(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2016-0133927
(43) 공개일자 2016년11월23일

(51) 국제특허분류(Int. Cl.)
G06F 21/56 (2013.01) *G06F 21/50* (2013.01)
(52) CPC특허분류
G06F 21/561 (2013.01)
G06F 21/50 (2013.01)
(21) 출원번호 10-2015-0067122
(22) 출원일자 2015년05월14일
심사청구일자 2015년05월14일
기술이전 희망 : **기술양도**, **실시권허여**, **기술지도**

전체 청구항 수 : 총 19 항

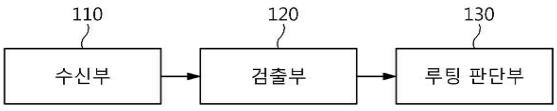
(71) 출원인
한국전자통신연구원
대전광역시 유성구 가정로 218 (가정동)
(72) 발명자
신진섭
대전광역시 유성구 유성대로 1559
정용익
대전광역시 유성구 유성대로 1559
(뒷면에 계속)
(74) 대리인
한양특허법인

(54) 발명의 명칭 **안드로이드 시스템 기반의 단말에서의 루팅 탐지 장치 및 방법**

(57) 요 약

루팅 탐지 장치 및 방법이 개시된다. 본 발명의 일실시예에 따른 루팅 탐지 장치는 안드로이드 운영 시스템 내부의 프로세스들/파일들을 수집하는 수집부; 상기 프로세스들/상기 파일들 내부의 명령어들에 기반하여 악성 프로세스 또는 악성 파일들을 검출하는 검출부; 및 검출된 악성 프로세스/상기 악성 파일에 기반하여 상기 안드로이드 운영 시스템의 루팅 여부를 판단하는 루팅 판단부를 포함한다.

대 표 도 - 도1



(72) 발명자

손종목

대전광역시 유성구 유성대로 1559

안재환

대전광역시 유성구 유성대로 1559

이동욱

대전광역시 유성구 유성대로 1559

백선엽

대전광역시 유성구 유성대로 1559

이준호

대전광역시 유성구 유성대로 1559

이한수

대전광역시 유성구 유성대로 1559

박용석

대전광역시 유성구 유성대로 1559

명 세 서

청구범위

청구항 1

안드로이드 운영 시스템 내부의 프로세스들/파일들을 수집하는 수집부;

상기 프로세스들/상기 파일들 내부의 명령어들에 기반하여 악성 프로세스 또는 악성 파일들을 검출하는 검출부; 및

검출된 악성 프로세스/상기 악성 파일에 기반하여 상기 안드로이드 운영 시스템의 루팅 여부를 판단하는 루팅 판단부

를 포함하는 것을 특징으로 하는 루팅 탐지 장치.

청구항 2

청구항 1에 있어서,

상기 검출부는

상기 프로세스들에 상응하는 실행 파일을 탐색하는 탐색부;

상기 실행 파일의 명령어 부분을 탐색하고, 상기 명령어 부분과 악성 파일들의 명령어 부분을 비교하는 비교부; 및

상기 명령어 부분에 상기 악성 파일들의 명령어 부분이 존재하는 경우, 상기 파일들이 악성 파일임을 판단하는 판단부

를 포함하는 것을 특징으로 하는 루팅 탐지 장치.

청구항 3

청구항 2에 있어서,

상기 루팅 판단부는

상기 탐색부에서 탐색한 결과, 상기 프로세스들에 상응하는 실행 파일이 존재하지 않는 경우, 상기 안드로이드 시스템이 루팅 되었음을 판단하는 것을 특징으로 하는 루팅 탐지 장치.

청구항 4

청구항 2에 있어서,

상기 루팅 판단부는

상기 탐색부에서 탐색한 결과, 상기 프로세스들에 상응하는 실행 파일이 읽기 권한이 없는 파일인 경우, 상기 안드로이드 시스템이 루팅 되었음을 판단하는 것을 특징으로 하는 루팅 탐지 장치.

청구항 5

청구항 1에 있어서,

상기 검출부는

상기 프로세스들을 부모 프로세스들과 자식 프로세스들로 분류하는 분류부; 및

상기 부모 프로세스들의 UID(User Identifier)와 상기 자식 프로세스들의 UID에 기반하여 상기 프로세스들이 악성 프로세스인지 판단하는 판단부

를 포함하는 것을 특징으로 하는 루팅 탐지 장치.

청구항 6

청구항 5에 있어서,

상기 판단부는

상기 부모 프로세스의 UID가 0이 아니고, 상기 자식 프로세스들의 UID가 0인 경우, 상기 자식 프로세스가 악성 프로세스인지 판단하는 것을 특징으로 하는 루팅 탐지 장치.

청구항 7

청구항 5에 있어서,

상기 판단부는

제조사 고유의 프로세스 리스트 또는 통신사 고유의 프로세스 리스트를 저장하는 데이터 베이스

를 더 포함하는 것을 특징으로 하는 루팅 탐지 장치.

청구항 8

청구항 7에 있어서,

상기 판단부는

상기 프로세스가 상기 데이터 베이스에 저장된 프로세스 리스트에 포함된 프로세스들인 경우, 상기 프로세스를 정상 프로세스로 판단하는 것을 특징으로 하는 루팅 탐지 장치.

청구항 9

청구항 1에 있어서,

상기 수집부는

상기 파일들 중 최종 수정 시간이 특정 시간 이내인 파일들을 수집하는 것을 특징으로 하는 루팅 탐지 장치.

청구항 10

청구항 9에 있어서,

상기 검출부는

상기 파일들의 명령어 부분을 탐색하는 탐색부;

상기 명령어 부분과 악성 파일들의 명령어 부분을 비교하는 비교부; 및

상기 명령어 부분에 상기 악성 파일들의 명령어 부분이 존재하는 경우, 상기 파일들이 악성 파일임을 판단하는 판단부

를 포함하는 것을 특징으로 하는 루팅 탐지 장치.

청구항 11

안드로이드 운영 시스템 내부의 프로세스들/파일들의 데이터들을 수집하는 단계;

상기 프로세스들/상기 파일들 내부의 명령어들에 기반하여 악성 프로세스 또는 악성 파일을 검출하는 단계; 및

검출된 악성 프로세스/악성 파일에 기반하여 상기 안드로이드 운영 시스템의 루팅 여부를 판단하는 단계

를 포함하는 것을 특징으로 하는 루팅 담지 방법.

청구항 12

청구항 11에 있어서,

상기 악성 프로세스 또는 악성 파일을 검출하는 단계는

상기 프로세스들에 상응하는 실행 파일을 탐색하는 단계;

상기 실행 파일의 명령어 부분을 탐색하고, 상기 명령어 부분과 악성 파일들의 명령어 부분을 비교하는 단계; 및

상기 명령어 부분에 상기 악성 파일들의 명령어 부분이 존재하는 경우, 상기 파일들이 악성 파일임을 판단하는 단계

를 포함하는 것을 특징으로 하는 루팅 탐지 방법.

청구항 13

청구항 12에 있어서,

상기 루팅 여부를 판단하는 단계는

상기 실행 파일을 탐색하는 단계에서, 상기 프로세스들에 상응하는 실행 파일이 존재하지 않는 경우, 상기 안드로이드 시스템이 루팅 되었음을 판단하는 것을 특징으로 하는 루팅 탐지 방법.

청구항 14

청구항 12에 있어서,

상기 루팅 여부를 판단하는 단계는

상기 실행 파일을 탐색하는 단계에서, 상기 프로세스들에 상응하는 실행 파일이 읽기 권한이 없는 파일인 경우, 상기 안드로이드 시스템이 루팅 되었음을 판단하는 것을 특징으로 하는 루팅 탐지 방법.

청구항 15

청구항 11에 있어서,

상기 악성 프로세스 또는 악성 파일을 검출하는 단계는

상기 프로세스들을 부모 프로세스과 자식 프로세스로 분류하는 단계; 및

상기 부모 프로세스의 UID(User Identifier)와 상기 자식 프로세스의 UID에 기반하여 상기 프로세스가 악성 프로세스인지 판단하는 단계

를 포함하는 것을 특징으로 하는 루팅 탐지 방법.

청구항 16

청구항 15에 있어서,

상기 악성 프로세스인지 판단하는 단계는

상기 부모 프로세스의 UID가 0이 아니고, 상기 자식 프로세스의 UID가 0으로 탐색된 경우, 상기 자식 프로세스가 악성 프로세스인지 판단하는 것을 특징으로 하는 루팅 탐지 방법.

청구항 17

청구항 15에 있어서,

상기 악성 프로세스인지 판단하는 단계는

상기 프로세스가 제조사 고유의 프로세스 리스트 또는 통신사 고유의 프로세스 리스트들이 저장된 데이터 베이스에 포함된 프로세스인 경우, 상기 프로세스를 정상 프로세스로 판단하는 것을 특징으로 하는 루팅 탐지 방법.

청구항 18

청구항 11에 있어서,

상기 데이터들을 수집하는 단계는

상기 파일들 중 최종 수정 시간이 특정 시간 이내인 파일들을 수집하는 것을 특징으로 하는 루팅 탐지 방법.

청구항 19

청구항 18에 있어서,

상기 악성 프로세스 또는 악성 파일을 검출하는 단계는

상기 파일들의 명령어 부분을 탐색하는 단계;

상기 명령어 부분과 악성 파일들의 명령어 부분을 비교하는 단계; 및

상기 명령어 부분에 상기 악성 파일들의 명령어 부분이 존재하는 경우, 상기 파일들이 악성 파일임을 판단하는 단계

를 포함하는 것을 특징으로 하는 루팅 탐지 방법.

발명의 설명

기 술 분 야

[0001] 본 발명은 안드로이드 시스템 기반의 단말에서 루팅을 탐지하는 방법에 관한 것으로, 특히 안드로이드 시스템 기반에서 수행되는 프로세스 또는 파일을 수집하여 악성 프로세스 또는 악성 파일을 검출하여 루팅 여부를 판단하는 기술에 관한 것이다.

배 경 기 술

[0002] 스마트 기기의 보급과 함께 은행업무가 스마트 기기에서 이루어지고, 회사 업무를 수행하는 모바일 오피스 환경이 구축되어 이를 이용한 중요한 처리가 스마트 기기에서 이루어지고 있다. 이로 인해 기존 PC에 한정된 악의적인 공격이 스마트폰까지 확대 되고 있고, 이러한 행위를 탐지하고 방어하는 연구가 필요해지고 이를 위한 환경 구축이 중요시 되고 있다. 이에 따라 안드로이드는 SELinux(Security Enhancements Linux)를 기반으로 한 SEAndroid를 안드로이드에 적용을 하는 등 보안 강화를 위한 노력을 하고 있다. 또한 개인정보 수집 및 여러 악성행위를 위해서 공격자는 안드로이드 시스템을 대상으로 최고관리자 권한을 가질 수 있는 루팅(Rooting)과정을 거치는데, 이를 탐지하는 방법에 대한 연구가 진행되고 있다.

[0003] 종래의 루팅 탐지 방식은 크게 2가지로 나뉜다. 안드로이드 시스템 변경을 통해 커널 수준에서 삽입되는 루팅 탐지, 그리고 안드로이드 애플리케이션 레벨에서 수행하는 일반 사용자 권한의 루팅탐지로 나뉜다.

[0004] 안드로이드 커널 레벨에서 삽입되는 루팅 탐지의 경우 시스템콜 후킹, 네트워크 패킷 모니터링 등이 가능함에 따라 이를 이용한 고수준의 루팅 탐지 방안의 도출이 가능하다.

[0005] 애플리케이션에서 수행하는 일반 사용자 권한의 루팅 탐지의 경우 시스템 자원의 접근 권한 제한으로 인해 상기의 루팅 탐지와 같은 고수준 루팅 탐지 방식의 도출의 어려움으로 이어진다.

[0006] 또한 종래에 존재하는 프로세스 리스트를 이용해 탐지를 할 경우 구글에서 제공하는 안드로이드 기본 프로세스 리스트 이외 모든 루트 프로세스를 검출하는데, 각 안드로이드 스마트폰 제조사마다 존재하는 루트 프로세스가 검출되어 오탐의 가능성이 존재한다.

[0007] 종종 풀링(Pooling)방식으로 애플리케이션에서 발생하는 이벤트를 모니터링하여 로그를 저장한다거나 하는 방식은 휴대기기 특성상 배터리 소모에 큰 영향을 줄 수 있으므로 상용화에 걸림돌이 된다.

[0008] 루팅탐지 방법의 가장 간단한 방법으로 루팅앱 설치시 함께 설치되는 'su' 파일의 검색이 있고, 이 'su' 파일의 검색을 하는 방법으로 흔히 시스템 명령어 디렉토리인 '/system/bin/' 및 '/system/xbin/' 등을 기준으로 찾는 구체적인 방법이 존재한다.

[0009] 또 이를 우회하는 방법에 대해 'su' 파일 내에 존재하는 특정 문자열을 기준으로 찾는 방법이 있다.

[0010] 또한 취약점을 이용하여 최고관리자 권한이 되었을 경우 일반사용자 권한에서 탐지하는 방법의 어려움이 있다.

[0011] 한국 공개 특허 제 2013-0060188호는 모바일 단말 상에서 개시되는 어플리케이션에 대한 프로세스 정보를 수신하여, 루팅 어플리케이션 정보와 프로세스 정보를 비교하여 루팅 어플리케이션 정보를 검출하는 기술에 대해서

개시하고 있고, 한국 등록 특허 제 1388053호는 안드로이드 운영체제의 관리자 권한 상승을 탐지하고, 악성 코드의 유무를 검출하는 기술에 대하여 개시하고 있다.

[0012] 하지만, 한국 공개 특허 제 2103-0060188호 및 한국 등록 특허 제 1388053호 역시 명령어 수준으로 루팅 탐지를 개시하지 못하고 있으며, 특히 단순히 권한 상승을 탐지하는 수준의 기술만을 개시할 뿐, 안드로이드 기반 시스템의 부모 프로세스와 자식 프로세스간의 관계를 고려한 루팅 탐지 기술에 대해서는 전혀 개시하지 못하고 있다.

[0013] 따라서, 최근의 안드로이드 기반 시스템을 탑재한 스마트폰의 폭발적인 보급 및 보안의 중요성이 강화되는 최근의 추세를 고려할 때, 좀 더 효과적으로 루팅을 탐지하고 안전한 실행 환경을 탐지하는 기술의 필요성이 대두되고 있다.

발명의 내용

해결하려는 과제

[0014] 본 발명의 목적은 안드로이드 시스템 내부의 파일 또는 프로세스를 명령어 수준에서 분석하여 루팅 탐지를 우회하는 여러 가지 방법들을 효과적으로 회피하는 것이다.

[0015] 또한, 본 발명의 목적은 프로세스뿐 만 아니라 최근 수정된 파일도 루팅을 탐지하는데 이용하는 것이다.

[0016] 또한, 본 발명의 목적은 안드로이드 시스템의 기본 프로세스뿐 만 아니라 제조사 또는 통신사의 기본 프로세스를 고려하여 루팅을 탐지하는 것이다.

과제의 해결 수단

[0017] 상기한 목적을 달성하기 위한 본 발명에 따른 루팅 탐지 장치는 안드로이드 운영 시스템 내부의 프로세스들/파일들을 수집하는 수집부; 상기 프로세스들/상기 파일들 내부의 명령어들에 기반하여 악성 프로세스 또는 악성 파일들을 검출하는 검출부; 및 검출된 악성 프로세스/상기 악성 파일에 기반하여 상기 안드로이드 운영 시스템의 루팅 여부를 판단하는 루팅 판단부를 포함한다.

[0018] 이 때, 상기 검출부는 상기 프로세스들에 상응하는 실행 파일을 탐색하는 탐색부; 상기 실행 파일의 명령어 부분을 탐색하고, 상기 명령어 부분과 악성 파일들의 명령어 부분을 비교하는 비교부; 및 상기 명령어 부분에 상기 악성 파일들의 명령어 부분이 존재하는 경우, 상기 파일들이 악성 파일임을 판단하는 판단부를 포함할 수 있다.

[0019] 이 때, 상기 루팅 판단부는 상기 탐색부에서 탐색한 결과, 상기 프로세스들에 상응하는 실행 파일이 존재하지 않는 경우, 상기 안드로이드 시스템이 루팅 되었음을 판단할 수 있다.

[0020] 이 때, 상기 루팅 판단부는 상기 탐색부에서 탐색한 결과, 상기 프로세스들에 상응하는 실행 파일이 읽기 권한이 없는 파일인 경우, 상기 안드로이드 시스템이 루팅 되었음을 판단할 수 있다.

[0021] 이 때, 상기 검출부는 상기 프로세스들을 부모 프로세스들과 자식 프로세스들로 분류하는 분류부; 및 상기 부모 프로세스들의 UID(User Identification)와 상기 자식 프로세스들의 UID에 기반하여 상기 프로세스들이 악성 프로세스인지 판단하는 판단부를 포함할 수 있다.

[0022] 이 때, 상기 판단부는 상기 부모 프로세스의 UID가 0이 아니고, 상기 자식 프로세스들의 UID가 0인 경우, 상기 자식 프로세스가 악성 프로세스인지 판단할 수 있다.

[0023] 이 때, 상기 판단부는 제조사 고유의 프로세스 리스트 또는 통신사 고유의 프로세스 리스트를 저장하는 데이터베이스를 더 포함할 수 있다.

[0024] 이 때, 상기 판단부는 상기 프로세스가 상기 데이터 베이스에 저장된 프로세스 리스트에 포함된 프로세스들인 경우, 상기 프로세스를 정상 프로세스로 판단할 수 있다.

[0025] 이 때, 상기 수집부는 상기 파일들 중 최종 수정 시간이 특정 시간 이내인 파일들을 수집할 수 있다.

[0026] 이 때, 상기 검출부는 상기 파일들의 명령어 부분을 탐색하는 탐색부; 상기 명령어 부분과 악성 파일들의 명령어 부분을 비교하는 비교부; 및 상기 명령어 부분에 상기 악성 파일들의 명령어 부분이 존재하는 경우, 상기 파일들이 악성 파일임을 판단하는 판단부를 포함할 수 있다.

[0027] 또한, 본 발명의 일실시예에 따른 루팅 탐지 방법은 안드로이드 운영 시스템 내부의 프로세스들/파일들의 데이터들을 수집하는 단계; 상기 프로세스들/상기 파일들 내부의 명령어들에 기반하여 악성 프로세스 또는 악성 파일을 검출하는 단계; 및 검출된 악성 프로세스/악성 파일에 기반하여 상기 안드로이드 운영 시스템의 루팅 여부를 판단하는 단계를 포함한다.

[0028] 이 때, 상기 악성 프로세스 또는 악성 파일을 검출하는 단계는 상기 프로세스들에 상응하는 실행 파일을 탐색하는 단계; 상기 실행 파일의 명령어 부분을 탐색하고, 상기 명령어 부분과 악성 파일들의 명령어 부분을 비교하는 단계; 및 상기 명령어 부분에 상기 악성 파일들의 명령어 부분이 존재하는 경우, 상기 파일들이 악성 파일임을 판단하는 단계를 포함할 수 있다.

[0029] 이 때, 상기 루팅 여부를 판단하는 단계는 상기 실행 파일을 탐색하는 단계에서, 상기 프로세스들에 상응하는 실행 파일이 존재하지 않는 경우, 상기 안드로이드 시스템이 루팅 되었음을 판단할 수 있다.

[0030] 이 때, 상기 루팅 여부를 판단하는 단계는 상기 실행 파일을 탐색하는 단계에서, 상기 프로세스들에 상응하는 실행 파일이 읽기 권한이 없는 파일인 경우, 상기 안드로이드 시스템이 루팅 되었음을 판단할 수 있다.

[0031] 이 때, 상기 악성 프로세스 또는 악성 파일을 검출하는 단계는 상기 프로세스들을 부모 프로세스과 자식 프로세스로 분류하는 단계; 및 상기 부모 프로세스의 UID(User Identifier)와 상기 자식 프로세스의 UID에 기반하여 상기 프로세스가 악성 프로세스인지 판단하는 단계를 포함할 수 있다.

[0032] 이 때, 상기 악성 프로세스인지 판단하는 단계는 상기 부모 프로세스의 UID가 0이 아니고, 상기 자식 프로세스의 UID가 0으로 탐색된 경우, 상기 자식 프로세스가 악성 프로세스인지 판단할 수 있다.

[0033] 이 때, 상기 악성 프로세스인지 판단하는 단계는 상기 프로세스가 제조사 고유의 프로세스 리스트 또는 통신사 고유의 프로세스 리스트들이 저장된 데이터 베이스에 포함된 프로세스인 경우, 상기 프로세스를 정상 프로세스로 판단할 수 있다.

[0034] 이 때, 상기 데이터들을 수집하는 단계는 상기 파일들 중 최종 수정 시간이 특정 시간 이내인 파일들을 수집할 수 있다.

[0035] 이 때, 상기 악성 프로세스 또는 악성 파일을 검출하는 단계는 상기 파일들의 명령어 부분을 탐색하는 단계; 상기 명령어 부분과 악성 파일들의 명령어 부분을 비교하는 단계; 및 상기 명령어 부분에 상기 악성 파일들의 명령어 부분이 존재하는 경우, 상기 파일들이 악성 파일임을 검출하는 단계를 포함할 수 있다.

발명의 효과

[0036] 본 발명은 안드로이드 시스템 내부의 파일 또는 프로세스를 명령어 수준에서 분석하여 루팅 탐지를 우회하는 여러 가지 방법들을 효과적으로 회피하여 정확한 루팅 탐지가 가능하다.

[0037] 또한, 본 발명은 프로세스뿐 만 아니라 최근 수정된 파일도 분석하여 정확한 루팅 탐지가 가능하다.

[0038] 또한, 본 발명은 안드로이드 시스템의 기본 프로세스뿐 만 아니라 제조사 또는 통신사의 기본 프로세스를 고려하여 루팅을 탐지하여, 루팅 탐지 오류를 획기적으로 감소시킬 수 있다.

도면의 간단한 설명

[0039] 도 1은 본 발명의 일실시예에 따른 루팅 탐지 장치를 나타낸 블록도이다.

도 2는 도 1에 도시된 검출부의 일실시예를 나타낸 블록도이다.

도 3은 본 발명의 일실시예에 따른 루팅 탐지 장치에서 악성 프로세스를 탐지하는 원리를 도시한 도면이다.

도 4는 본 발명의 일실시예에 따른 루팅 탐지 장치가 은행 업무 모바일 어플리케이션에 이용되는 것을 도시한 도면이다.

도 5는 본 발명의 일실시예에 따른 루팅 탐지 장치에서 파일의 최종 수정 시간에 따라서 수집하는 것을 도시한 도면이다.

도 6은 본 발명의 일실시예에 따른 루팅 탐지 방법을 나타낸 동작 흐름도이다.

도 7은 도 6에 도시된 악성 파일 또는 악성 프로세스 검출을 좀 더 자세히 나타낸 동작 흐름도이다.

발명을 실시하기 위한 구체적인 내용

[0040] 본 발명을 첨부된 도면을 참조하여 상세히 설명하면 다음과 같다. 여기서, 반복되는 설명, 본 발명의 요지를 불필요하게 흐릴 수 있는 공지 기능, 및 구성에 대한 상세한 설명은 생략한다. 본 발명의 실시형태는 당 업계에서 평균적인 지식을 가진 자에게 본 발명을 보다 완전하게 설명하기 위해서 제공되는 것이다. 따라서, 도면에서의 요소들의 형상 및 크기 등은 보다 명확한 설명을 위해 과장될 수 있다.

[0041] 이하, 본 발명에 따른 바람직한 실시예를 첨부된 도면을 참조하여 상세하게 설명한다.

[0042] 도 1은 본 발명의 일실시예에 따른 루팅 탐지 장치를 나타낸 블록도이다.

[0043] 도 1을 참조하면, 본 발명의 일실시예에 따른 루팅 탐지 장치는 수집부(110), 검출부(120) 및 루팅 판단부(130)로 구성되어 있다.

[0044] 수집부(110)는 안드로이드 파일 시스템 내부의 프로세스들 또는 파일들의 데이터들을 수집한다.

[0045] 이 때, 수집부(110)는 안드로이드 파일 시스템에서 수행되는 프로세스들 중에서 UID(User Identifier, 사용자 식별자)가 0인 프로세스들, 즉 루트 프로세스를 수집할 수 있다.

[0046] 이 때, 수집부(110)는 프로세스들을 수집할 때, 프로세스에 대한 정보를 가지고 있는 proc 디렉토리에서 모든 프로세스를 수집할 수 있다.

[0047] 이 때, 수집부(110)는 파일을 수집함에 있어, 파일들의 최종 수정 시간에 기반하여 파일을 수집할 수 있다. 예를 들어, 이전에 루팅 탐지를 수행한 시각 이후에 수정된 파일들을 수집부(110)에서 수집할 수도 있다. 이에 대한 설명은 도 5에서 서술한다.

[0048] 검출부(120)는 프로세스들 또는 파일들의 데이터에 기반하여 악성 프로세스 또는 악성 파일들을 검출한다.

[0049] 이 때, 검출부(120)는 수집부(110)에서 수집한 파일 내부의 명령어를 탐색할 수 있다.

[0050] 이 때, 파일 내부의 명령어를 탐색하는 것은 종래에는 문자열 또는 파일 이름으로 악성 프로세스를 검출하였으나, 본 발명은 여러 가지의 루팅 탐지 회피 방법에 대응하기 위해서 실행 코드 수준의 비교를 위해서 파일 내부의 명령어를 탐색한다.

[0051] 이 때, 검출부(120)는 파일 내부의 명령어와 악성 파일들의 명령어 부분을 비교할 수 있다.

[0052] 이 때, 악성 파일들의 명령어 부분은 미리 학습된 데이터 베이스에서 불러와서 비교할 수 있다.

[0053] 이 때, 악성 파일들의 명령어 부분은 본 발명인 루팅 탐지 장치를 업데이트하면서 데이터 베이스에 저장된 명령어 부분들의 수를 증가시킬 수도 있다.

[0054] 이 때, 검출부(120)는 파일 내부의 명령어에 악성 파일의 명령어 부분이 존재하는 경우, 상기 파일이 악성 파일임을 검출 할 수 있다.

[0055] 이 때, 검출부(120)는 수집부(110)에서 수집한 프로세스에 상응하는 실행 파일을 검색하고, 실행 파일의 명령어 부분을 찾아 악성 파일의 명령어 부분을 비교하여, 수집한 프로세스가 악성 프로세스인지 검출할 수도 있다. 이에 대한 설명은 도 2에서 서술한다.

[0056] 이 때, 검출부(120)는 프로세스들을 부모 프로세스와 자식 프로세스들로 분류하고, 부모 프로세스의 UID(User Identifier)와 자식 프로세스의 UID에 기반하여 프로세스가 악성 프로세스인지 검출할 수도 있다. 이는 도 3에서 서술한다.

[0057] 루팅 판단부(130)는 상기 악성 프로세스 또는 상기 악성 파일에 기반하여 상기 안드로이드 시스템의 루팅 여부를 판단한다.

[0058] 이 때, 루팅은 모바일 기기에서 구동되는 안드로이드 운영 체제 상에서 최상위 권한(루트 권한)을 얻음으로써, 해당 기기의 생산자 또는 판매자 측에서 걸어 놓은 제약을 해제하는 것을 의미한다.

[0059] 이 때, 모바일 기기에서 루팅을 함으로써, 상기 모바일 기기의 보안이 취약해지고, 루팅이 진행된 모바일 기기에서 해커들이 불법적인 행위를 할 수 있게 된다.

[0060] 이 때, 검출부(120)에서 악성 파일 또는 악성 프로세스를 검출한 경우, 모바일 기기에 설치된 안드로이드 시스템이 루팅 되었음을 탐지할 수 있다.

[0061] 이 때, 검출부(120)에서 프로세스에 상응하는 실행 파일이 존재하지 않는 경우, 모바일 기기에 설치된 안드로이드 시스템이 루팅 되었음을 탐지할 수 있다. 이는 악성 프로세스의 경우 실행 중 프로세스에 상응하는 실행 파일을 지울 수 있기 때문이다.

[0062] 이 때, 검출부(120)에서 프로세스에 상응하는 실행 파일이 읽기 권한이 없는 파일인 경우, 안드로이드 시스템이 루팅 되었음을 탐지할 수 있다. 안드로이드 시스템 내부의 폴더인 /system/bin 또는 /system/xbin에 존재하는 프로세스 또는 새로 추가된 실행 파일 중 읽기 권한이 없는 실행 파일들은 주로 악성 프로세스에 상응하는 실행 파일일 확률이 높을 수 있다.

[0063] 도 2는 도 1에 도시된 검출부의 일실시예를 나타낸 블록도이다.

[0064] 도 2를 참조하면, 검출부(120)는 탐색부(210), 비교부(220) 및 판단부(230)로 구성되어 있다.

[0065] 탐색부(210)는 프로세스들에 상응하는 실행 파일을 검색한다.

[0066] 비교부(220)는 실행 파일의 명령어 부분을 탐색하고, 실행 파일의 명령어 부분과 악성 파일들의 명령어 부분을 비교하고, 일치하는 구문이 존재하는지 확인한다.

[0067] 이 때, 악성 파일은 루팅된 안드로이드 시스템 내부에 존재하는 su(Superuser) 파일일 수도 있다.

[0068] 이 때, 악성 파일의 명령어 부분은 루팅된 안드로이드 시스템 내부에 존재하는 su 파일의 명령어 부분과 동일할 수 있다.

[0069] 이 때, 비교부(220)는 실행 파일의 명령어 부분을 탐색하면서, su 파일의 명령어 부분과 일치하는 부분이 존재하는지를 확인할 수 있고, 실행 파일의 명령어 부분을 탐색한 결과 su 파일의 명령어 부분과 일치하는 부분이 존재하는 경우, 실행 파일을 su 파일로 인식할 수 있다.

[0070] 판단부(230)는 실행 파일의 명령어 부분에 악성 파일들의 명령어 부분과 일치하는 부분이 존재하는 경우, 상기 프로세스가 악성 프로세스임을 판단한다.

[0071] 도 3은 본 발명의 일실시예에 따른 루팅 탐지 장치에서 악성 프로세스를 탐지하는 원리를 도시한 도면이다.

[0072] 도 3을 참조하면, 도 3은 정상적인 프로세스(310, 320, 350, 360), 악성 프로세스(330, 340, 370), 제조사 또는 통신사 고유의 프로세스(380)로 구성되어 있다.

[0073] 도 3을 다시 참조하면, 프로세스들은 부모 프로세스와 자식 프로세스로 분류되어 있는데, 이는 검출부(120)에서 수행될 수 있다.

[0074] 이 때, 자식 프로세스는 시스템 호출 등으로 의해 새롭게 형성된 프로세스를 의미하고, 부모 프로세스는 자식 프로세스를 발생시킨 프로세스를 의미할 수 있다.

[0075] 악성 프로세스(330, 340)는 미지의 루트 데몬이 프로세스 리스트 상에서 구동되어 있을 때, 수행되는 프로세스이다.

[0076] 이 때, 악성 프로세스(330, 340)는 특정 루팅 어플리케이션을 설치하면 구동되는 프로세스로, 일반적인 안드로이드 기본 프로세스 외 루트 프로세스일 수 있다.

[0077] 이 때, 검출부(120)는 악성 프로세스(330, 340)를 검출하고, 이를 루팅 판단부(130)에 전송할 수 있다.

[0078] 악성 프로세스(340)는 정상인 프로세스(350)의 자식 프로세스이나, UID가 0인 프로세스일 수 있다.

[0079] 이 때, 부모 프로세스의 UID가 0이 아니고, 자식 프로세스의 UID가 0인 경우는 안드로이드 기반 시스템에서는 정상적인 관계가 아니다. 부모 프로세스의 실행 중 취약한 부분이 존재하고, 취약한 부분에서의 악의적인 행위를 통해 권한 상승을 하여 자식 프로세스를 형성한 결과일 수 있다. 예를 들어, 악성 프로세스(340)은 정상인 프로세스(350)의 자식 프로세스이나, UID가 0으로 악성 프로세스이다.

[0080] 제조사 또는 통신사 고유의 프로세스(370)는 안드로이드 기반 시스템에서의 기본 프로세스 이외의 프로세스일 수 있다.

[0081] 이 때, 제조사 또는 통신사 고유의 프로세스(370)는 악성 프로세스가 아니므로, 이는 별도의 데이터 베이스에 저장된 프로세스 리스트에 포함되는 프로세스인지 판단하고, 제조사 또는 통신사 고유의 프로세스(370)이 별도의 데이터 베이스에 저장된 프로세스 리스트에 포함되는 프로세스인 경우, 악성 프로세스가 아닌 것으로 판단할 수도 있다.

[0082] 도 4는 본 발명의 일실시예에 따른 루팅 탐지 장치가 은행 업무 모바일 어플리케이션에 이용되는 것을 도시한 도면이다.

[0083] 도 4를 참조하면, 루팅이 수행된 모바일 기기에서 은행 업무 모바일 어플리케이션을 실행하였을 경우 나타나는 화면이다.

[0084] 이 때, 루팅 탐지 장치가 모바일 기기가 루팅 되었음을 판단할 수 있다.

[0085] 이 때, 루팅 탐지 장치가 모바일 기기가 루팅 되었다는 정보를 은행 업무 모바일 어플리케이션에 전송할 수 있다.

[0086] 이 때, 은행 업무 모바일 어플리케이션은 모바일 기기가 루팅 되었으므로 어플리케이션을 이용할 수 없음을 모바일 기기의 디스플레이상에 출력할 수 있다.

[0087] 도 5는 본 발명의 일실시예에 따른 루팅 탐지 장치에서 파일의 최종 수정 시간에 따라서 수집하는 것을 도시한 도면이다.

[0088] 도 5를 참조하면, 파일들(510 내지 590)의 목록이 도시되어 있다.

[0089] 수집부(110)는 안드로이드 파일 시스템 내부의 파일들(510 내지 590)을 모두 수집할 수도 있다.

[0090] 이 때, 수집부(110)는 안드로이드 파일 시스템 내부의 파일들(510 내지 590) 중에서 특정 시간 안에 유입된 파일 목록을 수집할 수도 있다.

[0091] 이 때, 수집부(110)는 파일을 수집함에 있어, 파일들의 최종 수정 시간에 기반하여 파일을 수집할 수 있다. 예를 들어, 이전에 루팅 탐지를 수행한 시각 이후에 수정된 파일들을 수집부(110)에서 수집할 수도 있다. 또 예를 들면, 루팅 탐지를 수행하고자 하는 시간에 대비하여 특정 시간 안에 유입된 파일들을 수집부(110)에서 수집할 수도 있다.

[0092] 도 5를 참조하여 예를 들면, 루팅 탐지를 수행하고자 하는 시간이 2014년 6월 28일이고, 2014년 6월 28일 이전 1달인 2014년 5월 28일부터 수정된 파일들(510, 520, 560, 570, 580, 590)을 수집부(110)에서 수집할 수 있다.

[0093] 이 때, 2014년 5월 28일 이전에 수정된 파일들(530, 540, 550)은 수집부(110)에서 수집하지 않을 수 있다.

[0094] 이 때, 특정 시간은 제한이 없다. 루팅 탐지 장치가 설치된 디바이스의 성능에 기반하여 특정 시간을 결정할 수 있다. 예를 들어 디바이스의 성능이 높은 경우, 특정 시간을 1년으로 잡고, 루팅 탐지를 수행하여 좀 더 정확한 루팅 탐지가 가능하게 할 수도 있다. 또 예를 들면, 디바이스의 성능이 다소 낮은 경우, 특정 시간을 2주로 잡고, 루팅 탐지를 수행하여 정확성이 낮지만 좀 더 빠른 루팅 탐지를 수행할 수도 있다.

[0095] 도 6은 본 발명의 일실시예에 따른 루팅 탐지 방법을 나타낸 동작 흐름도이다.

[0096] 도 6을 참조하면, 먼저 안드로이드 시스템 내부의 프로세스 또는 파일을 수집한다(S610).

[0097] 이 때, 안드로이드 파일 시스템에서 수행되는 프로세스들 중에서 UID(User Identifier, 사용자 식별자)가 0인 프로세스들, 즉 루트 프로세스를 수집할 수 있다.

[0098] 이 때, 프로세스에 대한 정보를 가지고 있는 proc 디렉토리에서 모든 프로세스를 수집할 수 있다.

[0099] 이 때, 파일들의 최종 수정 시간에 기반하여 파일을 수집할 수 있다. 예를 들어, 이전에 루팅 탐지를 수행한 시

각 이후에 수정된 파일들을 수집부(110)에서 수집할 수도 있다. 이에 대한 설명은 도 5에서 서술하였다.

[0100] 또한, 악성 프로세스, 악성 파일을 검출한다(S620).

[0101] 이 때, 수집부(110)에서 수집한 파일 내부의 명령어를 탐색할 수 있다.

[0102] 이 때, 파일 내부의 명령어를 탐색하는 것은 종래에는 문자열 또는 파일 이름으로 악성 프로세스를 검출하였으나, 본 발명은 여러 가지의 루팅 탐지 회피 방법에 대응하기 위해서 실행 코드 수준의 비교를 위해서 파일 내부의 명령어를 탐색한다.

[0103] 이 때, 파일 내부의 명령어와 악성 파일들의 명령어 부분을 비교할 수 있다.

[0104] 이 때, 악성 파일들의 명령어 부분은 미리 학습된 데이터 베이스에서 불러와서 비교할 수 있다.

[0105] 이 때, 악성 파일들의 명령어 부분은 본 발명인 루팅 탐지 장치를 업데이트하면서 데이터 베이스에 저장된 명령어 부분들의 수를 증가시킬 수도 있다.

[0106] 이 때, 파일 내부의 명령어에 악성 파일의 명령어 부분이 존재하는 경우, 상기 파일이 악성 파일임을 검출 할 수 있다.

[0107] 이 때, 수집부(110)에서 수집한 프로세스에 상응하는 실행 파일을 검색하고, 실행 파일의 명령어 부분을 찾아 악성 파일의 명령어 부분을 비교하여, 수집한 프로세스가 악성 프로세스인지 검출할 수도 있다. 이에 대한 설명은 도 2에서 서술한다.

[0108] 이 때, 프로세스들을 부모 프로세스와 자식 프로세스들로 분류하고, 부모 프로세스의 UID(User Identifier)와 자식 프로세스의 UID에 기반하여 프로세스가 악성 프로세스인지 검출할 수도 있다. 이는 도 3에서 서술하였다.

[0109] 또한, 안드로이드 시스템에 루팅 작업이 수행되었는지 여부를 판단한다(S630).

[0110] 이 때, 루팅은 모바일 기기에서 구동되는 안드로이드 운영 체제 상에서 최상위 권한(루트 권한)을 얻음으로써, 해당 기기의 생산자 또는 판매자 측에서 걸어 놓은 제약을 해제하는 것을 의미한다.

[0111] 이 때, 모바일 기기에서 루팅을 함으로써, 상기 모바일 기기의 보안이 취약해지고, 루팅이 진행된 모바일 기기에서 해커들이 불법적인 행위를 할 수 있게 된다.

[0112] 이 때, 악성 파일 또는 악성 프로세스를 검출한 경우, 모바일 기기에 설치된 안드로이드 시스템이 루팅 되었음을 탐지할 수 있다.

[0113] 이 때, 프로세스에 상응하는 실행 파일이 존재하지 않는 경우, 모바일 기기에 설치된 안드로이드 시스템이 루팅 되었음을 탐지할 수 있다. 이는 악성 프로세스의 경우 실행 중 프로세스에 상응하는 실행 파일을 지울 수 있기 때문이다.

[0114] 이 때, 프로세스에 상응하는 실행 파일이 읽기 권한이 없는 파일인 경우, 안드로이드 시스템이 루팅 되었음을 탐지할 수 있다. 안드로이드 시스템 내부의 폴더인 /system/bin 또는 /system/xbin에 존재하는 프로세스 또는 새로 추가된 실행 파일 중 읽기 권한이 없는 실행 파일들은 주로 악성 프로세스에 상응하는 실행 파일일 확률이 높을 수 있다.

[0115] 도 7은 도 6에 도시된 악성 파일 또는 악성 프로세스 검출을 좀 더 자세히 나타낸 동작 흐름도이다.

[0116] 도 7을 참조하면, 먼저 수집된 데이터들이 프로세스에 해당되는지 여부를 판단한다(S710).

[0117] 또한, 수집된 데이터들이 프로세스에 해당되는 경우, 프로세스가 실행되는 파일의 경로를 검색하고, 실행 파일을 수집한다(S720).

[0118] 또한, 수집된 데이터들이 파일에 해당되거나, S720에서 수집한 파일을 연다(S730).

[0119] 이 때, 파일이 성공적으로 열어졌는지 여부를 판단하고(S740), 성공적으로 파일이 열어진 경우, 파일 내의 명령어와 악성 파일 내의 명령어를 비교한다(S750).

[0120] 이 때, 악성 파일은 루팅된 안드로이드 시스템 내부에 존재하는 su(Superuser) 파일일 수도 있다.

[0121] 이 때, 악성 파일의 명령어 부분은 루팅된 안드로이드 시스템 내부에 존재하는 su 파일의 명령어 부분과 동일할

수 있다.

[0122] 또한, 파일 내의 명령어에 악성 파일 내의 명령어가 존재하는 경우, 악성 파일로 판단한다(S760, S790).

[0123] 이 때, 비교부(220)는 실행 파일의 명령어 부분을 탐색하면서, su 파일의 명령어 부분과 일치하는 부분이 존재하는지를 확인할 수 있고, 실행 파일의 명령어 부분을 탐색한 결과 su 파일의 명령어 부분과 일치하는 부분이 존재하는 경우, 실행 파일을 su 파일로 인식할 수 있다.

[0124] 또한, 파일이 열어지지 않았을 경우, 먼저 파일이 존재하는지 여부를 판단한다(S770).

[0125] 이 때, 파일이 존재하지 않는다면, 프로세스 또는 파일을 악성 프로세스 또는 악성 파일로 판정한다(S790).

[0126] 이 때, 프로세스에 상응하는 실행 파일이 존재하지 않는 경우, 모바일 기기에 설치된 안드로이드 시스템이 루팅되었음을 탐지할 수 있다. 이는 악성 프로세스의 경우 실행 중 프로세스에 상응하는 실행 파일을 지울 수 있기 때문이다.

[0127] 이 때, 파일이 존재하는 경우, 읽기 권한이 있는지 여부를 판단하고(S780), 읽기 권한이 없는 경우 악성 파일로 판정한다(S790).

[0128] 이 때, 프로세스에 상응하는 실행 파일이 읽기 권한이 없는 파일인 경우, 안드로이드 시스템이 루팅 되었음을 탐지할 수 있다. 안드로이드 시스템 내부의 폴더인 /system/bin 또는 /system/xbin에 존재하는 프로세스 또는 새로 추가된 실행 파일 중 읽기 권한이 없는 실행 파일들은 주로 악성 프로세스에 상응하는 실행 파일일 확률이 높을 수 있다.

[0129] 본 발명의 실시예는 컴퓨터로 읽을 수 있는 기록매체와 같은 컴퓨터 시스템에서 구현될 수 있다. 도 8에 도시된 바와 같이, 컴퓨터 시스템(820-1)은 버스(822)를 통하여 서로 통신하는 하나 이상의 프로세서(821), 메모리(823), 사용자 입력 장치(826), 사용자 출력 장치(827) 및 스토리지(828)를 포함할 수 있다. 또한, 컴퓨터 시스템(820-1)은 네트워크(830)에 연결되는 네트워크 인터페이스(829)를 더 포함할 수 있다. 프로세서(821)는 중앙 처리 장치 또는 메모리(823)나 스토리지(828)에 저장된 프로세싱 인스트럭션들을 실행하는 반도체 장치일 수 있다. 메모리(823) 및 스토리지(828)는 다양한 형태의 휘발성 또는 비휘발성 저장 매체일 수 있다. 예를 들어, 메모리는 ROM(824)이나 RAM(825)을 포함할 수 있다.

[0130] 따라서, 본 발명의 실시예는 컴퓨터로 구현된 방법이나 컴퓨터에서 실행 가능한 명령어들이 기록된 비일시적인 컴퓨터에서 읽을 수 있는 매체로 구현될 수 있다. 컴퓨터에서 읽을 수 있는 명령어들이 프로세서에 의해서 수행될 때, 컴퓨터에서 읽을 수 있는 명령어들은 본 발명의 적어도 한 가지 태양에 따른 방법을 수행할 수 있다.

[0131] 이상에서와 같이 본 발명에 따른 안드로이드 시스템 기반의 루팅 탐지 장치 및 방법은 상기한 바와 같이 설명된 실시예들의 구성과 방법이 한정되게 적용될 수 있는 것이 아니라, 상기 실시예들은 다양한 변형이 이루어질 수 있도록 각 실시예들의 전부 또는 일부가 선택적으로 조합되어 구성될 수도 있다.

부호의 설명

[0132] 310, 320, 350, 360: 정상적인 프로세스

330, 340, 370: 악성 프로세스

380: 제조사 또는 통신사 고유의 프로세스

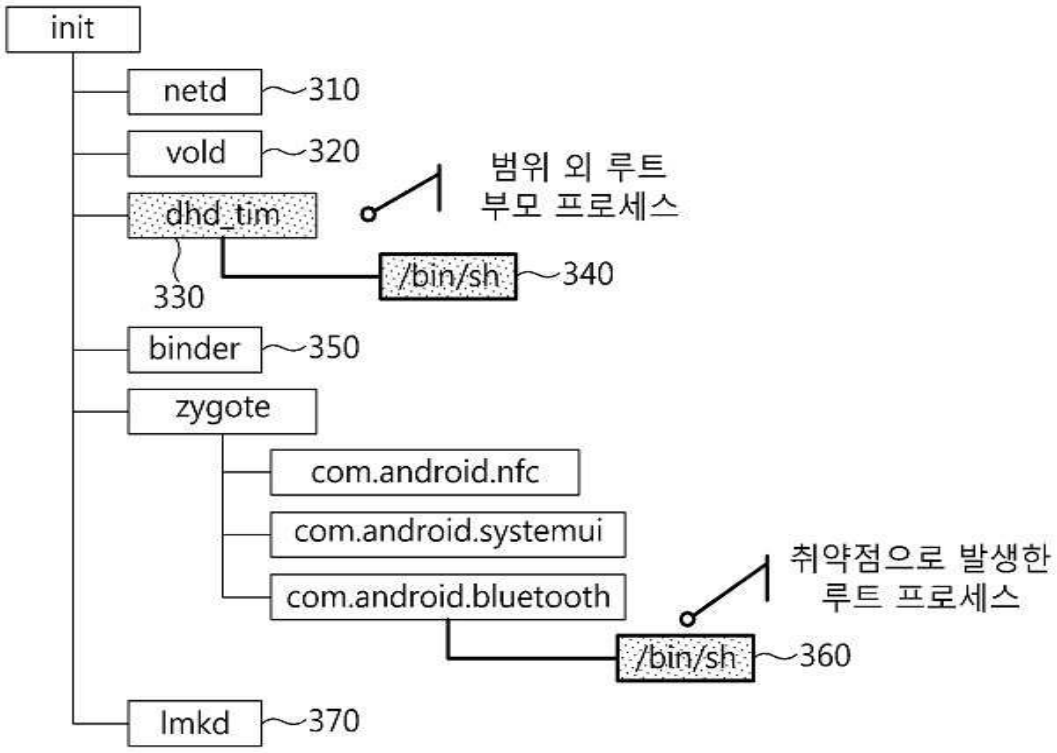
510, 520, 530, 540, 550, 560, 570, 580, 590: 파일 목록

도면

도면1

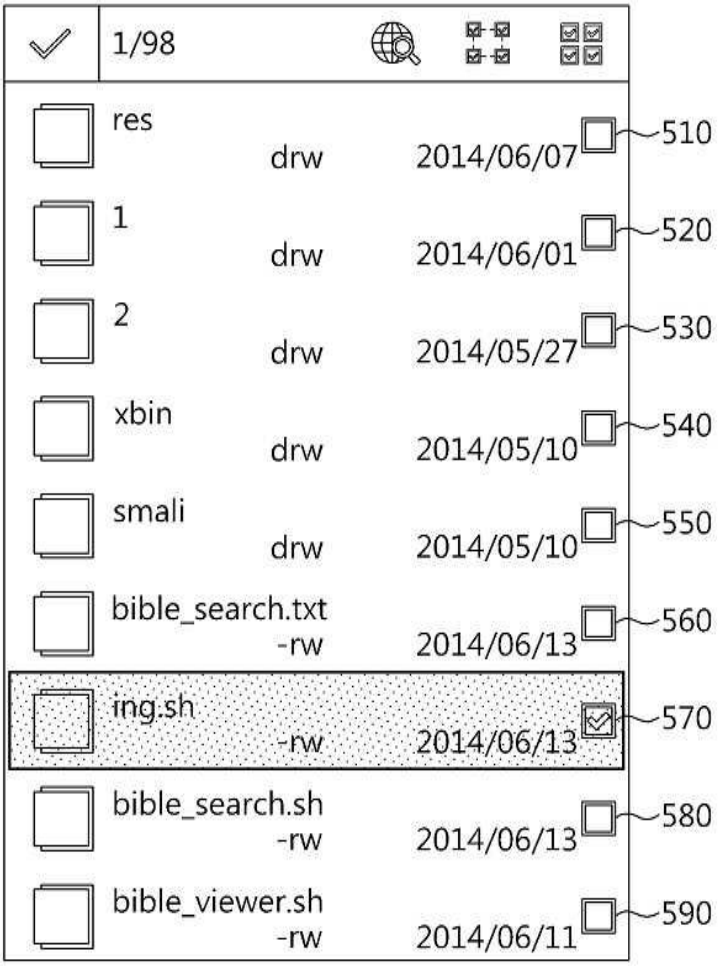
도면2

도면3

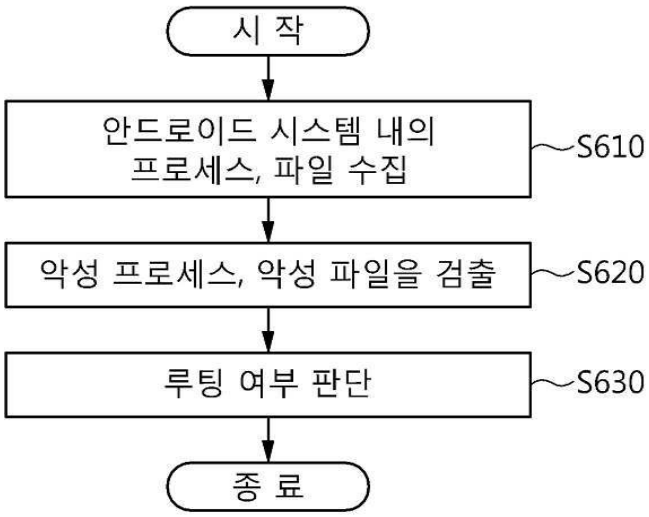


도면4

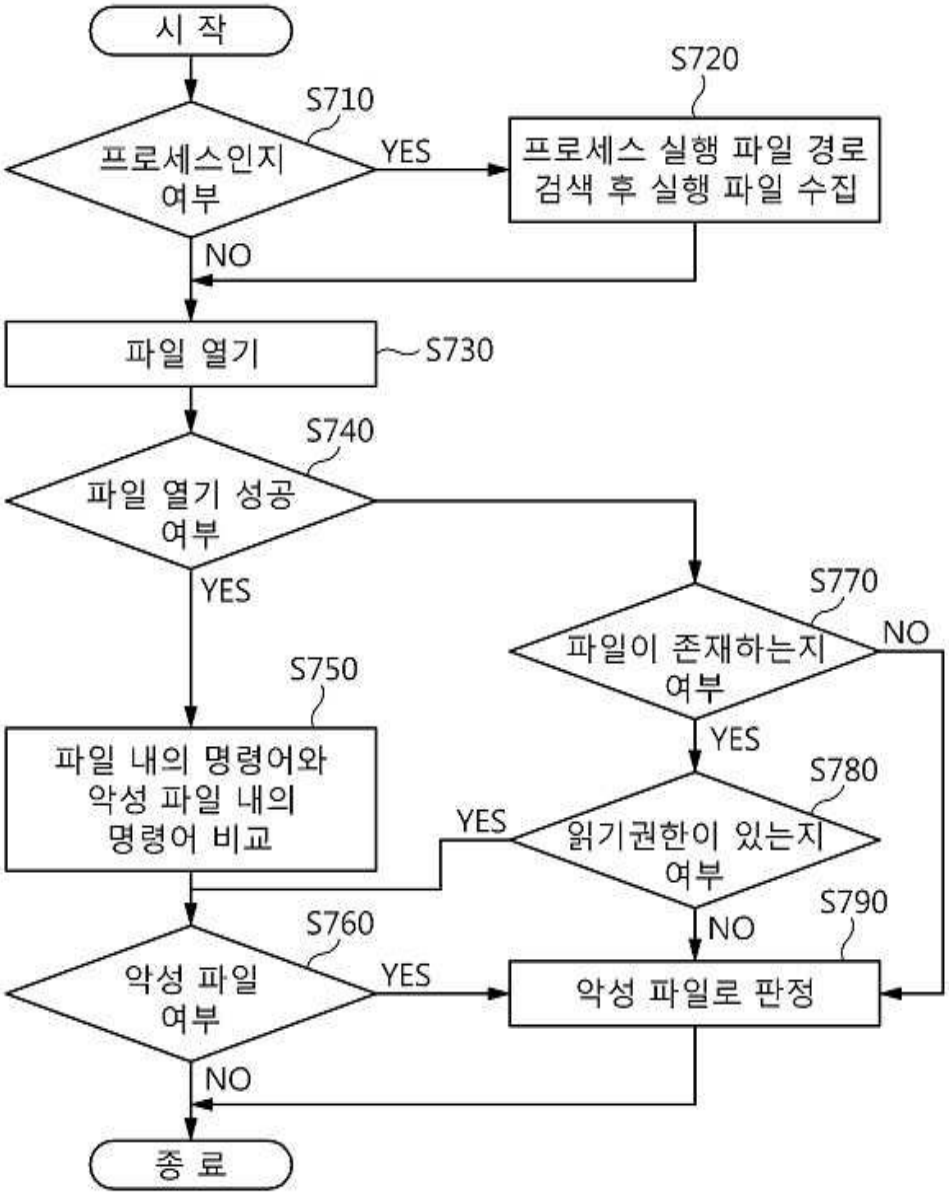
도면5


1/98
res
drw
2014/06/07
510
1
drw
2014/06/01
520
2
drw
2014/05/27
530
xbin
drw
2014/05/10
540
smali
drw
2014/05/10
550
bible_search.txt
-rw
2014/06/13
560
ing.sh
-rw
2014/06/13
570
bible_search.sh
-rw
2014/06/13
580
bible_viewer.sh
-rw
2014/06/11
590

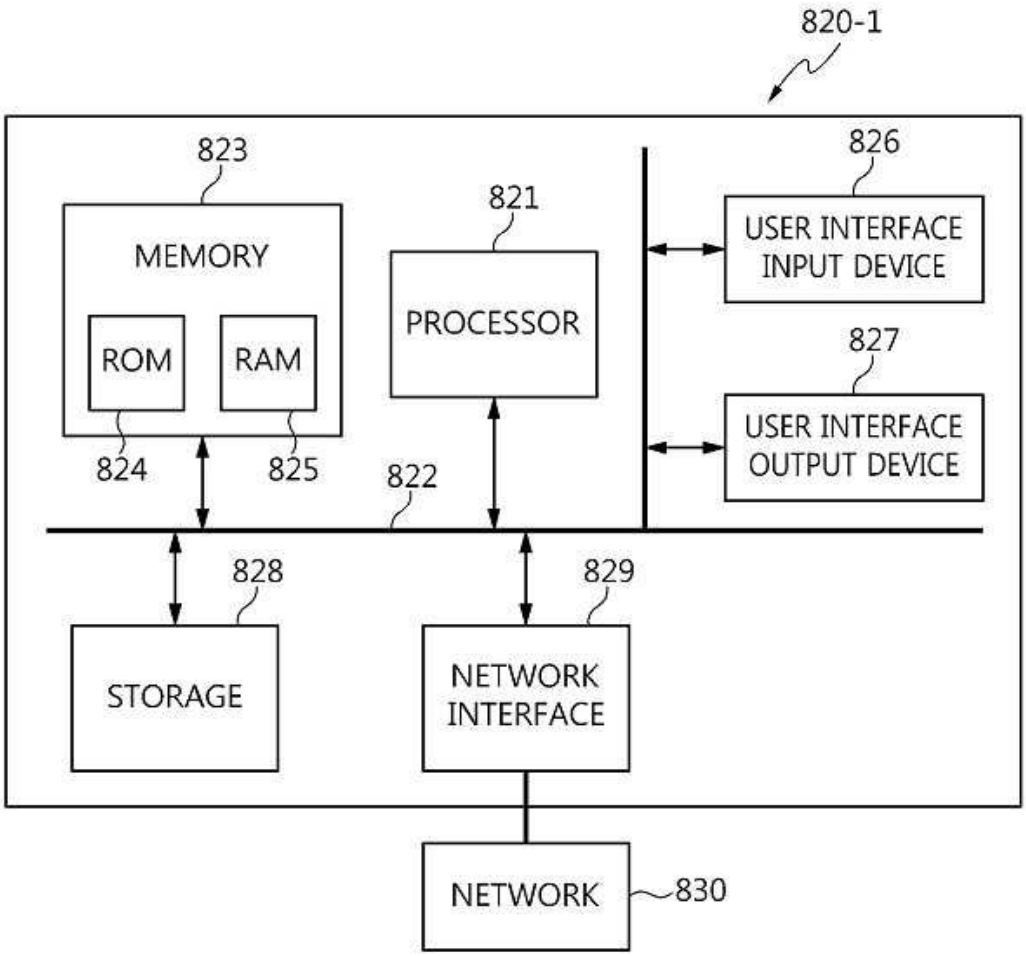
도면6


시 작
안드로이드 시스템 내의
프로세스, 파일 수집
S610
악성 프로세스, 악성 파일을 검출
S620
루팅 여부 판단
S630
종 료

도면7


시 작
S710
프로세스인지 여부
YES
S720
프로세스 실행 파일 경로 검색 후 실행 파일 수집
NO
파일 열기
S730
S740
파일 열기 성공 여부
YES
S770
파일이 존재하는지 여부
NO
S750
파일 내의 명령어와 악성 파일 내의 명령어 비교
YES
S780
읽기권한이 있는지 여부
YES
NO
S790
S760
악성 파일 여부
YES
악성 파일로 판정
NO
종 료

도면8


820-1
823
MEMORY
ROM
RAM
824
825
821
PROCESSOR
822
826
USER INTERFACE INPUT DEVICE
827
USER INTERFACE OUTPUT DEVICE
828
STORAGE
829
NETWORK INTERFACE
NETWORK
830