의 기능을 수행하는, 상기 부품의 최소 단위 또는 그 일부가 될 수 있다. 예를 들면, 일 실시 예에 따르면, 모듈은 ASIC(application-specific integrated circuit)의 형태로 구현될 수 있다.

[0030] 본 문서의 다양한 실시 예들은 기기(machine) 의해 읽을 수 있는 저장 매체(storage medium)(예: 메모리)에 저장된 하나 이상의 명령어들을 포함하는 소프트웨어(예: 프로그램 또는 애플리케이션)로서 구현될 수 있다. 예를 들면, 기기의 프로세서는, 저장 매체로부터 저장된 하나 이상의 명령어들 중 적어도 하나의 명령을 호출하고, 그것을 실행할 수 있다. 이것은 기기가 상기 호출된 적어도 하나의 명령어에 따라 적어도 하나의 기능을 수행하도록 운영되는 것을 가능하게 한다. 상기 하나 이상의 명령어들은 컴파일러에 의해 생성된 코드 또는 인터프리터에 의해 실행될 수 있는 코드를 포함할 수 있다. 기기로 읽을 수 있는 저장 매체는, 비일시적(non-transitory) 저장 매체의 형태로 제공될 수 있다. 여기서, '비일시적'은 저장 매체가 실재(tangible)하는 장치이고, 신호(signal)(예: 전자기파)를 포함하지 않는다는 것을 의미할 뿐이며, 이 용어는 데이터가 저장 매체에 반영구적으로 저장되는 경우와 임시적으로 저장되는 경우를 구분하지 않는다.

[0031] 본 문서에 개시된 다양한 실시 예들에 따른 방법은 컴퓨터 프로그램 제품(computer program product)에 포함되어 제공될 수 있다. 컴퓨터 프로그램 제품은 상품으로서 판매자 및 구매자 간에 거래될 수 있다. 컴퓨터 프로그램 제품은 기기로 읽을 수 있는 저장 매체(예: compact disc read only memory(CD-ROM))의 형태로 배포되거나, 또는 애플리케이션 스토어를 통해 또는 두 개의 사용자 장치들(예: 스마트폰들) 간에 직접, 온라인으로 배포(예: 다운로드 또는 업로드)될 수 있다. 온라인 배포의 경우에, 컴퓨터 프로그램 제품의 적어도 일부는 제조사의 서버, 애플리케이션 스토어의 서버, 또는 중계 서버의 메모리와 같은 기기로 읽을 수 있는 저장 매체에 적어도 일시 저장되거나, 임시적으로 생성될 수 있다.

[0032] 도 1은 다양한 실시 예들에 따른 네트워크 환경 내의 아키텍처를 나타낸다.

[0033] 도 1에 도시된 노드(201)는 데이터 통신을 수행할 수 있는 다양한 형태의 장치일 수 있다. 예를 들어, 노드(201)는 스마트폰 또는 태블릿과 같은 휴대용 장치, 데스크탑(desktop) 또는 랩탑(laptop)과 같은 컴퓨터 장치, 멀티미디어 장치, 의료 기기, 카메라, 웨어러블 장치, VR(virtual reality) 장치, 또는 가전 장치를 포함할 수 있으며 전술한 기기들에 한정되지 않는다. 예를 들어, 노드(201)는 애플리케이션을 통해 데이터 패킷을 전송할 수 있는 서버 또는 게이트웨이를 포함할 수 있다. 노드(201)는 '전자 장치' 또는 '단말'로도 참조될 수 있다.

[0034] 노드(201)는 복수의 타겟 애플리케이션(211-1, 211-2, 211-3) 및 접속 제어 애플리케이션(212)을 저장할 수 있다.

[0035] 타겟 애플리케이션(211-1, 211-2, 211-3) 각각은 접속 제어 애플리케이션(212)의 제어 하에 게이트웨이(203-1, 203-2)를 통해 서비스 서버(205-1, 205-2)로 데이터 패킷을 전송하거나 반대로 데이터 패킷을 수신할 수 있다.

[0036] 타겟 애플리케이션(211-1, 211-2, 211-3) 중 일부는 웹 브라우저 또는 비즈니스 애플리케이션과 같이 허용된 및/또는 보안된 애플리케이션인 반면에 다른 일부는 허용되지 않은 프로그램(예: 멀웨어, 랜섬웨어, 기타 허용되지 않은 애플리케이션)이거나 보안되지 않은 악성 프로그램(악성 코드에 감염되거나, 위조 또는 변조된 애플리케이션)일 수 있다. 실시 예들에 따르면 접속 제어 애플리케이션(212)은 데이터 패킷의 송신을 요청하는 프로그램(또는, 타겟 애플리케이션, 타겟 애플리케이션의 프로세스)을 식별하고, 허용되지 않은 프로그램의 데이터 패킷이 노드(201)의 외부로 송신되는 것을 방지할 수 있다. 또한, 실시 예들에 따르면, 접속 제어 애플리케이션(212)은 접속 제어 애플리케이션(212)과 게이트웨이(203-1, 203-2) 간 보안 세션(220-1, 220-2)을 통해 인가되지 않은 프로그램의 서비스 서버(205-1, 205-2)에 대한 접속을 차단하고 해당 프로그램을 격리할 수 있다.

[0037] 예를 들어, 실시 예들에 따른 타겟 애플리케이션(211-1, 211-2, 211-3)이 서비스 서버(205-1, 205-2)와 통신하기 이전에 접속 제어 애플리케이션(212)은 컨트롤러(202)로부터 접속 가능 여부를 확인하고, 접속 가능한 경우 컨트롤러(202)에 의하여 인증된 데이터 패킷을 통해 게이트웨이(203-1, 203-2)와 인증을 수행할 수 있다. 인증이 완료되면 접속 제어 애플리케이션(212)은 게이트웨이(203-1, 203-2)와 보안 세션(220-1, 220-2)을 생성할 수 있다.

[0038] 다양한 실시 예들에 따르면, 노드(201)는 노드(201) 내에 저장된 타겟 애플리케이션(211-1, 211-2, 211-3)의 네트워크 접속을 관리하기 위한 접속 제어 애플리케이션(212) 및 네트워크 드라이버(미도시)를 포함할 수 있다. 예를 들어, 노드(201)에 저장된 타겟 애플리케이션(211-1, 211-2, 211-3)의 서비스 서버(205-1, 205-2)에 대한 접속 이벤트가 발생하면, 접속 제어 애플리케이션(212)은 타겟 애플리케이션(211-1, 211-2, 211-3)의 접속 가능 여부를 결정할 수 있다. 타겟 애플리케이션(211-1, 211-2, 211-3)이 접속 가능하면, 접속 제어 애플리케이션(212)은 보안 세션(220-1, 220-2)을 통해 게이트웨이(203-1, 203-2)로 데이터 패킷을 전송할 수 있다. 접속 제어 애플리케이션(212)은 노드(201) 내에서 운영체제를 포함하는 커널(kernel) 및 네트워크 드라이버를 통해 데이터 패킷의 전송을 제어할 수 있다.

[0039] 또한 접속 제어 애플리케이션(212)은 커널에서 파일 시스템 IO(input output)를 감시 및 제어할 수 있다. 파일 시스템 IO는 임의 애플리케이션(또는, 임의 프로세스)에 의해 커널의 파일 시스템에서 발생하는 입력 및/또는 출력을 의미한다. 파일 시스템 IO는 파일 시스템에 특정 정보를 쓰거나, 파일 시스템의 특정 정보를 읽거나, 또는 갱신하는 동작 등을 포함한다. 파일 시스템 IO를 위해, 임의 애플리케이션(또는, 임의 프로세스)은 파일 시스템을 사용하기 위한 핸들(handle)(예: 파일 핸들)을 획득하여야 한다.

[0040] 접속 제어 애플리케이션(212)은 파일 시스템 IO을 감시하기 위해 파일 시스템 드라이버(미도시) 또는 파일 시스템 IO 이벤트에 대한 후킹(hooking)을 통한 파일 시스템 IO의 감시가 가능한 파일 IO 제어 모듈(미도시)을 포함할 수 있다. 파일 시스템 IO는 파일 IO로도 참조될 수 있다.

[0041] 파일 IO 제어 모듈은 파일 IO를 감시하여 타겟 애플리케이션(211-1, 211-2, 211-3) 중 어느 애플리케이션이 파일 시스템에 접근(또는, 파일 IO 접근)하는지를 식별할 수 있다.

[0042] 파일 IO 제어 모듈은 데이터 플로우와 관련되는 애플리케이션의 식별 정보 및 프로세스 식별자(PID: process identifier)를 통해 해당 애플리케이션이 접속된 네트워크와 연계하여 해당 애플리케이션의 파일 IO 접근 가능 여부를 확인할 수 있다. 파일 IO 제어 모듈은 애플리케이션의 파일 IO 접근이 불가능한 경우 해당 애플리케이션이 더 이상 파일 IO에 접근할 수 없도록 처리하여, 노드(201)가 파일 전송 및 파일 수신에 대해 세밀하게 제어하게 한다. 파일 IO 제어 모듈은 애플리케이션의 파일 IO 접근이 불가능한 경우 해당 애플리케이션이 더 이상 파일 IO에 접근할 수 없도록 파일 IO 핸들을 제거할 수 있다. 파일 IO 핸들의 제거는 파일 IO 제거로도 참조될 수 있다.

[0043] 컨트롤러(202)는 예를 들어, 서버(또는, 클라우드 서버)일 수 있다. 컨트롤러(202)는 노드(201), 게이트웨이(203-1, 203-2), 및 서비스 서버(205-1, 205-2) 간 데이터 전송을 관리함으로써 네트워크 환경 내에서 신뢰되는 데이터 전송을 보장할 수 있다. 예를 들어, 컨트롤러(202)는 정책 정보 또는 블랙리스트 정보를 통해 인가된 노드(201)(또는, 접속 제어 애플리케이션(212))의 네트워크 접속을 허용할 수 있다.

[0044] 또한, 컨트롤러(202)는 접속 제어 애플리케이션(212)과 게이트웨이(203-1, 203-2) 간 보안 세션(220-1, 220-2)의 생성을 중개하거나, 노드(201) 또는 게이트웨이(203-1, 203-2)로부터 수집된 보안 이벤트에 따라서 보안 세션(220-1, 220-2)을 제거할 수 있다. 접속 제어 애플리케이션(212)은 컨트롤러(202)에 의하여 인가된 보안 세

션(220-1, 220-2)을 통해서만 서비스 서버(205-1, 205-2)와 통신할 수 있으며, 인가된 보안 세션(220-1, 220-2)이 존재하지 않으면 노드(201) 및 접속 제어 애플리케이션(212)의 네트워크 접속이 차단될 수 있다. 일 실시예에서, 타겟 애플리케이션(211-1, 211-2, 211-3)은 컨트롤러(202)에 의하여 인가된 보안 세션(220-1, 220-2)을 통해서만 서비스 서버(205-1, 205-2)와 통신할 수 있으며, 인가된 보안 세션(220-1, 220-2)이 존재하지 않으면 타겟 애플리케이션(211-1, 211-2, 211-3)의 네트워크 접속은 접속 제어 애플리케이션(212), 컨트롤러(202) 또는 게이트웨이(203)로부터 차단될 수 있다. 일 실시 예에 따르면, 컨트롤러(202)는 노드(201) 또는 접속 제어 애플리케이션(212)의 네트워크 접속과 연관된 다양한 동작(예: 등록, 승인, 인증, 갱신, 종료)을 수행하기 위하여 접속 제어 애플리케이션(212)과 제어 데이터 패킷을 송수신할 수 있다. 제어 데이터 패킷이 전송되는 흐름(예: 230)은 '제어 플로우(control flow)'로 참조될 수 있다.

[0045] 게이트웨이(203-1, 203-2)는 노드(201)가 속하는 네트워크의 경계 또는 서비스 서버(205-1, 205-2)가 속하는 네트워크의 경계에 위치할 수 있다. 일 실시 예에 따르면, 게이트웨이(203-1, 203-2)는 클라우드(cloud) 기반으로 컨트롤러(202)와 연결될 수 있다. 게이트웨이(203-1, 203-2)는 접속 제어 애플리케이션(212)으로부터 수신된 데이터 패킷 중에서 인가된 데이터 패킷만을 서비스 서버(205-1, 205-2)로 포워딩할 수 있다. 접속 제어 애플리케이션(212) 및 게이트웨이(203-1, 203-2) 사이에서 데이터 패킷이 전송되는 흐름(예: 240-1, 240-2)은 '데이터 플로우(data flow)'로 참조될 수 있다.

[0046] 데이터 플로우는 노드 또는 IP 단위뿐만 아니라 보다 세부적인 단위(예: 애플리케이션 단위)로도 생성될 수 있다. 게이트웨이(203-1, 203-2)는 접속 제어 애플리케이션(212)과 게이트웨이(203-1, 203-2) 간 보안 세션(220-1, 220-2) 생성 이전에 접속 제어 애플리케이션(212)의 인증을 수행하고, 접속 제어 애플리케이션(212)으로부터 전송된 데이터 패킷 중 보안 세션(220-1, 220-2)을 통해 전송된 데이터 패킷만을 서비스 서버(205-1, 205-2)로 포워딩함으로써 무분별한 네트워크 접속을 사전에 차단할 수 있다.

[0047] 도 2는 다양한 실시 예들에 따라 컨트롤러(202)에 저장된 데이터베이스를 나타내는 기능적 블록도이며, 도 3은 데이터베이스에 포함된 정보 중 제어 플로우 정보(340), 데이터 플로우 정보(350) 및 파일 IO 테이블 정보(360)를 나타낸다.

[0048] 도 2를 참조하면, 컨트롤러(202)는 네트워크 접속 및 데이터 전송의 제어를 위한 데이터베이스(311 내지 317)를 메모리(330)에 저장할 수 있다. 도 2는 메모리(330)만을 도시하지만, 컨트롤러(202)는 외부 전자 장치(예: 도 1의 노드(201), 게이트웨이(203-1, 203-2) 또는 서비스 서버(205-1, 205-2))와 통신을 수행하기 위한 통신 회로(예: 도 4의 통신 회로(430)) 및 컨트롤러(202)의 전반적인 동작을 제어하기 위한 프로세서(예: 도 4의 프로세서(410))를 더 포함할 수 있다. 관리자는 컨트롤러(202)에 접속하여 접속 제어 애플리케이션(212)과 서비스 서버(205-1, 205-2) 간 접속을 제어하기 위한 연결 중심의 정책을 설정할 수 있으므로, 서비스 단에서 세션을 관리하는 것 보다 세밀하고 안전하게 네트워크 접속을 제어할 수 있다.

[0049] 접속 정책 데이터베이스(311)는 식별된 네트워크, 노드, 또는 애플리케이션이 접속 가능한 네트워크 및/또는 서비스에 대한 정보를 포함할 수 있다. 예를 들어, 컨트롤러(202)는 접속 제어 애플리케이션(212)의 네트워크 접속 요청 시, 접속 정책 데이터베이스(311)의 정책에 기반하여 식별된 네트워크(예: 노드(201)가 속하는 네트워크), 노드, 사용자(예: 노드(201)의 사용자), 및/또는 애플리케이션(예: 노드(201)에 포함되는 타겟 애플리케이션(211-1, 211-2, 211-3))이 서비스 서버(205-1, 205-2)에 접속이 가능한지 여부를 결정할 수 있다. 일 실시예에서, 컨트롤러(202)는 접속 정책 데이터베이스(311)에 기반하여 특정 서비스(예: IP 및 포트)로 접속 가능한 타겟 애플리케이션(211-1, 211-2, 211-3)의 화이트리스트를 생성할 수 있다.

[0050] 파일 IO 정책 데이터베이스(312)는 접속 정책과 연계되며, 타겟 애플리케이션(211-1, 211-2, 211-3)에 대한 파일 IO 정책을 수립하기 위한 정보를 포함할 수 있다. 예를 들어, 파일 IO 정책 데이터베이스(312)는 타겟 애플리케이션(211-1, 211-2, 211-3)의 파일 수신 및 전송을 허용할 것인지 또는 허용하지 않을 것인지에 대한 정보를 포함할 수 있다.

[0051] 블랙리스트 정책 데이터베이스(313)는 노드(201) 또는 게이트웨이(203-1, 203-2)에서 주기적으로 수집되는 보안 이벤트 중에서 보안 이벤트의 위험도, 발생 주기, 및/또는 행위 분석을 통해 식별된 대상(예: 노드 ID(identifier), IP 주소, MAC(media access control) 주소, 또는 사용자 ID 중 적어도 하나)의 접속을 차단하기 위한 블랙리스트 등록 정책을 나타낼 수 있다.

[0052] 블랙리스트 데이터베이스(314)는 블랙리스트 정책 데이터베이스(313)에 의해서 차단된 대상에 대한 목록을 포함할 수 있다. 예를 들어, 컨트롤러(202)는 네트워크 접속을 요청하는 노드(201)의 식별 정보가 블랙리스트 데이

터베이스(314)에 포함되면 네트워크 접속 요청을 거부함으로써 노드(201)를 격리시킬 수 있다.

[0053] 제어 플로우 테이블(315)은 접속 제어 애플리케이션(212)과 컨트롤러(202) 사이에 생성된 제어 데이터 패킷의 흐름(즉, 제어 플로우)을 관리하기 위한 세션 테이블의 일 예이다.

[0054] 예를 들어 도 3을 참조하면, 접속 제어 애플리케이션(212)이 성공적으로 컨트롤러(202)에 접속하는 경우 제어 플로우 정보(340) 및 제어 플로우 ID(342)(또는 '제어 플로우 식별 정보'로도 참조될 수 있다)가 컨트롤러(202)에 의하여 생성될 수 있다. 제어 플로우 정보(340)는 컨트롤러 접속 및 인증 시 식별된 IP, 노드, 애플리케이션, 또는 서비스 서버와의 연계를 통해 추가적으로 식별된 대상 중 적어도 하나를 나타내는 식별 정보(344)를 포함할 수 있다. 예를 들어 접속 제어 애플리케이션(212)의 네트워크 접속 요청이 수신되면, 컨트롤러(202)는 접속 요청에 포함된 제어 플로우 ID 및 식별 정보를 제어 플로우 정보(340)에 포함된 제어 플로우 ID(342) 및 식별 정보(344)와 매핑함으로써 접속 제어 애플리케이션(212)의 서비스 서버(205-1, 205-2)에 대한 접속 가능 여부 및 게이트웨이(203)와의 보안 세션(220-1, 220-2) 생성을 위한 데이터 플로우 생성 가능 여부를 결정할 수 있다. 상태 정보(346)는 제어 플로우의 생성, 인증, 갱신, 종료 등과 같은 다양한 상태를 나타낼 수 있다.

[0055] 일 실시 예에 따르면, 제어 플로우는 만료 시간을 가지므로, 노드(201)는 시간 정보(348)에 기반하여 제어 플로우의 만료 시간을 갱신해야 하며, 일정 시간 동안에 만료 시간이 갱신되지 않으면 제어 플로우(또는, 제어 플로우 정보(340))는 제거될 수 있다. 또한, 노드(201), 접속 제어 애플리케이션(212), 다른 보안 애플리케이션(미도시), 또는 게이트웨이로(203-1, 203-2)부터 수집된 보안 이벤트에 따라서 즉각적인 접속 차단이 필요하다고 결정되거나, 이들로부터 접속 종료 요청이 수신되면, 컨트롤러(202)는 제어 플로우를 제거할 수 있다. 제어 플로우가 제거되면 관련된 데이터 플로우 또한 제거되며, 게이트웨이(203-1, 203-2)는 접속 제어 애플리케이션(212) 또는 타겟 애플리케이션(211-1, 211-2, 211-3)의 서비스 서버(205-1, 205-2)에 대한 접속을 차단할 수 있다.

[0056] 데이터 플로우 테이블(316)은 노드(201)와 게이트웨이(203-1, 203-2), 및 서비스 서버(205-1, 205-2) 사이에 세부적인 데이터 패킷이 전송되는 흐름(예: 데이터 플로우)을 관리하기 위한 테이블이다. 데이터 플로우는 TCP 세션, 애플리케이션, 또는 보다 세부적인 단위로 생성될 수 있다. 데이터 플로우 테이블(316)은 출발지로부터 전송된 데이터 패킷이 인가된 데이터 패킷인지를 식별하기 위한 애플리케이션 ID, 도착지 IP 주소, 및/또는 서비스 포트를 포함할 수 있다.

[0057] 도 3을 참조하면, 데이터 플로우 정보(350)는 데이터 플로우 ID(351)와 데이터 플로우가 제어 플로우에 종속되는 경우에는 제어 플로우 ID(352)를 포함할 수 있다. 또한, 데이터 플로우 정보(350)는 데이터 패킷의 출발지 IP와 도착지 IP 및 포트에 기반하여 데이터 패킷의 포워딩을 판단하기 위한 인가된 대상 정보(353), 데이터 플로우가 사용 가능한 유효한 상태인지 여부를 나타내는 상태 정보(355), 및/또는 데이터 플로우의 인증 만료 시각 정보를 나타내는 시간 정보(356)를 포함 할 수 있다. 또한, 데이터 플로우 정보(350)는 인가된 파일 IO 정보(354)를 더 포함할 수 있다. 인가된 파일 IO 정보(354)는 애플리케이션의 파일 IO 접근(예: 읽기(read), 쓰기(write)와 같은 접근)을 허용할 것인지에 대한 정보를 포함할 수 있다. 예를 들어, 인가된 파일 IO 정보(354)는 애플리케이션의 파일 수신을 위한 파일 IO 시 쓰기 가능 여부 및 쓰기 가능한 경로 정보를 포함할 수 있다. 예를 들어, 인가된 파일 IO 정보(354)는 애플리케이션의 파일 전송을 위한 파일 IO 시 읽기 가능 여부 및 읽기 가능한 경로 정보를 포함할 수 있다. 또한 예를 들어, 인가된 파일 IO 정보(354)는 데이터 플로우를 적용하는 애플리케이션의 파일 IO 허용 여부(예: 파일 IO 처리 예외 등)에 대한 정보를 포함한다.

[0058] 파일 IO 테이블(317)은 노드(201)에서 발생한 파일 IO를 기록하고 관리하기 위한 데이터베이스로, 파일 IO 테이블 정보(360)를 포함할 수 있다.

[0059] 도 3을 참조하면, 파일 IO 테이블 정보(360)는 파일 시스템 IO가 발생된 프로세스의 PID(361)와 애플리케이션을 식별하기 위한 애플리케이션 식별 정보(362)(예: 애플리케이션 명칭 및 실행 경로, 고유 정보 등)를 포함할 수 있다. 파일 IO 테이블 정보(360)는 데이터 플로우가 식별된 상태에서 파일 시스템 IO가 발생되었을 경우, 네트워크 접속 제어 정보와 연계를 위한 데이터 플로우 ID(363)를 포함할 수 있다. 파일 IO 테이블 정보(360)는 파일 시스템 IO가 어떠한 형태(예: 일기 및/또는 쓰기)로 발생하였는지에 대한 파일 IO 정보(364), 파일 시스템 IO가 발생한 경로, 파일명, 및/또는 파일을 관리하기 위한 고유 정보(예: 해쉬 및 시그니처, 헤더 정보 등)를 포함하는 파일 정보(365), 파일 IO가 종료되었는지 여부 및 허용되지 않은 파일 IO에 따른 파일 IO 해제 여부에 관한 정보를 포함하는 상태 정보(366), 및/또는 파일 IO가 발생한 시각을 나타내는 파일 IO 시간 정보(367)을 포함할 수 있다.

[0060] 도 4는 다양한 실시 예들에 따른 노드(201)의 기능적 블록도를 나타낸다. 도 4에 도시된 구성들 중 적어도 일부는 컨트롤러(202), 게이트웨이(203-1, 203-2), 또는 서비스 서버(205-1, 205-2)에 적용될 수 있다.

[0061] 도 4를 참조하면, 노드(201)는 프로세서(410), 메모리(420), 및 통신 회로(430)를 포함할 수 있다. 일 실시 예에 따르면, 노드(201)는 사용자 인터페이스를 제공하기 위하여 디스플레이(440)를 더 포함할 수 있다.

[0062] 프로세서(410)는 노드의 전반적인 동작을 제어할 수 있다. 다양한 실시 예들에서, 프로세서(410)는 하나의 프로세서 코어(single core)를 포함하거나, 복수의 프로세서 코어들을 포함할 수 있다. 예를 들면, 프로세서(410)는 듀얼 코어(dual-core), 쿼드 코어(quad-core), 헥사 코어(hexa-core) 등의 멀티 코어(multi-core)를 포함할 수 있다. 실시 예들에 따라, 프로세서(410)는 내부 또는 외부에 위치된 캐시 메모리(cache memory)를 더 포함할 수 있다. 실시 예들에 따라, 프로세서(410)는 하나 이상의 프로세서들로 구성될(configured with) 수 있다. 예를 들면, 프로세서(410)는, 애플리케이션 프로세서(application processor), 통신 프로세서(communication processor), 또는 GPU(graphical processing unit) 중 적어도 하나를 포함할 수 있다.

[0063] 프로세서(410)의 전부 또는 일부는 노드(201) 내의 다른 구성 요소(예를 들면, 메모리(420), 통신 회로(430), 또는 디스플레이(440))와 전기적으로(electrically) 또는 작동적으로(operatively) 결합(coupled with)되거나 연결될(connected to) 수 있다. 프로세서(410)는 다른 구성 요소들의 명령을 수신할 수 있고, 수신된 명령을 해석할 수 있으며, 해석된 명령에 따라 계산을 수행하거나 데이터를 처리할 수 있다. 프로세서(410)는 메모리(420), 통신 회로(430), 또는 디스플레이(440)로부터 수신되는 메시지, 데이터, 명령어, 또는 신호를 해석할 수 있고, 가공할 수 있다. 프로세서(410)는 수신된 메시지, 데이터, 명령어, 또는 신호에 기반하여 새로운 메시지, 데이터, 명령어, 또는 신호를 생성할 수 있다. 프로세서(410)는 가공되거나 생성된 메시지, 데이터, 명령어, 또는 신호를 메모리(420), 통신 회로(430), 또는 디스플레이(440)에게 제공할 수 있다.

[0064] 프로세서(410)는 프로그램에서 생성되거나 발생되는 데이터 또는 신호를 처리할 수 있다. 예를 들면, 프로세서(410)는 프로그램을 실행하거나 제어하기 위해 메모리(420)에게 명령어, 데이터 또는 신호를 요청할 수 있다. 프로세서(410)는 프로그램을 실행하거나 제어하기 위해 메모리(420)에게 명령어, 데이터, 또는 신호를 기록(또는 저장)하거나 갱신할 수 있다.

[0065] 메모리(420)는 노드를 제어하는 명령어, 제어 명령어 코드, 제어 데이터, 또는 사용자 데이터를 저장할 수 있다. 예를 들면, 메모리(420)는 애플리케이션 프로그램, OS(operating system), 미들웨어(middleware), 또는 디바이스 드라이버(device driver) 중 적어도 하나를 포함할 수 있다. 메모리(420)는 휘발성 메모리(volatile memory) 또는 불휘발성(non-volatile memory) 중 하나 이상을 포함할 수 있다. 휘발성 메모리는 DRAM(dynamic random access memory), SRAM(static RAM), SDRAM(synchronous DRAM), PRAM(phase-change RAM), MRAM(magnetic RAM), RRAM(resistive RAM), FeRAM(ferroelectric RAM) 등을 포함할 수 있다. 불휘발성 메모리는 ROM(read only memory), PROM(programmable ROM), EPROM(electrically programmable ROM), EEPROM(electrically erasable programmable ROM), 플래시 메모리(flash memory) 등을 포함할 수 있다. 메모리(420)는 하드 디스크 드라이브(HDD, hard disk drive), 솔리드 스테이트 디스크(SSD, solid state disk), eMMC(embedded multi media card), UFS(universal flash storage)와 같은 불휘발성 매체(medium)를 더 포함할 수 있다.

[0066] 일 실시 예에 따르면, 메모리(420)는 도 1의 타겟 애플리케이션(211-1, 211-2, 211-3) 및 접속 제어 애플리케이션(212)을 저장할 수 있다. 접속 제어 애플리케이션(212)은 게이트웨이(203-1, 203-2)와의 네트워크 접속 및 보안 세션(220-1, 220-2) 생성과, 컨트롤러(202)와의 제어 플로우 생성 및 갱신 기능을 수행할 수 있다. 이를 위하여 접속 제어 애플리케이션(212)은 하나 이상의 보안 모듈을 포함할 수 있다. 일 실시 예에서, 메모리(420)는 도 3의 제어 플로우 정보(340) 데이터 플로우 정보(350), 및 파일 IO 테이블 정보(360)를 저장할 수 있다.

[0067] 일 실시 예에서, 타겟 애플리케이션(211-1, 211-2, 211-3)은 게이트웨이(203-1, 203-2)와의 보안 세션(220-1, 220-2) 생성을 위하여 하나 이상의 보안 모듈을 포함할 수 있다.

[0068] 통신 회로(430)는 노드(201)와 외부 전자 장치(예: 컨트롤러(202) 또는 게이트웨이(203-1, 203-2)) 간의 유선 또는 무선 통신 연결의 수립, 및 수립된 연결을 통한 통신 수행을 지원할 수 있다. 일 실시 예에 따르면, 통신 회로(430)는 무선 통신 회로(예: 셀룰러 통신 회로, 근거리 무선 통신 회로, 또는 GNSS(global navigation satellite system) 통신 회로) 또는 유선 통신 회로(예: LAN(local area network) 통신 회로, 또는 전력선 통신 회로)를 포함하고, 그 중 해당하는 통신 회로를 이용하여 블루투스, WiFi direct 또는 IrDA(infrared data association) 같은 근거리 통신 네트워크 또는 셀룰러 네트워크, 인터넷, 또는 컴퓨터 네트워크와 같은 원거리

통신 네트워크를 통하여 외부 전자 장치와 통신할 수 있다. 상술한 여러 종류의 통신 회로(430)는 하나의 칩으로 구현되거나 또는 각각 별도의 칩으로 구현될 수 있다.

[0069] 디스플레이(440)는, 컨텐츠, 데이터, 또는 신호를 출력할 수 있다. 다양한 실시 예들에서, 디스플레이(440)는 프로세서(410)에 의해 가공된 이미지 데이터를 표시할 수 있다. 실시 예들에 따라, 디스플레이(440)는 터치 입력 등을 수신할 수 있는 복수의 터치 센서들(미도시)과 결합됨으로써, 일체형의 터치 스크린(touch screen)으로 구성될(configured with) 수도 있다. 디스플레이(440)가 터치 스크린으로 구성되는 경우, 복수의 터치 센서들은, 디스플레이(440) 위에 배치되거나, 디스플레이(440) 아래에 배치될 수 있다.

[0070] 도 5는 다양한 실시 예들에 따라 데이터 패킷의 전송을 제어하는 동작을 설명한다.

[0071] 도 5를 참조하면, 접속 제어 애플리케이션(212)은 노드(201)에 포함된 타겟 애플리케이션(211)으로부터 서비스 서버(205)에 대한 네트워크 접속 요청을 감지하고, 노드(201) 또는 접속 제어 애플리케이션(212)이 컨트롤러(202)와 접속된 상태인지 여부를 결정할 수 있다. 노드(201) 또는 접속 제어 애플리케이션(212)이 컨트롤러(202)와 접속된 상태가 아닌 경우, 접속 제어 애플리케이션(212)은 운영체제가 포함되는 커널이나 네트워크 드라이버에서 데이터 패킷의 전송을 차단할 수 있다. 접속 제어 애플리케이션(212)을 통해, 노드(201)는 OSI(open system interconnection) 7 계층 중 응용 계층에서 악의적인 애플리케이션의 접속을 사전에 차단할 수 있다.

[0072] 일 실시 예에서, 접속 제어 애플리케이션(212)이 컨트롤러(202)와 접속된 상태가 아니거나, 접속 제어 애플리케이션(212)이 게이트웨이(203)에 의하여 인증되지 않았거나, 또는 접속 제어 애플리케이션(212)과 게이트웨이(203) 간 보안 세션이 생성되지 않은 경우, 접속 제어 애플리케이션(212)으로부터 전송되는 데이터 패킷은 게이트웨이(203)에 의하여 차단되며 접속 제어 애플리케이션(212)은 컨트롤러(202)로 접속을 요청할 수 있을 뿐이다. 실시 예에 따라서, 접속 제어 애플리케이션(212)이 게이트웨이(203)에 의하여 인증되지 않아도 접속 제어 애플리케이션(212)으로부터 전송되는 데이터 패킷은 게이트웨이(203)에 의하여 차단되지 않는 경우가 존재할 수 있다.

[0073] 다른 실시 예에 따르면, 노드(201)에 접속 제어 애플리케이션(212)이 설치되지 않거나 악성 애플리케이션이 접속 제어 애플리케이션(212)의 제어를 우회하는 경우, 비인가된 데이터 패킷이 노드(201)로부터 전송될 수 있다. 이 경우, 네트워크의 경계에 존재하는 게이트웨이(203)는 인가되지 않은 보안 세션으로 수신되는 데이터 패킷 및 데이터 플로우가 존재하지 않는 데이터 패킷을 차단하므로 노드(201)로부터 송신된 데이터 패킷(예: TCP 세션 생성을 위한 데이터 패킷)은 서비스 서버(205)에 도달하지 않을 수 있다. 다시 말해, 노드(201)는 서비스 서버(205)로부터 격리될 수 있다.

[0074] 도 6은 다양한 실시 예들에 따른 컨트롤러 접속을 위한 신호 흐름도를 나타낸다.

[0075] 노드(201)가 네트워크에 접속하기 위해서는 컨트롤러(202)에 의하여 인가될 필요가 있으므로, 노드(201)의 접속 제어 애플리케이션(212)은 컨트롤러(202)에게 제어 플로우의 생성을 요청함으로써 노드(201)의 컨트롤러 접속을 시도할 수 있다.

[0076] 도 6을 참조하면, 동작 605에서, 노드(201)는 컨트롤러 접속 이벤트를 감지할 수 있다. 예를 들어, 노드(201) 내에서 접속 제어 애플리케이션(212)이 설치 및/또는 실행되면, 노드(201)는 컨트롤러(202)에 대한 접속이 요청됨을 감지할 수 있다.

[0077] 동작 610에서, 노드(201)는 컨트롤러(202)에게 컨트롤러 접속을 요청할 수 있다. 예를 들어, 접속 제어 애플리케이션(212)은 접속 제어 애플리케이션(212)의 식별 정보를 컨트롤러(202)에게 전송할 수 있다. 추가적으로, 접속 제어 애플리케이션(212)은 노드(201)의 식별 정보(예: 단말 ID, IP 주소, MAC 주소), 종류, 위치, 환경, 노드(201)가 속하는 네트워크의 식별 정보, 및/또는 네트워크 시스템에 의하여 자체적으로 생성된 임의의 식별 정보를 더 전송할 수 있다.

[0078] 동작 615에서, 컨트롤러(202)는 컨트롤러 접속을 요청한 대상(예: 접속 제어 애플리케이션(212) 및 노드(201))의 컨트롤러 접속 가능 여부를 확인(identify)할 수 있다. 일 실시 예에 따르면, 컨트롤러(202)는 노드(201)로부터 수신된 정보가 접속 정책 데이터베이스(311)에 포함되는지 여부 또는, 노드(201), 노드(201)가 속한 네트워크, 및/또는 접속 제어 애플리케이션(212)의 식별 정보가 블랙리스트 데이터베이스(314)에 포함되는지 여부 중 적어도 하나에 기반하여 컨트롤러 접속을 요청한 대상의 컨트롤러 접속 가능 여부를 확인할 수 있다.

[0079] 컨트롤러 접속을 요청한 대상의 컨트롤러 접속이 가능하면, 컨트롤러(202)는 노드(201)(또는, 접속 제어 애플리케이션(212))와 컨트롤러(202) 간 제어 플로우를 생성할 수 있다. 이 경우, 컨트롤러(202)는 난수 형태로 제어

플로우 식별 정보를 생성하고, 노드(201), 노드(201)가 속한 네트워크, 또는 접속 제어 애플리케이션(212) 중 적어도 하나의 식별 정보를 제어 플로우 테이블(315)에 저장할 수 있다. 제어 플로우 테이블(315)에 저장된 정보(예: 제어 플로우 정보(340))는 노드(201)의 사용자 인증, 노드(201)의 정보 업데이트, 노드(201)의 네트워크 접속을 위한 정책 확인, 및/또는 유효성 검사에 이용될 수 있다.

[0080] 동작 620에서, 컨트롤러(202)는 접속 정책 데이터베이스(311)에서 식별된 정보(예: 노드(201), 노드(201)가 속하는 출발지 네트워크 정보)와 대응되는 접속 정책을 확인할 수 있다. 컨트롤러(202)는 확인된 접속 정책에 기초하여 접속 가능한 애플리케이션의 화이트리스트 정보를 생성할 수 있다. 다른 실시 예에서, 컨트롤러(202)는 동작 620을 수행하지 않을 수 있다. 예를 들어, 접속을 요청한 대상의 컨트롤러 접속이 가능하지 않으면, 컨트롤러(202)는 동작 620을 수행하지 않을 수 있다.

[0081] 동작 625에서, 컨트롤러(202)는 컨트롤러 접속 요청에 대한 응답을 노드(201)에게 전송할 수 있다. 일 실시 예에서, 컨트롤러(202)는 컨트롤러 접속 요청에 대한 응답으로 제어 플로우 식별 정보가 포함된 제어 플로우 정보(340)를 노드(201)에게 전송할 수 있다. 일 실시 예에서, 컨트롤러(202)는, 동작 620의 수행을 통하여 생성된 화이트 리스트를 접속 제어 애플리케이션(212)에게 전송할 수 있다. 실시 예에 따라, 컨트롤러 접속을 요청한 대상이 접속 불가능하거나 블랙리스트에 포함된 경우, 컨트롤러(202)는 제어 플로우를 생성하지 않고, 컨트롤러 접속 요청에 대한 응답으로 컨트롤러 접속 불가를 통보할 수 있다.

[0082] 일 실시 예에서, 노드(201)는 수신된 응답에 따라서 결과값을 처리할 수 있다. 예를 들어, 접속 제어 애플리케이션(212)은 수신된 제어 플로우 식별 정보를 저장하고, 컨트롤러 접속이 완료됨을 나타내는 사용자 인터페이스 화면(미도시)을 사용자에게 표시할 수 있다. 컨트롤러 접속이 완료되면, 노드(201)의 서비스 서버(205)에 대한 네트워크 접속 요청은 컨트롤러(202)에 의하여 통제될 수 있다.

[0083] 실시 예에 따라서, 컨트롤러(202)는 노드(201)가 접속이 불가능한 것으로 결정할 수 있다. 예를 들어, 컨트롤러 접속 불가 정보가 수신되면 노드(201)는 컨트롤러 접속이 불가능함을 나타내는 사용자 인터페이스 화면을 사용자에게 출력할 수 있다. 예를 들어, 사용자 인터페이스 화면은 노드(201)의 접속이 차단됨을 나타내고, 관리자(예: 컨트롤러(202))를 통한 격리 해제를 가이드 하는 사용자 인터페이스를 포함할 수 있다.

[0084] 일 실시 예에서, 노드(201)의 접속 제어 애플리케이션(212), 컨트롤러(202) 및 게이트웨이(203)는 동작 630 내지 동작 650을 더 수행할 수 있다. 실시 예에 따라, 노드(201)의 접속 제어 애플리케이션(212), 컨트롤러(202) 및 게이트웨이(203)는 동작 630 내지 동작 650을 전부 또는 일부만 수행할 수 있다.

[0085] 동작 630에서, 접속 제어 애플리케이션(212)은 애플리케이션의 검사를 수행할 수 있다. 예를 들어, 접속 제어 애플리케이션(212)은 컨트롤러(202)로부터 화이트 리스트가 수신된 경우에, 화이트 리스트에 포함되어 있는 애플리케이션이 노드(201)에 설치되어 있는지 여부를 확인할 수 있다. 접속 제어 애플리케이션(212)은 노드(201)에 설치된 애플리케이션 중 화이트 리스트에 포함된 애플리케이션의 경우, 유효성 검사 정책에 따라 애플리케이션의 무결성 및 안정성 여부(애플리케이션 위조 및 변조 여부, 코드 사이닝 검사, 핑거 프린트 검사 중 적어도 하나)를 검사할 수 있다.

[0086] 동작 635에서, 접속 제어 애플리케이션(212)은 애플리케이션의 검사 결과를 컨트롤러(202)로 전송할 수 있다.

[0087] 컨트롤러(202)는 검사 결과에 기초하여 애플리케이션이 유효(valid)한지를 확인할 수 있다. 컨트롤러(202)는 애플리케이션이 유효하면, 노드(201)의 네트워크 접속을 허용하기 위해 노드(201)에 대한 접속 정책에서 노드(201)가 위치한 게이트웨이(203-1, 203-2)를 확인하고, 이후, 동작 640을 수행할 수 있다.

[0088] 동작 640에서, 컨트롤러(202)는 노드(201)가 네트워크 접속 요청 절차 없이 데이터 패킷을 게이트웨이(203-1, 203-2)를 통해 서비스 서버(205-1, 205-2)로 전송할 수 있도록 출발지 IP, 도착지 IP 및 포트 정보에 기반하여 데이터 플로우를 생성할 수 있다.

[0089] 컨트롤러(202)는 데이터 플로우 생성 시 파일 IO 정책을 확인하고, 파일 IO 정책에 기초하여 인가된 파일 IO 정보(354)를 생성할 수 있다. 데이터 플로우의 데이터 플로우 정보(350)에는 생성된 인가된 파일 IO 정보(354)가 포함될 수 있다. 인가된 파일 IO 정보(354)는 데이터 플로우를 사용하는 애플리케이션의 파일 IO 접근과 관련된 정책 정보를 포함할 수 있다. 예를 들어, 인가된 파일 IO 정보(354)는 애플리케이션의 파일 IO 접근(예: 읽기(read), 쓰기(write)와 같은 접근)을 허용할 것인지에 대한 정보를 포함할 수 있다. 그리고, 인가된 파일 IO 정보(354)는 애플리케이션이 접근이 허용된 파일 경로, 및/또는 접근이 허용되지 않은 파일 경로에 대한 정보를 포함할 수 있다.

[0090] 동작 645에서, 컨트롤러(202)는 접속 제어 애플리케이션(212)의 애플리케이션 검사 결과 전송에 대한 응답을 전송할 수 있다. 예를 들어, 컨트롤러(202)는 인가된 파일 IO 정보(354)가 포함된 데이터 플로우 정보(350)를 접속 제어 애플리케이션(212)으로 전송할 수 있다. 또한, 동작 650에서, 컨트롤러(202)는 인가된 파일 IO 정보(354)가 포함된 데이터 플로우 정보(350)를 게이트웨이(203)로 전송할 수 있다.

[0091] 도 7은 다양한 실시 예들에 따른 사용자 인증을 위한 신호 흐름도를 도시하며, 도 8은 다양한 실시 예들에 따른 컨트롤러 접속을 위한 사용자 인터페이스 화면을 도시한다. 도 7에 도시된 동작들은 예를 들어, 도 6의 신호 흐름도 이후에 구현될 수 있다.

[0092] 노드(201)가 목적지 네트워크에 대한 상세한 접속 권한을 부여 받기 위해서, 노드(201)의 접속 제어 애플리케이션(212)은 컨트롤러(202)로부터 노드(201)의 사용자에 대한 인증을 받을 수 있다.

[0093] 도 7을 참조하면, 동작 705에서, 노드(201)는 사용자 인증을 위한 입력을 수신할 수 있다. 사용자 인증을 위한 입력은 예를 들어, 사용자 ID 및 비밀번호를 입력하는 사용자 입력일 수 있다. 다른 예를 들어, 사용자 인증을 위한 입력은 보다 강화된 인증을 위한 사용자 입력(예: 생체 정보)일 수 있다.

[0094] 일 예로, 도 8을 참조하면, 접속 제어 애플리케이션(212)이 실행되면 노드(201)는 컨트롤러 접속을 위하여 필요한 정보를 수신하기 위한 사용자 인터페이스 화면(810)을 표시할 수 있다. 사용자 인터페이스 화면(810)은 사용자 ID를 입력하기 위한 입력 창(811), 및/또는 비밀번호를 입력하기 위한 입력 창(812)을 포함할 수 있다. 입력 창들(811 및 812)에 정보가 입력된 후 사용자의 인증을 위한 버튼(813)에 대한 입력을 수신함으로써 노드(201)는 컨트롤러 접속 이벤트를 감지할 수 있다.

[0095] 동작 710에서, 노드(201)는 컨트롤러(202)에게 사용자 인증을 요청할 수 있다. 예를 들어, 접속 제어 애플리케이션(212)은 사용자 인증을 위한 입력 정보를 컨트롤러(202)에게 전송할 수 있다. 노드(201)와 컨트롤러(202) 간 제어 플로우가 이미 생성된 상태이면, 접속 제어 애플리케이션(212)은 사용자 인증을 위한 입력 정보를 제어 플로우 식별 정보와 함께 전송할 수 있다.

[0096] 동작 715에서, 컨트롤러(202)는 노드(201)로부터 수신된 정보에 기반하여 사용자를 인증할 수 있다. 예를 들어, 컨트롤러(202)는 수신된 정보에 포함된 사용자 ID, 비밀번호, 및/또는 강화된 인증 정보와, 컨트롤러(202)의 메모리에 포함된 데이터베이스(예: 도 2의 접속 정책 데이터베이스(311) 또는 블랙리스트 데이터베이스(314))에 기반하여 사용자가 접속 정책에 따라 접속 가능한지 여부 및 사용자가 블랙리스트에 포함되는지 여부를 결정할 수 있다.

[0097] 사용자가 인증되면, 컨트롤러(202)는 노드(201)가 송신한 제어 플로우 식별 정보에 기초하여 제어 플로우 테이블(315)에서 제어 플로우 정보(340)를 확인하고, 확인된 제어 플로우 정보(340)에 사용자의 식별 정보(예: 사용자 ID)를 추가할 수 있다. 추가된 사용자 식별 정보는 인증된 사용자의 컨트롤러 접속 또는 네트워크 접속에 이용될 수 있다.

[0098] 동작 720에서, 컨트롤러(202)는 접속 정책 데이터베이스(311)에서 식별된 정보(예: 노드(201), 노드(201)가 속하는 출발지 네트워크 정보)와 대응되는 접속 정책을 확인할 수 있다. 컨트롤러(202)는 확인된 접속 정책에 기초하여 접속 가능한 애플리케이션의 화이트 리스트 정보를 생성할 수 있다. 다른 실시 예에서, 컨트롤러(202)는 동작 720을 수행하지 않을 수 있다. 예를 들어, 접속을 요청한 대상의 컨트롤러 접속이 가능하지 않으면, 컨트롤러(202)는 동작 720을 수행하지 않을 수 있다.

[0099] 동작 725에서, 컨트롤러(202)는 사용자 인증 요청에 대한 응답을 노드(201)에게 전송할 수 있다. 일 실시 예에서, 컨트롤러(202)는 사용자 인증 요청에 대한 응답으로 사용자가 인증됨을 나타내는 정보를 노드(201)에게 전송할 수 있다. 일 실시 예에서, 컨트롤러(202)는 동작 720의 수행을 통하여 생성된 화이트 리스트를 접속 제어 애플리케이션(212)에게 전송할 수 있다. 실시 예에 따라, 컨트롤러 접속을 요청한 대상이 접속 불가능하거나 블랙리스트에 포함된 경우, 컨트롤러(202)는 사용자 인증 요청에 대한 응답으로 컨트롤러 접속 불가를 통보할 수 있다.

[0100] 일 실시 예에서, 노드(201)는 사용자 인증에 대한 결과값을 처리할 수 있다. 예를 들어, 노드(201)는 사용자 인증이 완료됨을 나타내는 사용자 인터페이스 화면을 디스플레이를 통해 사용자에게 출력할 수 있다.

[0101] 다른 실시 예에 따라 컨트롤러(202)는 사용자 인증이 불가능한 것으로 결정할 수 있다. 예를 들어, 사용자의 식별 정보가 블랙리스트 데이터베이스에 포함되면 컨트롤러(202)는 사용자 인증이 불가능한 것으로 결정할 수 있다. 이 경우, 동작 725에서 컨트롤러(202)는 사용자 인증이 불가능함을 나타내는 정보를 노드(201)에게 전송할

수 있고, 노드(201)는 사용자 인증이 실패함을 나타내는 사용자 인터페이스 화면을 디스플레이를 통해 출력할 수 있다. 예를 들어, 도 8을 참조하면, 노드(201)는 접속 제어 애플리케이션(212)을 통해 사용자 인터페이스 화면(820)을 표시할 수 있다. 사용자 인터페이스 화면(820)은 노드(201)의 접속이 차단됨을 나타내고, 관리자(예: 컨트롤러(202))를 통한 격리 해제를 가이드 하는 사용자 인터페이스(825)를 포함할 수 있다.

[0102] 일 실시 예에서, 노드(201)의 접속 제어 애플리케이션(212), 컨트롤러(202) 및 게이트웨이(203)는 동작 730 내지 동작 750을 더 수행할 수 있다. 실시 예에 따라, 노드(201)의 접속 제어 애플리케이션(212), 컨트롤러(202) 및 게이트웨이(203)는 동작 730 내지 동작 750을 전부 또는 일부 수행할 수 있다.

[0103] 동작 730에서, 접속 제어 애플리케이션(212)은 애플리케이션의 검사를 수행할 수 있다. 예를 들어, 접속 제어 애플리케이션(212)은 컨트롤러(202)로부터 화이트 리스트가 수신된 경우에, 화이트 리스트에 포함되어 있는 애플리케이션이 노드(201)에 설치되어 있는지 여부를 확인할 수 있다. 접속 제어 애플리케이션(212)은 노드(201)에 설치된 애플리케이션 중 화이트 리스트에 포함된 애플리케이션의 경우, 유효성 검사 정책에 따라 애플리케이션의 무결성 및 안정성 여부(애플리케이션 위조 및 변조 여부, 코드 사이닝 검사, 핑거 프린트 검사 중 적어도 하나)를 검사할 수 있다.

[0104] 동작 735에서, 접속 제어 애플리케이션(212)은 애플리케이션의 검사 결과를 컨트롤러(202)로 전송할 수 있다.

[0105] 컨트롤러(202)는 검사 결과에 기초하여 애플리케이션이 유효한지를 확인(valid)할 수 있다. 컨트롤러(202)는 애플리케이션이 유효하면, 노드(201)의 네트워크 접속을 허용하기 위해 노드(201)에 대한 접속 정책에서 노드(201)가 위치한 게이트웨이(203-1, 203-2)를 확인하고, 이후, 동작 740을 수행할 수 있다.

[0106] 동작 740에서, 컨트롤러(202)는 노드(201)가 네트워크 접속 요청 절차 없이 데이터 패킷을 게이트웨이(203-1, 203-2)를 통해 서비스 서버(205-1, 205-2)로 전송할 수 있도록 출발지 IP, 도착지 IP 및 포트 정보에 기반하여 데이터 플로우를 생성할 수 있다.

[0107] 컨트롤러(202)는 데이터 플로우 생성 시 파일 IO 정책을 확인하고, 파일 IO 정책에 기초하여 인가된 파일 IO 정보(354)를 생성할 수 있다. 데이터 플로우의 데이터 플로우 정보(350)에는 생성된 인가된 파일 IO 정보(354)가 포함될 수 있다. 인가된 파일 IO 정보(354)는 데이터 플로우를 사용하는 애플리케이션의 파일 IO 접근과 관련된 정책 정보를 포함할 수 있다. 예를 들어, 인가된 파일 IO 정보(354)는 애플리케이션의 파일 IO 접근(예: 읽기(read), 쓰기(write)와 같은 접근)을 허용할 것인지에 대한 정보를 포함할 수 있다. 그리고, 인가된 파일 IO 정보(354)는 애플리케이션이 접근이 허용된 파일 경로, 및/또는 접근이 허용되지 않은 파일 경로에 대한 정보를 포함할 수 있다.

[0108] 동작 745에서, 컨트롤러(202)는 접속 제어 애플리케이션(212)의 애플리케이션 검사 결과 전송에 대한 응답을 전송할 수 있다. 예를 들어, 컨트롤러(202)는 인가된 파일 IO 정보(354)가 포함된 데이터 플로우 정보(350)를 접속 제어 애플리케이션(212)으로 전송할 수 있다. 또한, 동작 750에서, 컨트롤러(202)는 인가된 파일 IO 정보(354)가 포함된 데이터 플로우 정보(350)를 게이트웨이(203)로 전송할 수 있다.

[0109] 도 9는 다양한 실시 예들에 따른 네트워크 접속을 위한 신호 흐름도를 도시하며, 도 9에 도시된 동작들은 예를 들어, 도 6 또는 도 7의 신호 흐름도 이후에 구현될 수 있다.

[0110] 도 9를 참조하면, 동작 905에서, 노드(201)는 네트워크 접속 이벤트를 감지할 수 있다. 예를 들어, 노드(201)는 접속 제어 애플리케이션(212)을 통해 타겟 애플리케이션이 서비스 서버(205)로의 접속을 시도함을 감지할 수 있다.

[0111] 일 실시 예에서, 접속 제어 애플리케이션(212)은 네트워크 접속 이벤트가 감지되면, 데이터 플로우를 검사할 수 있다. 예를 들어, 접속 제어 애플리케이션(212)은 타겟 애플리케이션의 식별 정보, 도착지 IP 및/또는 포트 정보에 대응하는 데이터 플로우가 존재하는지 여부를 확인할 수 있다. 또한, 접속 제어 애플리케이션(212)은 데이터 플로우가 존재하더라도 해당 데이터 플로우가 유효한지 여부를 확인할 수 있다. 일 예로, 접속 제어 애플리케이션(212)은 유효성 검사 정책에 따라서 접속 제어 애플리케이션(212) 및 타겟 애플리케이션의 무결성 및 안전성 검사(예: 애플리케이션의 위조, 변조 여부, 코드 사이닝 검사, 핑거프린트 검사)를 수행하고 및 컨트롤러(202)로부터 수신된 접속 정책에 따라 노드(201)에 설치된 타겟 애플리케이션의 도착지 IP 및 포트에 대한 접속 가능 여부를 확인할 수 있다.

[0112] 유효한 데이터 플로우가 존재한다면, 접속 제어 애플리케이션(212)은 이하의 동작들을 생략하고 보안 세션을 통해 게이트웨이(203)로 타겟 애플리케이션의 데이터 패킷을 전송할 수 있다.

[0113] 데이터 플로우가 존재하지만 유효하지 않은 경우, 접속 제어 애플리케이션(212)은 데이터 패킷을 전송하지 않고 네트워크 접속이 실패함을 나타내는 사용자 인터페이스 화면을 출력할 수 있다.

[0114] 데이터 플로우가 존재하지 않거나, 데이터 플로우의 인증 시간이 만료되어 갱신이 필요한 경우, 또는 다른 사유에 의하여 데이터 플로우의 갱신이 필요한 경우, 접속 제어 애플리케이션(212)은 컨트롤러(202)에게 네트워크 접속을 요청할 수 있다. 실시 예에 따라, 접속 제어 애플리케이션(212)은 네트워크 접속을 요청하기 전 유효성 검사 정책에 따라서 접속 제어 애플리케이션(212) 및 타겟 애플리케이션의 무결성 및 안전성 검사를 수행하고, 애플리케이션의 무결성 및 안정성이 확인되면 컨트롤러(202)에게 네트워크 접속을 요청할 수 있다.

[0115] 동작 910에서, 접속 제어 애플리케이션(212)은 컨트롤러(202)에게 서비스 서버(205)로의 네트워크 접속을 요청할 수 있다. 이 경우, 접속 제어 애플리케이션(212)은 타겟 애플리케이션 식별 정보 및 서비스 서버(205)의 식별 정보(예: IP 및 서비스 포트)를 제어 플로우의 식별 정보와 함께 컨트롤러(202)에게 전송할 수 있다.

[0116] 동작 915에서, 컨트롤러(202)는 접속 제어 애플리케이션(212)으로부터 수신된 요청에 응답하여 타겟 애플리케이션과 관련된 접속 정책 및 파일 IO 정책을 확인할 수 있다.

[0117] 컨트롤러(202)는 타겟 애플리케이션이 접속 정책 데이터베이스(311)의 접속 정책을 만족하는지를 확인할 수 있다. 예를 들어, 컨트롤러(202)는 네트워크 접속을 요청한 대상(예: 타겟 애플리케이션)과 요청된 대상(예: 서비스 서버(205))의 식별 정보가 접속 정책 데이터베이스(311)에 포함되는지 여부 및 네트워크 접속을 요청한 대상의 게이트웨이 접속 가능 여부를 확인할 수 있다.

[0118] 타겟 애플리케이션의 접속이 불가능 하면, 컨트롤러(202)는 동작 920에서 노드(201)에게 접속이 불가능함을 나타내는 정보를 전송할 수 있다. 이 경우, 접속 제어 애플리케이션(212)은 접속이 불가능함을 나타내는 사용자 인터페이스 화면을 디스플레이를 통해 출력할 수 있다.

[0119] 타겟 애플리케이션의 접속이 가능하면, 컨트롤러(202)는 접속을 요청한 대상(예: 타겟 애플리케이션)과 관련된 인가된 파일 IO 정보(354)를 확인하고, 인가된 파일 IO 정보(354)에 기반하여 접속을 요청한 대상(예: 타겟 애플리케이션)의 파일 IO 접근의 허용 여부를 확인할 수 있다. 파일 IO 접근의 허용 여부의 확인은 애플리케이션의 파일 IO 접근(예: 읽기(read), 쓰기(write)와 같은 접근)을 허용할 것인지에 대한 확인, 또는 애플리케이션이 접근이 허용된 파일 경로, 및/또는 접근이 허용되지 않은 파일 경로에 대한 확인을 포함할 수 있다.

[0120] 컨트롤러(202)는 노드(201)의 네트워크 접속을 허용하기 위해 데이터 플로우 테이블(316)에서 해당 도착지 IP와 포트로 접속 가능한 데이터 플로우가 존재하는지 확인할 수 있다.

[0121] 유효한 데이터 플로우가 없는 경우, 컨트롤러(202)는 대상(예: 타겟 애플리케이션)이 노드(201)와 서비스 서버(205-1, 205-2) 사이의 네트워크에 대한 접속을 허용하는 데이터 플로우를 생성할 수 있다. 컨트롤러(202)는 출발지 IP, 도착지 IP 및 포트 정보, 인가된 파일 IO 정보(354)와 같은 정보에 기반하여 데이터 플로우 정보(350)를 생성할 수 있다. 여기에서, 인가된 파일 IO 정보(354)는 파일 IO 정책에 따라 생성된 것일 수 있다.

[0122] 컨트롤러(202)는 데이터 플로우 생성 시 파일 IO 정책을 확인하고, 파일 IO 정책에 기초하여 인가된 파일 IO 정보(354)를 생성할 수 있다. 데이터 플로우의 데이터 플로우정보(350)에는 생성된 인가된 파일 IO 정보(354)가 포함될 수 있다. 인가된 파일 IO 정보(354)는 데이터 플로우를 사용하는 애플리케이션의 파일 IO 접근과 관련된 정책 정보를 포함할 수 있다. 예를 들어, 인가된 파일 IO 정보(354)는 애플리케이션의 파일 IO 접근(예: 읽기(read), 쓰기(write)와 같은 접근)을 허용할 것인지에 대한 정보를 포함할 수 있다. 그리고, 인가된 파일 IO 정보(354)는 애플리케이션이 접근이 허용된 파일 경로, 및/또는 접근이 허용되지 않은 파일 경로에 대한 정보를 포함할 수 있다.

[0123] 동작 920에서, 컨트롤러(202)는 접속 제어 애플리케이션(212)의 네트워크 접속 요청에 대한 응답을 노드(201)에게 전송할 수 있다. 예를 들어, 컨트롤러(202)는, 동작 915의 수행을 통하여 생성된 데이터 플로우 정보(350)를 접속 제어 애플리케이션(212)에게 전송할 수 있다. 실시 예에 따라, 접속을 요청한 대상이 접속 불가능한 경우, 컨트롤러(202)는 네트워크 접속 요청에 대한 응답으로 네트워크 접속 불가를 통보할 수 있다.

[0124] 노드(201)는 수신된 응답에 따라서 결과값을 처리할 수 있다. 예를 들어, 접속 제어 애플리케이션(212)은 수신된 데이터 플로우 정보(350)를 저장하고, 네트워크 접속이 가능한 경우 파일 IO 테이블 검사를 수행할 수 있다. 다른 예를 들어, 네트워크 접속이 불가하다는 응답을 수신한 경우, 노드(201)는 타겟 애플리케이션의 데이터 패킷을 드롭(drop)할 수 있다. 또 다른 예를 들어, 네트워크 접속이 불가하다는 응답을 수신한 경우, 노드(201)는 타겟 애플리케이션의 파일 시스템 IO를 제거(또는, 파일 시스템 IO 핸들을 제거)할 수 있다.

[0125] 또한, 동작 925에서, 컨트롤러(202)는 데이터 플로우 정보(350)를 게이트웨이(203)로 전송할 수 있다.

[0126] 도 10은 다양한 실시 예들에 따른 파일 IO 접근 감시를 위한 노드(201)의 동작 흐름도를 도시하고, 도 11은 다양한 실시 예들에 따른 파일 IO 접근 감시 결과를 알리는 사용자 인터페이스 화면을 도시한다. 이하의 동작 흐름도는 노드(201) 또는 노드(201)에 설치된 접속 제어 애플리케이션(212)에 의하여 구현될 수 있다.

[0127] 접속 제어 애플리케이션(212)은 타겟 애플리케이션(211-1, 211-2, 211-3)이 사용 중인 파일 IO가 존재하는 경우, 타겟 애플리케이션(211-1, 211-2, 211-3)과 관련된 데이터 플로우의 인가된 파일 IO 정보(354)에 기초하여, 사용 중인 파일 IO의 동작을 관리할 수 있다. 파일 IO의 동작을 관리에는 파일 IO의 허용 및/또는 차단(또는, 제거)을 포함할 수 있다.

[0128] 도 10을 참조하면, 동작 1010에서, 접속 제어 애플리케이션(212)은 파일 IO 접근 이벤트를 감지할 수 있다. 접속 제어 애플리케이션(212)은 타겟 애플리케이션(211-1, 211-2, 211-3)(또는 이의 프로세스)이 파일 IO에 접근하면 파일 IO 접근 이벤트가 발생한 것으로 확인할 수 있다.

[0129] 동작 1020에서, 접속 제어 애플리케이션(212)은 파일 IO에 접근하는 프로세스의 PID를 확인하고, 해당 PID에 할당된 데이터 플로우가 존재하는지 확인할 수 있다.

[0130] 데이터 플로우가 존재하는 경우, 접속 제어 애플리케이션(212)은 동작 1030으로 진행할 수 있다. 데이터 플로우가 존재하지 않는 경우, 접속 제어 애플리케이션(212)은 동작 1035로 진행할 수 있다. 또한, 파일 IO에 접근하는 프로세스의 PID(361)가 존재하지 않는 경우, 접속 제어 애플리케이션(212)은 동작 1035로 진행할 수 있다.

[0131] 동작 1030에서, 데이터 플로우가 존재하는 경우, 접속 제어 애플리케이션(212)은 해당 프로세스가 접근하는 파일 IO의 종류를 확인할 수 있다.

[0132] 해당 프로세스가 접근하는 파일 IO의 종류가 쓰기(또는, 쓰기와 관련된 모든 파일 접근 종류)인 경우, 접속 제어 애플리케이션(212)은 동작 1041으로 진행할 수 있다. 해당 프로세스가 접근하는 파일 IO의 종류가 읽기(또는, 읽기와 관련된 모든 파일 접근 종류)인 경우, 접속 제어 애플리케이션(212)은 동작 1051으로 진행할 수 있다.

[0133] 해당 프로세스가 접근하는 파일 IO의 종류는 파일 하나 이상의 읽기 및/또는 하나 이상의 쓰기를 동시에 요구할 수 있으며, 다중의 파일 IO가 존재하는 경우 각각의 단계를 순차적으로 및/또는 병렬적으로 진행할 수 있다. 예를 들어, 동작 1041, 1043, 1045, 및 1047과 동작 1051, 1053, 1055, 및 1057은 순차적으로 및/또는 병렬적으로 수행될 수 있다.

[0134] 동작 1035에서, 데이터 플로우가 존재하지 않는 경우, 접속 제어 애플리케이션(212)은 파일 IO 정보를 파일 IO 테이블(317)에 저장할 수 있다. 예를 들어, 접속 제어 애플리케이션(212)은 해당 프로세스의 식별된 파일 IO 정보(예: 파일 위치, 파일명, 파일 IO의 종류(예: 파일 읽기, 또는 쓰기) 파일 IO 시간 정보)를 파일 IO 테이블(317)에 저장할 수 있다. 이후 해당 프로세스가 네트워크에 접속할 경우, 접속 제어 애플리케이션(212)은 파일 IO 테이블(317)에서 내역을 조회하여 데이터 패킷 검사를 할 수 있도록 한다.

[0135] 동작 1041에서, 접속 제어 애플리케이션(212)은 데이터 플로우 정보(350)에서 해당 애플리케이션(또는, 해당 애플리케이션의 프로세스)에게 파일 쓰기가 허용되는지를 확인할 수 있다.

[0136] 파일 쓰기가 허용되지 않는 경우, 접속 제어 애플리케이션(212)은 동작 1043으로 진행할 수 있다. 파일 쓰기가 허용되는 경우, 접속 제어 애플리케이션(212)은 동작 1047으로 진행할 수 있다.

[0137] 동작 1043에서, 데이터 플로우 정보(350)에서 해당 애플리케이션의 파일 쓰기를 허용하지 않는 경우, 접속 제어 애플리케이션(212)은 해당 애플리케이션의 파일 접근을 중단하고 파일 IO 테이블(317)에서 해당 파일 IO의 상태를 접근 중단으로 변경할 수 있다. 접속 제어 애플리케이션(212)은 PID(361)에 대응하는 파일 IO 테이블 정보(360)에서 파일 IO의 상태를 접근 중단으로 변경할 수 있다. 예를 들어, 접속 제어 애플리케이션(212)은 파일 핸들 제거 또는 파일 핸들 충돌, 파일 쓰기 버퍼 초기화 등을 통해 해당 애플리케이션의 파일 접근을 중단시킬 수 있다. 이를 위해, 접속 제어 애플리케이션(212)은 운영체제에서 제공하는 API(application program interface) 및 후킹, 해당 프로세스 강제 종료 또는 일련의 파일 접근 제어 중단 명령어 등을 포함할 수 있다.

[0138] 도 11을 참조하면, 노드(201)는 해당 애플리케이션의 파일 쓰기가 허용되지 않음을 나타내는 사용자 인터페이스 화면(1110)을 디스플레이를 통해 사용자에게 출력할 수 있다. 사용자 인터페이스 화면(1110)은 파일 다운로드가 차단됨을 나타내는 메시지를 포함하는 알림 창(1111)을 포함할 수 있다.

[0139] 동작 1045는 접속 제어 애플리케이션(212)이 파일 쓰기 중단 처리에 실패한 경우 진행될 수 있다. 동작 1045에서 접속 제어 애플리케이션(212)은 해당 데이터 플로우의 상태를 변경하고 네트워크 접속을 차단하여 네트워크로부터 유입되는 데이터 패킷을 차단할 수 있다.

[0140] 노드(201)는 파일 쓰기 중단 처리에 실패하여 네트워크 접속이 차단됨을 나타내는 사용자 인터페이스 화면(미도시)을 디스플레이를 통해 사용자에게 출력할 수 있다.

[0141] 동작 1047에서, 데이터 플로우 정보(350)에서 해당 애플리케이션의 파일 쓰기를 허용하는 경우, 접속 제어 애플리케이션(212)은 해당 애플리케이션의 파일 쓰기를 허용하고, 해당 애플리케이션은 파일 쓰기를 수행할 수 있다.

[0142] 동작 1051에서, 접속 제어 애플리케이션(212)은 데이터 플로우 정보(350)에서 해당 애플리케이션(또는, 해당 애플리케이션의 프로세스)에게 파일 읽기가 허용되는지를 확인할 수 있다.

[0143] 해당 애플리케이션에게 파일 읽기가 허용되지 않는 경우, 접속 제어 애플리케이션(212)은 동작 1053으로 진행할 수 있다. 해당 애플리케이션에게 파일 읽기가 허용되는 경우, 접속 제어 애플리케이션(212)은 동작 1057으로 진행할 수 있다.

[0144] 동작 1053에서, 접속 제어 애플리케이션(212)은 데이터 플로우 정보(350)에서 해당 애플리케이션의 파일 읽기를 허용하지 않는 경우, 해당 애플리케이션의 파일 접근을 중단하고 파일 IO 테이블에서 해당 파일 IO의 상태를 접근 중단으로 변경할 수 있다. 접속 제어 애플리케이션(212)은 PID(361)에 대응하는 파일 IO 테이블 정보(360)에서 파일 IO의 상태를 접근 중단으로 변경할 수 있다. 예를 들어, 접속 제어 애플리케이션(212)은 파일 핸들 제거 또는 파일 핸들 충돌, 파일 쓰기 버퍼 초기화 등을 통해 해당 애플리케이션의 파일 접근을 중단시킬 수 있다. 이를 위해, 접속 제어 애플리케이션(212)은 운영체제에서 제공하는 API 및 후킹, 해당 프로세스 강제 종료 또는 일련의 파일 접근 제어 중단 명령어 등을 포함할 수 있다.

[0145] 도 11을 참조하면, 노드(201)는 해당 애플리케이션의 파일 읽기가 허용되지 않음을 나타내는 사용자 인터페이스 화면(1120)을 디스플레이를 통해 사용자에게 출력할 수 있다. 사용자 인터페이스 화면(1120)은 파일 업로드가 차단됨을 나타내는 메시지를 포함하는 알림 창(1121)을 포함할 수 있다.

[0146] 동작 1055는 접속 제어 애플리케이션(212)이 파일 읽기 중단 처리에 실패한 경우 진행될 수 있다. 동작 1055에서 접속 제어 애플리케이션(212)은 해당 데이터 플로우의 상태를 변경하고 네트워크 접속을 차단하여 네트워크로부터 유입되는 데이터 패킷을 차단할 수 있다.

[0147] 도 11을 참조하면, 노드(201)는 파일 읽기 중단 처리에 실패하여 네트워크 접속이 차단됨을 나타내는 사용자 인터페이스 화면(1130)을 디스플레이를 통해 사용자에게 출력할 수 있다. 사용자 인터페이스 화면(1130)은 네트워크 접속이 종료됨을 나타내는 메시지를 포함하는 알림 창(1131)을 포함할 수 있다.

[0148] 동작 1057에서, 데이터 플로우 정보(350)에서 해당 애플리케이션의 파일 읽기를 허용하는 경우, 접속 제어 애플리케이션(212)은 해당 애플리케이션의 파일 읽기를 허용하고, 해당 애플리케이션은 파일 읽기를 수행할 수 있다.

[0149] 도 12는 다양한 실시 예들에 따른 파일 IO 테이블 검사를 위한 노드(201)의 동작 흐름도를 도시한다. 이하의 동작 흐름도는 노드(201) 또는 노드(201)에 설치된 접속 제어 애플리케이션(212)에 의하여 구현될 수 있다.

[0150] 도 12를 참조하면, 동작 1210에서, 접속 제어 애플리케이션(212)은 파일 IO 테이블 검사 이벤트를 감지할 수 있다. 접속 제어 애플리케이션(212)은 네트워크 접속이 발생된 데이터 플로우가 감지되는 경우, 파일 IO 테이블 검사 이벤트를 감지할 수 있다. 접속 제어 애플리케이션(212)은 데이터 플로우 정보(350)가 획득(또는, 갱신)되는 경우를 파일 IO 테이블 검사 이벤트로서 감지할 수 있다.

[0151] 동작 1220에서, 접속 제어 애플리케이션(212)은 파일 IO 테이블(317)을 검사하여, 데이터 플로우에 대해서 파일 IO를 사용 중인 타겟 애플리케이션(211-1, 211-2, 211-3)(또는 이의 프로세스)이 존재하는지를 확인할 수 있다. 접속 제어 애플리케이션(212)은 파일 IO 테이블(317)에서 타겟 애플리케이션 식별 정보 중 하나인 PID(361)에 해당하는 파일 IO 테이블 정보(360)를 검색할 수 있다.

[0152] 파일 IO를 사용 중인 애플리케이션이 존재하는 경우, 접속 제어 애플리케이션(212)은 동작 1230으로 진행할 수 있다. 애플리케이션이 파일 IO를 사용 중이 아닌 경우, 접속 제어 애플리케이션(212)은 동작 1235로 진행할 수 있다.

[0153] 동작 1230에서, 접속 제어 애플리케이션(212)은 현재 사용 중인 파일 IO가 존재하는 경우, 해당 프로세스가 접근하는 파일 IO의 종류를 확인할 수 있다.

[0154] 해당 프로세스가 접근하는 파일 IO의 종류가 쓰기(또는, 쓰기와 관련된 모든 파일 접근 종류)인 경우, 접속 제어 애플리케이션(212)은 동작 1241으로 진행할 수 있다. 해당 프로세스가 접근하는 파일 IO의 종류가 읽기(또는, 읽기와 관련된 모든 파일 접근 종류)인 경우, 접속 제어 애플리케이션(212)은 동작 1251으로 진행할 수 있다.

[0155] 해당 프로세스가 접근하는 파일 IO의 종류는 파일 하나 이상의 읽기 및/또는 하나 이상의 쓰기를 동시에 요구할 수 있으며, 다중의 파일 IO가 존재하는 경우 각각의 단계를 순차적으로 및/또는 병렬적으로 진행할 수 있다. 예를 들어, 동작 1241, 1243, 1245, 1247 및 1249와 동작 1251, 1253, 1255, 1257, 및 1259는 순차적으로 및/또는 병렬적으로 수행될 수 있다.

[0156] 동작 1235에서, 접속 제어 애플리케이션(212)은 현재 사용 중인 파일 IO가 존재하지 않는 경우, 데이터 패킷을 전송할 수 있다.

[0157] 동작 1241에서, 접속 제어 애플리케이션(212)은 데이터 플로우 정보(350)에서 해당 애플리케이션(또는, 해당 애플리케이션의 프로세스)에게 파일 쓰기가 허용되는지를 확인할 수 있다.

[0158] 해당 애플리케이션에게 파일 쓰기가 허용되지 않는 경우, 접속 제어 애플리케이션(212)은 동작 1243으로 진행할 수 있다. 해당 애플리케이션에게 파일 쓰기가 허용되는 경우, 접속 제어 애플리케이션(212)은 동작 1247으로 진행할 수 있다.

[0159] 동작 1243에서, 접속 제어 애플리케이션(212)은 데이터 플로우 정보(350)에서 해당 애플리케이션의 파일 쓰기를 허용하지 않는 경우, 접속 제어 애플리케이션(212)은 해당 애플리케이션의 파일 접근을 중단하고 파일 IO 테이블(317)에서 해당 파일 IO의 상태를 접근 중단으로 변경할 수 있다. 접속 제어 애플리케이션(212)은 PID(361)에 대응하는 파일 IO 테이블 정보(360)에서 파일 IO의 상태를 접근 중단으로 변경할 수 있다. 예를 들어, 접속 제어 애플리케이션(212)은 파일 핸들 제거 또는 파일 핸들 충돌, 파일 쓰기 버퍼 초기화 등을 통해 해당 애플리케이션의 파일 접근을 중단시킬 수 있다. 이를 위해, 접속 제어 애플리케이션(212)은 운영체제에서 제공하는 API 및 후킹, 해당 프로세스 강제 종료 또는 일련의 파일 접근 제어 중단 명령어 등을 포함할 수 있다.

[0160] 동작 1245에서, 접속 제어 애플리케이션(212)은 해당 데이터 플로우의 상태를 변경하고 네트워크 접속을 차단하여 네트워크로부터 유입되는 데이터 패킷을 차단할 수 있다. 접속 제어 애플리케이션(212)은 전송 중인 데이터 패킷이 존재하는 경우, 해당 데이터 패킷을 드롭할 수 있다.

[0161] 동작 1247에서, 접속 제어 애플리케이션(212)은 해당 애플리케이션의 파일 쓰기를 허용하는 경우, 접속 제어 애플리케이션(212)은 해당 애플리케이션의 파일 쓰기를 허용할 수 있다.

[0162] 동작 1249에서, 접속 제어 애플리케이션(212)은 데이터 패킷을 네트워크로 전송할 수 있다.

[0163] 동작 1251에서, 접속 제어 애플리케이션(212)은 데이터 플로우 정보(350)에서 해당 애플리케이션(또는, 해당 애플리케이션의 프로세스)에게 파일 읽기가 허용되는지를 확인할 수 있다.

[0164] 해당 애플리케이션에게 파일 읽기가 허용되지 않는 경우, 접속 제어 애플리케이션(212)은 동작 1253으로 진행할 수 있다. 해당 애플리케이션에게 파일 읽기가 허용되는 경우, 접속 제어 애플리케이션(212)은 동작 1257으로 진행할 수 있다.

[0165] 동작 1253에서, 접속 제어 애플리케이션(212)은 데이터 플로우 정보(350)에서 해당 애플리케이션의 파일 읽기를 허용하지 않는 경우, 해당 애플리케이션의 파일 접근을 중단하고 파일 IO 테이블에서 해당 파일 IO의 상태를 접근 중단으로 변경할 수 있다. 접속 제어 애플리케이션(212)은 PID(361)에 대응하는 파일 IO 테이블 정보(360)에서 파일 IO의 상태를 접근 중단으로 변경할 수 있다. 예를 들어, 접속 제어 애플리케이션(212)은 파일 핸들 제거 또는 파일 핸들 충돌, 파일 쓰기 버퍼 초기화 등을 통해 해당 애플리케이션의 파일 접근을 중단시킬 수 있다. 이를 위해, 접속 제어 애플리케이션(212)은 운영체제에서 제공하는 API 및 후킹, 해당 프로세스 강제 종료 또는 일련의 파일 접근 제어 중단 명령어 등을 포함할 수 있다.

[0166] 동작 1255에서, 접속 제어 애플리케이션(212)은 해당 데이터 플로우의 상태를 변경하고 네트워크 접속을 차단하여 네트워크로부터 유입되는 데이터 패킷을 차단할 수 있다. 접속 제어 애플리케이션(212)은 전송 중인 데이터 패킷이 존재하는 경우, 해당 데이터 패킷을 드롭할 수 있다.

[0167] 동작 1257에서, 접속 제어 애플리케이션(212)은 해당 애플리케이션의 파일 읽기를 허용하는 경우, 접속 제어 애플리케이션(212)은 해당 애플리케이션의 파일 읽기를 허용하고, 해당 애플리케이션은 파일 읽기를 수행할 수 있다.

[0168] 동작 1259에서, 접속 제어 애플리케이션(212)은 데이터 패킷을 네트워크로 전송할 수 있다.

[0169] 도 13은 다양한 실시 예들에 따른 데이터 패킷 전송을 위한 노드(201)의 동작 흐름도를 도시한다. 이하의 동작 흐름도는 노드(201) 또는 노드(201)에 설치된 접속 제어 애플리케이션(212)에 의하여 구현될 수 있다.

[0170] 도 13을 참조하면, 동작 1310에서, 접속 제어 애플리케이션(212)은 데이터 패킷 전송 이벤트를 감지할 수 있다. 접속 제어 애플리케이션(212)은 타겟 애플리케이션(211-1, 211-2, 211-3)(또는 이의 프로세스)이 요청하는 데이터 패킷 전송 요청을, 데이터 패킷 전송 이벤트로서 감지할 수 있다.

[0171] 동작 1320에서, 접속 제어 애플리케이션(212)은 네트워크 접속이 허용된 후 데이터 패킷 전송 이벤트가 감지되면, 전송되는 데이터 패킷 및 타겟 애플리케이션(211-1, 211-2, 211-3)을 식별하여 데이터 플로우가 존재하는지 여부를 검사할 수 있다. 데이터 패킷의 전송을 요청하는 타겟 애플리케이션(211-1, 211-2, 211-3)(또는, 이의 프로세스)를 식별하고, 식별된 타겟 애플리케이션(211-1, 211-2, 211-3)(또는, 이의 프로세스)의 PID를 확인하고, 해당 PID에 할당된 데이터 플로우가 존재하는지 확인할 수 있다.

[0172] 데이터 플로우가 존재하는 경우, 접속 제어 애플리케이션(212)은 동작 1330으로 진행할 수 있다. 데이터 플로우가 존재하지 않는 경우, 접속 제어 애플리케이션(212)은 동작 1335로 진행할 수 있다. 동작 1335에서, 접속 제어 애플리케이션(212)은 데이터 플로우가 존재하지 않는 경우 데이터 패킷을 드롭할 수 있다.

[0173] 동작 1330에서, 접속 제어 애플리케이션(212)은 데이터 플로우가 존재하는 경우, 파일 IO 테이블을 확인할 수 있다.

[0174] 동작 1340에서, 접속 제어 애플리케이션(212)은 데이터 플로우 정보(350)의 인가된 파일 IO 정보(354)에서 타겟 애플리케이션(또는, 이의 프로세스)의 파일 IO, 또는 읽기가 허용되는지를 확인할 수 있다. 파일 IO, 또는 읽기가 허용된 경우, 접속 제어 애플리케이션(212)은 동작 1355으로 진행할 수 있다. 파일 IO, 및 읽기가 허용되지 않은 경우, 접속 제어 애플리케이션(212)은 동작 1345로 진행할 수 있다. 동작 1355에서, 접속 제어 애플리케이션(212)은 데이터 패킷을 네트워크로 전송할 수 있다.

[0175] 동작 1345에서, 접속 제어 애플리케이션(212)은 파일 IO 테이블(317)에서 파일 IO가 존재하지 않는지를 확인할 수 있다. 접속 제어 애플리케이션(212)은 애플리케이션 식별 정보 중 하나인 PID를 이용하여 읽기 종류의 파일 IO가 존재하는지 여부, 또는 읽기 종류의 파일 IO로 접근 중인지 여부, 접근 후 사용을 종료한 파일 IO가 있는지 여부를 확인할 수 있다. 접속 제어 애플리케이션(212)은 타겟 애플리케이션(또는, 이의 프로세스)에 의해 수행되는 읽기 종류의 파일 IO가 존재하는지 여부, 또는 읽기 종류의 파일 IO로 접근 중인지 여부, 접근 후 사용을 종료한 파일 IO가 있는지 여부를 확인할 수 있다.

[0176] 파일 IO가 존재하지 않는 경우, 접속 제어 애플리케이션(212)은 동작 1355으로 진행할 수 있다. 파일 IO가 존재하는 경우, 접속 제어 애플리케이션(212)은 동작 1350로 진행할 수 있다. 동작 1355에서, 접속 제어 애플리케이션(212)은 데이터 패킷을 네트워크로 전송할 수 있다.

[0177] 동작 1350에서, 접속 제어 애플리케이션(212)은 파일 IO가 존재하는 경우, 데이터 패킷을 검사할 수 있다. 접속 제어 애플리케이션(212)은 파일 IO의 대상 파일을 확인하고, 대상 파일의 일부 내용 또는, 대상 파일을 관리하기 위한 고유 정보(예: 해쉬, 헤더 정보, 시그니처 정보 등)가 데이터 패킷에 포함되어 있는지를 검사할 수 있다. 접속 제어 애플리케이션(212)은 전송되는 데이터 패킷에서 파일의 일부 내용 또는, 파일을 관리하기 위한 고유 정보(예: 해쉬, 헤더 정보, 시그니처 정보 등)가 포함되어 있는지를 검사할 수 있다.

[0178] 동작 1360에서, 접속 제어 애플리케이션(212)은 데이터 패킷의 검사가 성공인지를 확인할 수 있다. 접속 제어 애플리케이션(212)은 데이터 패킷에 파일의 일부 내용 또는, 파일을 관리하기 위한 고유 정보(예: 해쉬, 헤더 정보, 시그니처 정보 등)가 포함되어 있는 경우, 데이터 패킷의 검사가 실패인 것으로 확인할 수 있다. 데이터 패킷의 검사가 성공인 경우, 접속 제어 애플리케이션(212)은 동작 1355으로 진행할 수 있다. 데이터 패킷의 검사가 실패인 경우(즉, 데이터 패킷에 파일의 일부 내용 또는, 파일을 관리하기 위한 고유 정보(예: 해쉬, 헤더 정보, 시그니처 정보 등)가 포함되지 않은 경우), 접속 제어 애플리케이션(212)은 동작 1350로 진행할 수 있다. 동작 1370에서, 접속 제어 애플리케이션(212)은 데이터 패킷을 네트워크로 전송할 수 있다.

[0179] 동작 1370에서, 접속 제어 애플리케이션(212)은 데이터 패킷을 드롭하고, 네트워크 접속을 차단할 수 있다.

[0180] 도 14는 다양한 실시 예들에 따른 제어 플로우 갱신을 위한 신호 흐름도를 도시한다.

[0181] 도 14는 참조하면, 동작 1405에서, 접속 제어 애플리케이션(212)은 제어 플로우 갱신을 요청할 수 있다.

[0182] 일 실시 예에서, 접속 제어 애플리케이션(212)은 제어 플로우 및 데이터 플로우를 유지하기 위하여 컨트롤러(202)에게 제어 플로우 갱신을 요청할 수 있다. 일 실시 예에서, 접속 제어 애플리케이션(212)은 파일 IO 테이블(317)에서 네트워크 접속을 통해 파일을 저장하였거나 읽은 경우, 파일 IO 접근이 차단된 내역을 전송하기 위해 제어 플로우를 갱신 요청할 수 있다. 또한, 접속 제어 애플리케이션(212)은 지정된 주기 마다 제어 플로우의 식별 정보를 포함하여 제어 플로우 갱신을 요청할 수 있다.

[0183] 동작 1410에서, 컨트롤러(202)는 제어 플로우를 갱신(검사)할 수 있다. 예를 들어, 컨트롤러(202)는 수신된 식별 정보가 나타내는 제어 플로우 정보(340)가 제어 플로우 테이블(315) 내에 존재하는지, 제어 플로우 정보(340)가 존재한다면 그에 종속되는 데이터 플로우 정보(350)가 존재하는지를 확인할 수 있다. 다른 보안 시스템에 의하여 접속이 해제되거나, 갱신 시간이 경과하거나, 또는 위험 탐지에 의하여 접속이 해제되는 경우, 제어 플로우가 존재하지 않을 수 있다.

[0184] 예를 들어 갱신(검사)이 실패하면, 컨트롤러(202)는 노드(201)는 접속 제어 애플리케이션(212)을 종료하거나 접속 제어 애플리케이션(212)의 네트워크 접속을 차단할 수 있다. 다른 예를 들어 갱신(검사)이 성공하면, 컨트롤러(202)는 제어 플로우 및 그에 종속되는 데이터 플로우의 만료 시간을 갱신할 수 있다. 이 경우, 노드(201)는 제어 플로우 정보(340) 및 데이터 플로우 정보(350)를 갱신할 수 있다.

[0185] 노드(201)로부터 수신된 파일 IO 테이블 정보(360)가 있는 경우, 컨트롤러(202)는 컨트롤러(202)에 존재하는 파일 IO 테이블(317)을 갱신하고, 갱신된 파일 IO 테이블(317)을 에 기반하여 사용자 및 노드(201)로부터의 행위 조회 및 감사 자료로 사용하며, 이상 행위가 발생한 경우 제어 플로우를 종료할 수 있다.

[0186] 동작 1415에서, 컨트롤러(202)는 갱신(검사) 결과 정보를 접속 제어 애플리케이션(212)에게 전송할 수 있다. 예를 들어 갱신(검사)이 실패하면, 컨트롤러(202)는 접속 불가 정보를 전송할 수 있다. 다른 예를 들어 갱신(검사)이 성공하면, 컨트롤러(202)는 갱신된 제어 플로우 정보(340) 및 그에 종속되는 데이터 플로우 정보(350)를 접속 제어 애플리케이션(212)에게 전송할 수 있다. 이 경우, 노드(201)는 제어 플로우 정보(340) 및 데이터 플로우 정보(350)를 갱신할 수 있다.

[0187] 접속 제어 애플리케이션(212)은 갱신(검사) 결과 정보를 처리할 수 있다.

[0188] 제어 플로우 갱신 결과가 접속 불가를 나타내면, 접속 제어 애플리케이션(212)은 타겟 애플리케이션을 종료하거나, 타겟 애플리케이션의 모든 네트워크 접속을 차단할 수 있다. 제어 플로우 갱신 결과가 정상을 나타내고, 갱신된 데이터 플로우 정보(350)가 있는 경우, 접속 제어 애플리케이션(212)은 접속 제어 애플리케이션(212) 내의 데이터 플로우 테이블(316)을 갱신할 수 있다.

[0189] 도 15는 다양한 실시 예들에 따른 제어 플로우 제거를 위한 신호 흐름도를 도시한다.

[0190] 도 15를 참조하면, 동작 1505에서 노드(201)는 컨트롤러(202)에게 제어 플로우 종료를 요청할 수 있다. 예를 들어, 노드(201)는 접속 제어 애플리케이션(212)을 종료하거나, 네트워크 접속이 지정된 시간 동안 사용되지 않거나, 또는 다른 시스템으로부터 접속 종료 요청이 수신되는 경우, 제어 플로우 종료를 요청할 수 있다.

[0191] 동작 1510에서, 컨트롤러(202)는 노드(201)로부터 수신된 식별 정보에 대응하는 제어 플로우를 제거할 수 있다. 컨트롤러(202)는 제거된 제어 플로우에 종속되는 데이터 플로우 또한 제거할 수 있다.

[0192] 동작 1515에서, 컨트롤러(202)는 게이트웨이(203)에게 데이터 플로우 제거를 요청하며, 동작 1520에서, 게이트웨이(203)는 데이터 플로우를 제거함으로써 접속 제어 애플리케이션(212)의 네트워크 접속을 차단할 수 있다.

[0193] 도 16은 다양한 실시 예들에 따른 데이터 플로우 제거를 위한 신호 흐름도를 도시한다.

[0194] 도 16을 참조하면, 동작 1605에서 접속 제어 애플리케이션(212)은 타겟 애플리케이션의 종료를 감지할 수 있다. 접속 제어 애플리케이션(212)은 노드(201)에서 실행 중인 타겟 애플리케이션의 종료 여부를 실시간 감시할 수 있다.

[0195] 동작 1610에서 접속 제어 애플리케이션(212)은 데이터 플로우 테이블(316)을 검사할 수 있다. 접속 제어 애플리케이션(212)은 데이터 플로우 테이블(316)에서 종료된 타겟 애플리케이션의 식별 정보 및 PID(또는, 자식

(child)의 PID)에 대응하는 데이터 플로우가 존재하는지를 확인할 수 있다. 타겟 애플리케이션의 데이터 플로우가 확인되면, 접속 제어 애플리케이션(212)은 데이터 플로우 테이블(316)에서 해당 데이터 플로우를 삭제할 수 있다.

[0196] 다중 실행 가능한 타겟 애플리케이션의 종료를 추적하기 위해, 접속 제어 애플리케이션(212)은 종료된 타겟 애플리케이션이 실행 중인 프로세스 목록에서 존재하지 않는 경우, 종료된 애플리케이션의 식별 정보에 대응하는 데이터 플로우를 데이터 플로우 테이블(316)에서 삭제할 수 있다.

[0197] 동작 1615에서 접속 제어 애플리케이션(212)은 컨트롤러(202)에게 데이터 플로우 종료를 요청할 수 있다. 예를 들어, 접속 제어 애플리케이션(212)은 동작 1610에서 삭제된 데이터 플로우 목록을 컨트롤러(202)로 전송하여 데이터 플로우 종료를 요청할 수 있다.

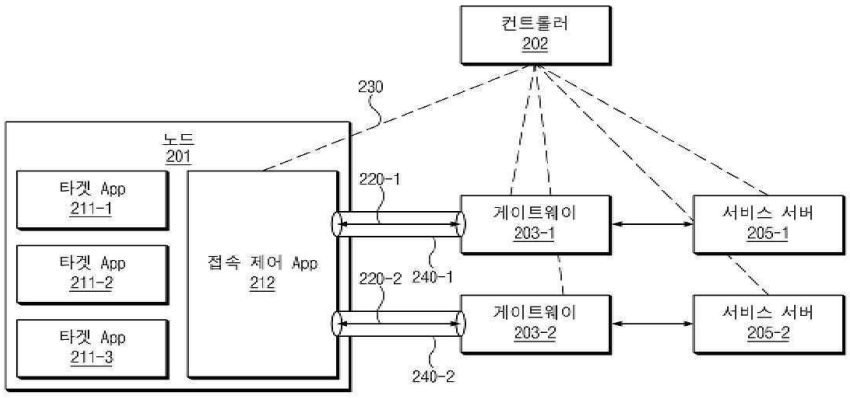
[0198] 동작 1620에서, 컨트롤러(202)는 노드(201)로부터 데이터 플로우 목록에 대응하는 데이터 플로우를 제거할 수 있다. 동작 1625에서, 컨트롤러(202)는 게이트웨이(203)에게 데이터 플로우 제거를 요청하며, 동작 1630에서, 게이트웨이(203)는 데이터 플로우를 제거함으로써 접속 제어 애플리케이션(212)의 제거된 데이터 플로우에 대응되는 서비스 서버(205)에 대한 접속을 차단할 수 있다.

[0199] 이상의 설명은 본 문서에 개시된 기술 사상을 예시적으로 설명한 것에 불과한 것으로서, 본 문서에 개시된 실시 예들이 속하는 기술 분야에서 통상의 지식을 가진 자라면 본 문서에 개시된 실시 예들의 본질적인 특성에서 벗어나지 않는 범위에서 다양한 수정 및 변형이 가능할 것이다.

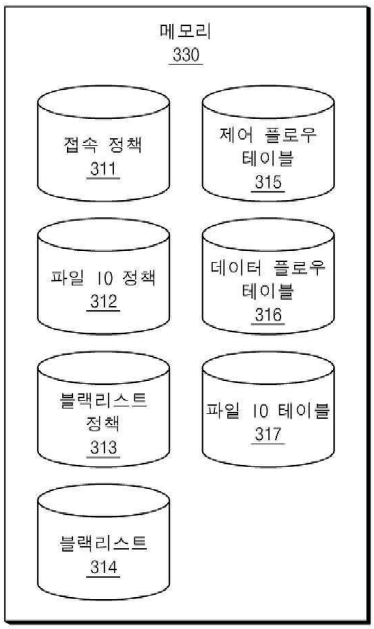
[0200] 따라서, 본 문서에 개시된 실시 예들은 본 문서에 개시된 기술 사상을 한정하기 위한 것이 아니라 설명하기 위한 것이고, 이러한 실시 예에 의하여 본 문서에 개시된 기술 사상의 범위가 한정되는 것은 아니다. 본 문서에 개시된 기술 사상의 보호 범위는 아래의 청구 범위에 의하여 해석되어야 하며, 그와 동등한 범위 내에 있는 모든 기술 사상은 본 문서의 권리범위에 포함되는 것으로 해석되어야 할 것이다.

도면

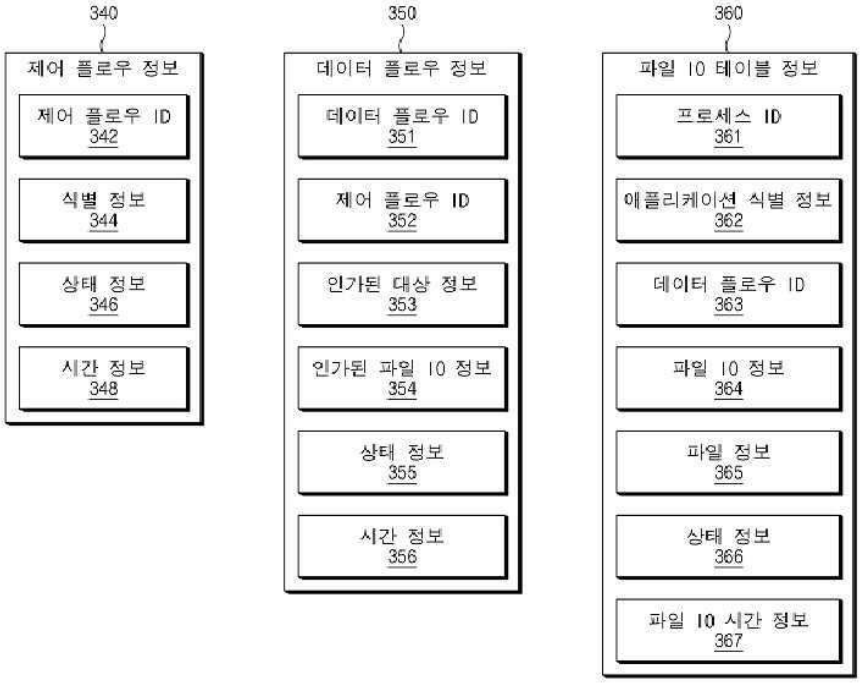
도면1



도면2


메모리
330
접속 정책
311
제어 플로우 테이블
315
파일 IO 정책
312
데이터 플로우 테이블
316
블랙리스트 정책
313
파일 IO 테이블
317
블랙리스트
314

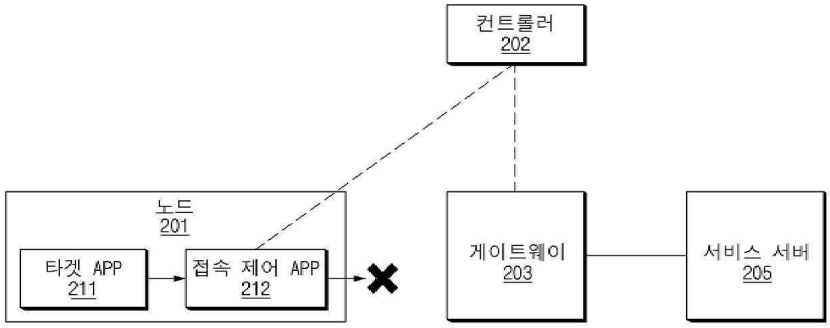
도면3


340
제어 플로우 정보
제어 플로우 ID
342
식별 정보
344
상태 정보
346
시간 정보
348
350
데이터 플로우 정보
데이터 플로우 ID
351
제어 플로우 ID
352
인가된 대상 정보
353
인가된 파일 IO 정보
354
상태 정보
355
시간 정보
356
360
파일 IO 테이블 정보
프로세스 ID
361
애플리케이션 식별 정보
362
데이터 플로우 ID
363
파일 IO 정보
364
파일 정보
365
상태 정보
366
파일 IO 시간 정보
367

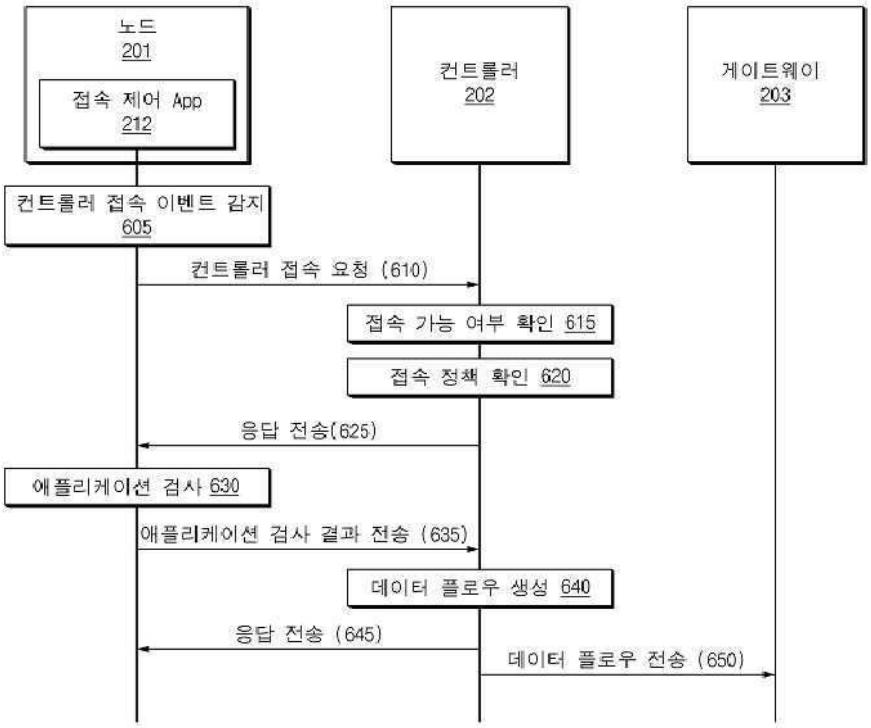
도면4


디스플레이
440
프로세서
410
통신회로
430
메모리
420

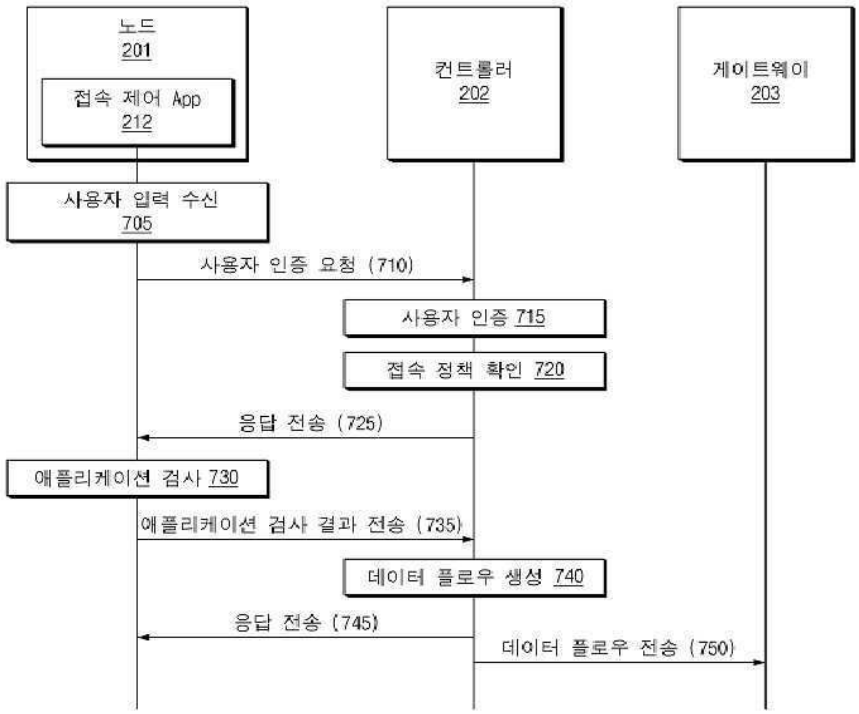
도면5


컨트롤러
202
노드
201
타겟 APP
211
접속 제어 APP
212
게이트웨이
203
서비스 서버
205

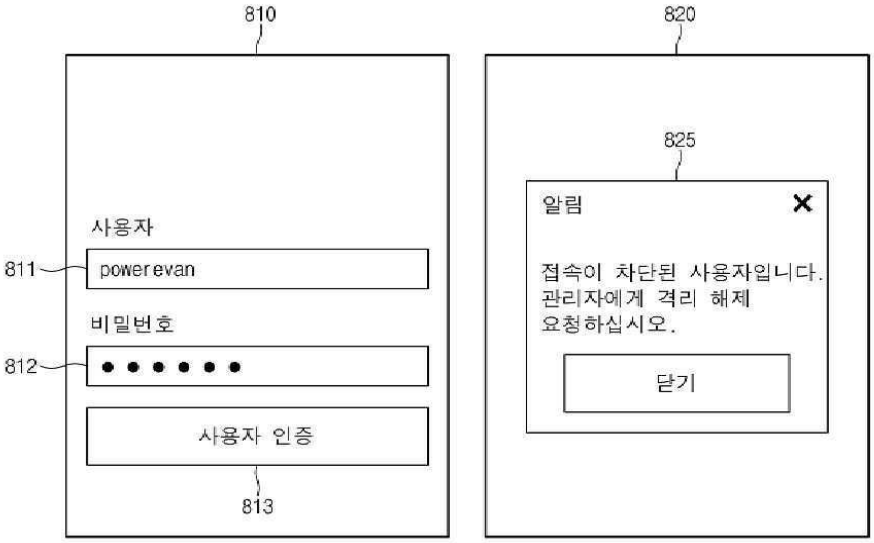
도면6


노드
201
접속 제어 App
212
컨트롤러
202
게이트웨이
203
컨트롤러 접속 이벤트 감지
605
컨트롤러 접속 요청 (610)
접속 가능 여부 확인 615
접속 정책 확인 620
응답 전송(625)
에플리케이션 검사 630
애플리케이션 검사 결과 전송 (635)
데이터 플로우 생성 640
응답 전송 (645)
데이터 플로우 전송 (650)

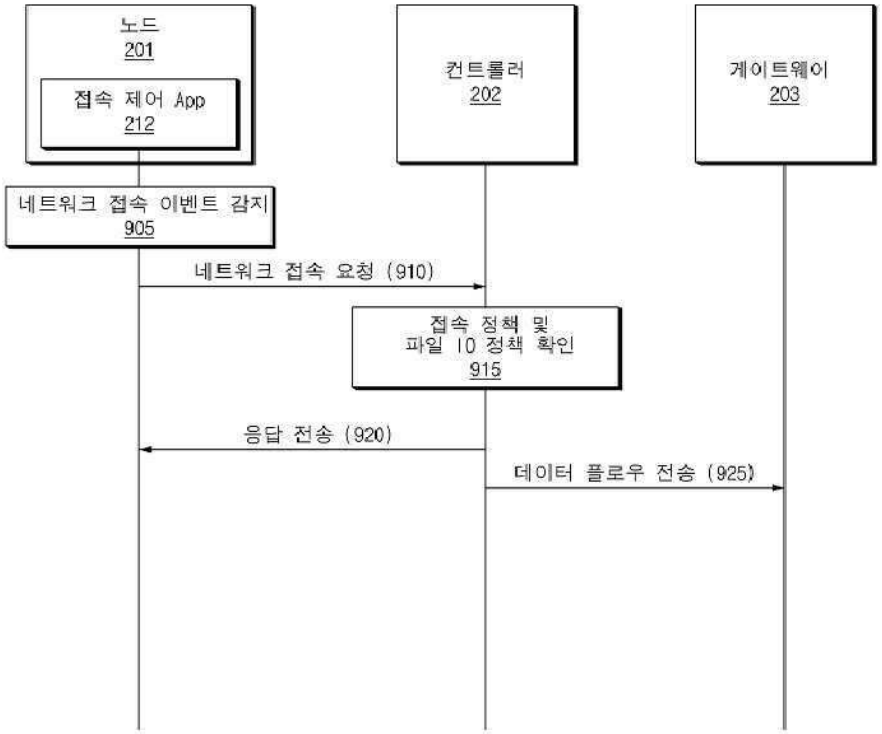
도면7


노드
201
접속 제어 App
212
컨트롤러
202
게이트웨이
203
사용자 입력 수신
705
사용자 인증 요청 (710)
사용자 인증 715
접속 정책 확인 720
응답 전송 (725)
애플리케이션 검사 730
애플리케이션 검사 결과 전송 (735)
데이터 플로우 생성 740
응답 전송 (745)
데이터 플로우 전송 (750)

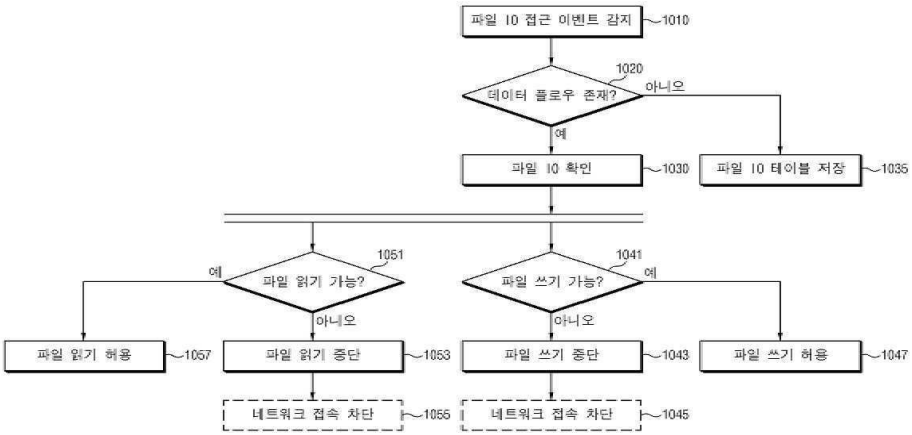
도면8


810
820
825
사용자
811
powerevan
비밀번호
812
사용자 인증
813
알림
접속이 차단된 사용자입니다.
관리자에게 격리 해제
요청하십시오.
닫기

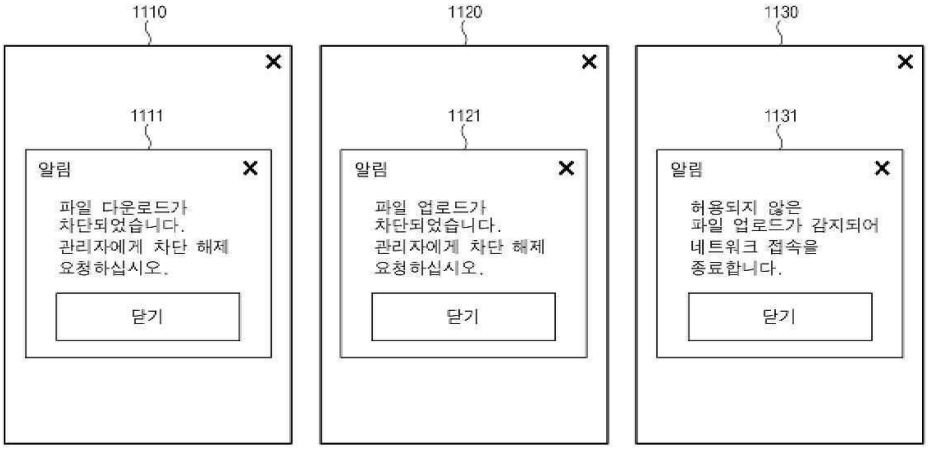
도면9


노드
201
접속 제어 App
212
컨트롤러
202
게이트웨이
203
네트워크 접속 이벤트 감지
905
네트워크 접속 요청 (910)
접속 정책 및
파일 IO 정책 확인
915
응답 전송 (920)
데이터 플로우 전송 (925)

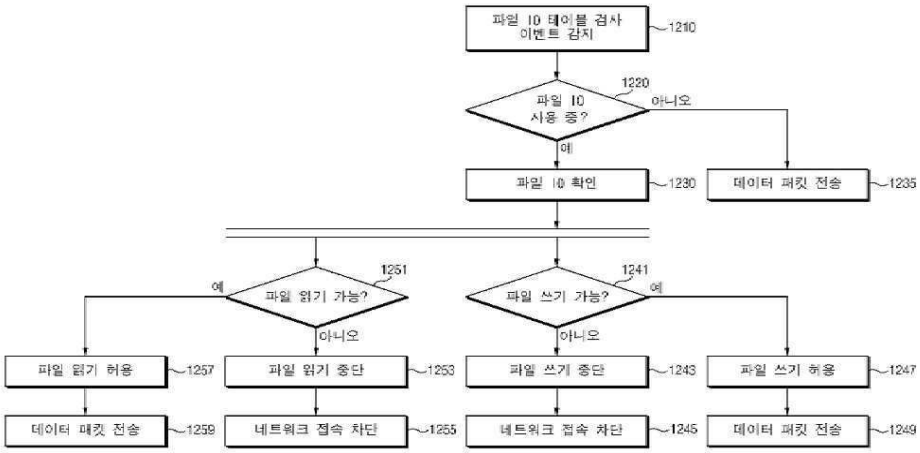
도면10


파일 IO 접근 이벤트 감지
1010
1020
데이터 플로우 존재?
아니오
예
파일 IO 확인
1030
파일 IO 테이블 저장
1035
1051
예
파일 읽기 가능?
아니오
1041
파일 쓰기 가능?
예
아니오
파일 읽기 허용
1057
파일 읽기 중단
1053
파일 쓰기 중단
1043
파일 쓰기 허용
1047
네트워크 접속 차단
1055
네트워크 접속 차단
1045

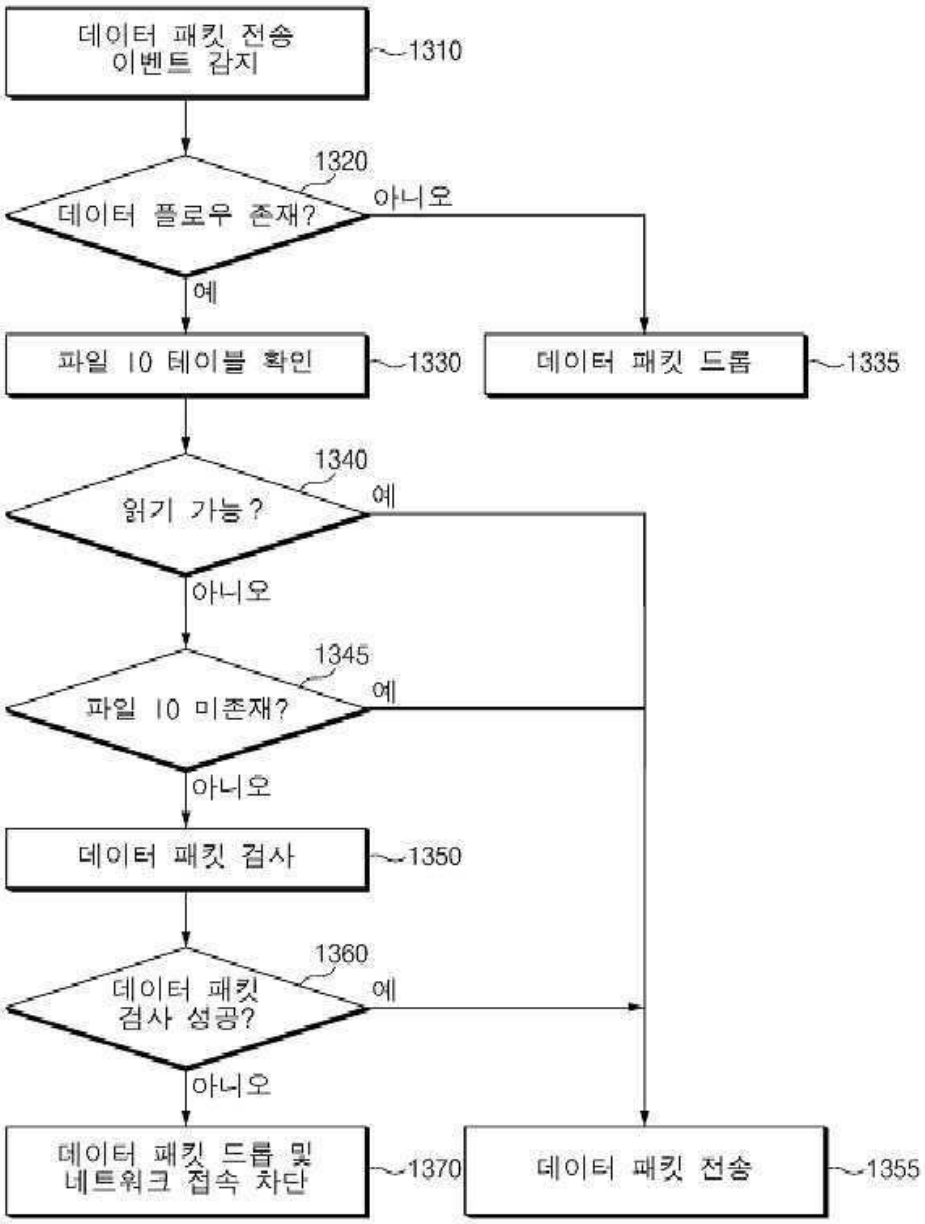
도면11


1110
1111
알림
파일 다운로드가
차단되었습니다.
관리자에게 차단 해제
요청하십시오.
닫기
1120
1121
알림
파일 업로드가
차단되었습니다.
관리자에게 차단 해제
요청하십시오.
닫기
1130
1131
알림
허용되지 않은
파일 업로드가 감지되어
네트워크 접속을
종료합니다.
닫기

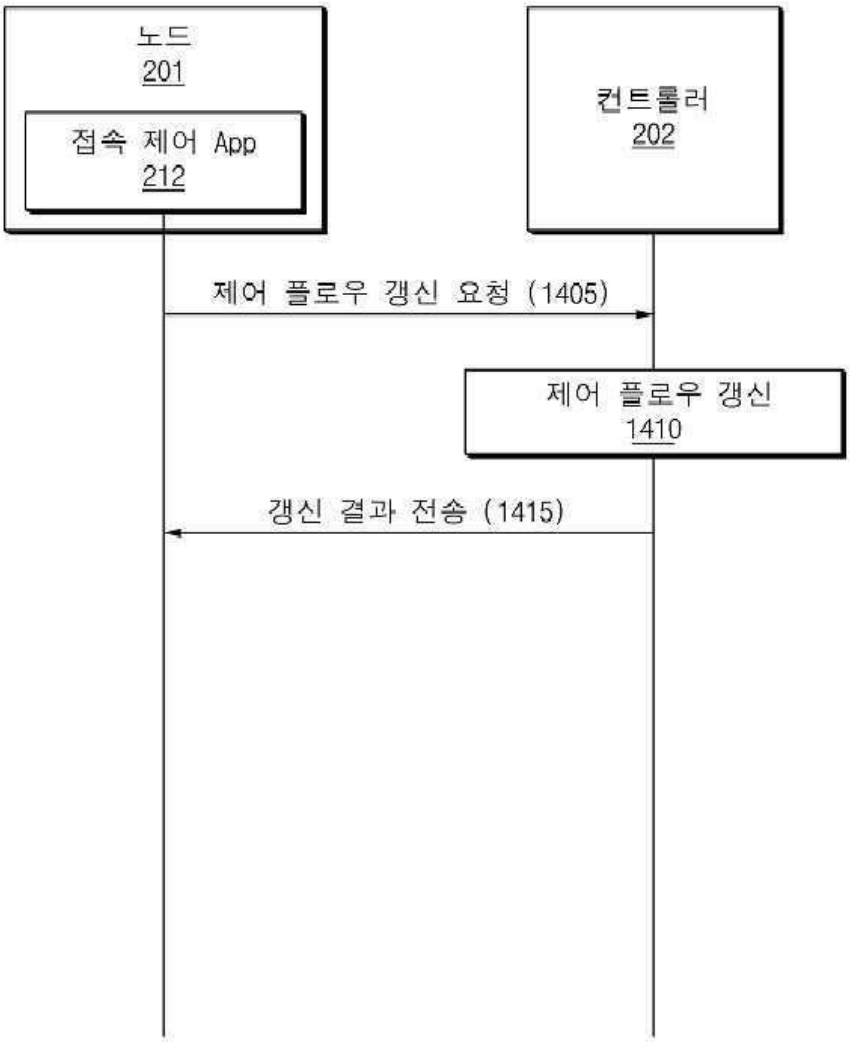
도면12


파일 IO 테이블 검사
이벤트 감지
1210
파일 IO
사용 중?
1220
아니오
예
파일 IO 확인
1230
데이터 패킷 전송
1235
파일 읽기 가능?
1251
예
아니오
파일 쓰기 가능?
1241
예
아니오
파일 읽기 허용
1257
파일 읽기 중단
1253
파일 쓰기 중단
1243
파일 쓰기 허용
1247
데이터 패킷 전송
1259
네트워크 접속 차단
1255
네트워크 접속 차단
1245
데이터 패킷 전송
1249

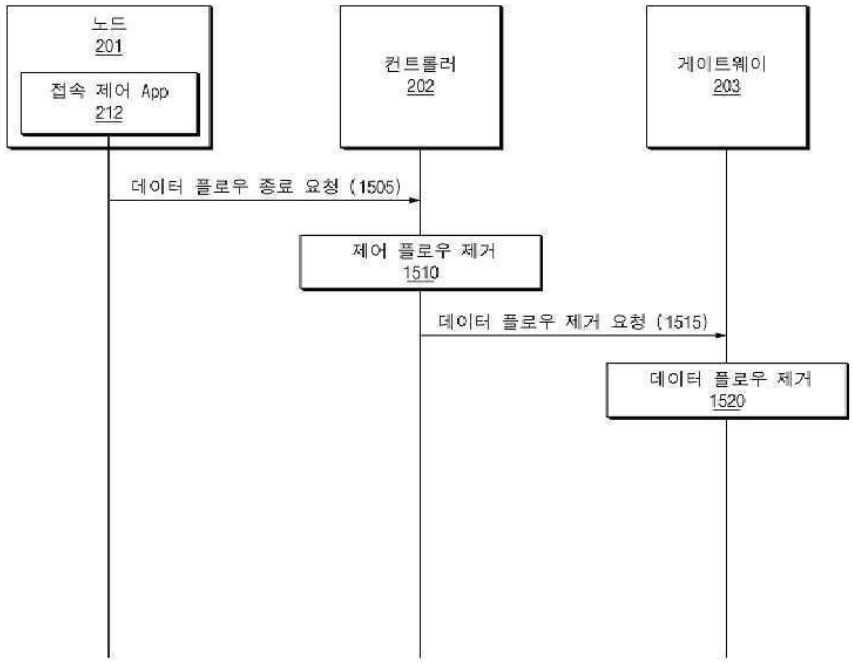
도면13


데이터 패킷 전송 이벤트 감지
1310
데이터 플로우 존재?
1320
아니오
예
파일 IO 테이블 확인
1330
데이터 패킷 드롭
1335
읽기 가능?
1340
예
아니오
파일 IO 미존재?
1345
예
아니오
데이터 패킷 검사
1350
데이터 패킷 검사 성공?
1360
예
아니오
데이터 패킷 드롭 및 네트워크 접속 차단
1370
데이터 패킷 전송
1355

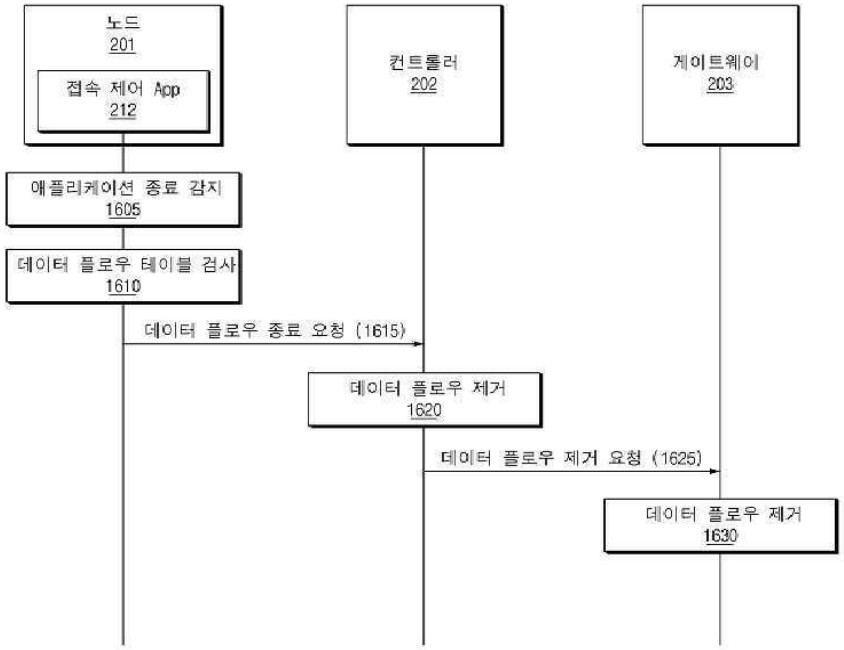
도면14


노드
201
접속 제어 App
212
컨트롤러
202
제어 플로우 갱신 요청 (1405)
제어 플로우 갱신
1410
갱신 결과 전송 (1415)

도면15


노드
201
접속 제어 App
212
컨트롤러
202
게이트웨이
203
데이터 플로우 종료 요청 (1505)
제어 플로우 제거
1510
데이터 플로우 제거 요청 (1515)
데이터 플로우 제거
1520

도면16


노드
201
접속 제어 App
212
컨트롤러
202
게이트웨이
203
애플리케이션 종료 감지
1605
데이터 플로우 테이블 검사
1610
데이터 플로우 종료 요청 (1615)
데이터 플로우 제거
1620
데이터 플로우 제거 요청 (1625)
데이터 플로우 제거
1630