



(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2018-0060819
(43) 공개일자 2018년06월07일

(51) 국제특허분류(Int. Cl.)

G06F 21/56 (2013.01)

(52) CPC특허분류

G06F 21/566 (2013.01)

G06F 21/56 (2013.01)

(21) 출원번호 10-2016-0160766

(22) 출원일자 2016년11월29일

심사청구일자 없음

(71) 출원인

충남대학교산학협력단

대전광역시 유성구 대학로 99 (궁동, 충남대학교)

(72) 발명자

류재철

대전광역시 유성구 어은로 57, 132동 801호

윤정무

대전광역시 유성구 대학로145번길 22

조제경

대전광역시 유성구 문화원로10번길 38, 306호

(74) 대리인

홍성욱, 심경식

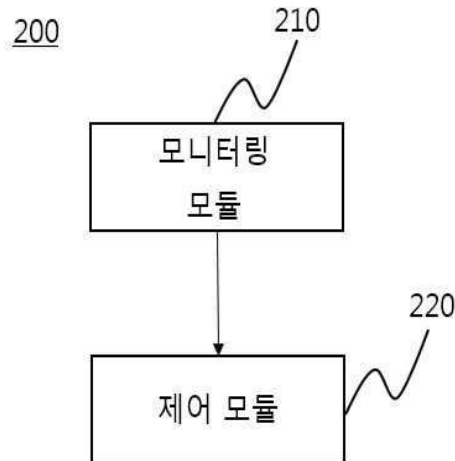
전체 청구항 수 : 총 1 항

(54) 발명의 명칭 랜섬웨어 공격 차단 장치 및 방법

(57) 요약

파일 입출력(I/O) 이벤트를 실시간으로 수집하고, 최근 기설정된 개수의 파일 입출력(I/O) 이벤트 중 소정 개수 이상의 파일 입출력(I/O) 이벤트가 동일 프로세스인지 판단한 뒤, 소정 개수 이상의 파일 입출력(I/O) 이벤트가 기설정된 시간 간격보다 짧은 시간 동안 발생하였는지 여부를 판단하는 모니터링 모듈 및 판단 결과에 기초하여, 파일 입출력(I/O) 요청을 전송한 프로그램에 대한 차단 여부를 결정하는 제어 모듈을 포함하는, 커널(Kernal) 영역에서 파일 입출력(I/O) 간격에 기초하여 랜섬웨어의 공격을 차단하는 장치 및 방법에 관한 것이다.

대표도 - 도2



이 발명을 지원한 국가연구개발사업

과제고유번호 1711035534

부처명 미래창조과학부

연구관리전문기관 정보통신기술진흥센터

연구사업명 정보통신기술인력양성

연구과제명 2016년 ICT/SW창의연구과정_충남대

기 여 율 1/1

주관기관 충남대학교 산학협력단

연구기간 2016.03.01 ~ 2017.02.28

명세서

청구범위

청구항 1

커널(Kernal) 영역에서 파일 입출력(I/O) 간격에 기초하여 랜섬웨어의 공격을 차단하는 장치에 있어서,

상기 파일 입출력(I/O) 이벤트를 실시간으로 수집하고, 최근 기설정된 개수의 상기 파일 입출력(I/O) 이벤트 중 소정 개수 이상의 상기 파일 입출력(I/O) 이벤트가 동일 프로세스인지 판단한 뒤, 상기 소정 개수 이상의 상기 파일 입출력(I/O) 이벤트가 기설정된 시간 간격보다 짧은 시간 동안 발생하였는지 여부를 판단하는 모니터링 모듈; 및

상기 판단 결과에 기초하여, 상기 파일 입출력(I/O) 요청을 전송한 프로그램에 대한 차단 여부를 결정하는 제어 모듈을 포함하는, 랜섬웨어 공격 차단 장치.

발명의 설명

기술 분야

[0001] 본 발명은 랜섬웨어의 공격을 차단하기 위한 랜섬웨어 공격 차단 장치 및 방법에 관한 것이다.

배경 기술

[0002] 현대사회는 컴퓨터의 발달로 무수히 많은 정보를 사람들의 편의에 맞게 제공한다. 그러나 컴퓨터가 가지는 정보의 중요도가 높아지고 양이 많아짐에 따라서 개인정보 침해에 대한 위험성 또한 증가하고 있다. 가장 대표적인 보안의 위험 요소로는 악성코드(malware)가 있으며 악성코드는 악의적인 목적을 위해 작성된 실행 가능한 코드의 통칭이다.

[0003] 악성코드는 자기 복제 능력과 감염 대상 유무에 따라 바이러스, 웜, 트로이목마, 랜섬웨어 등으로 분류된다. 그 중에서도 랜섬웨어란 인터넷 사용자의 컴퓨터에 잠입해 내부 문서나 스프레드시트, 그림파일 등을 암호화하여 열지 못하도록 만든 후 돈을 보내주면 해독용 열쇠 프로그램을 전송해 준다는 금품을 요구하는 악성 프로그램이다.

[0004] 랜섬웨어는 1999년 조셉 팝(Joseph Popp)에 의해 처음 만들어 졌으며 2013년에는 비트코인이라는 디지털 통화를 사용하는 크립토락커(cryptolocker)가 출현 하였다. 비트코인이란 지폐나 동전과 달리 물리적인 형태가 없는 온라인 가상화폐이다. 차세대 사이버 보안이슈와 위협 및 대처방안에 따르면 비트코인은 거래의 익명성이 보장되고 별다른 규제가 적용되지 않아서 불법적인 거래에 활용 될 수 있다. 비트코인의 익명성을 살린 크립토락커는 주로 이메일 첨부부를 통하여 구 버전의 인터넷 익스플로러, 플래시의 취약점을 공격하며 컴퓨터를 감염시킨다.

[0005] 여기서, 랜섬웨어란 몸값을 뜻하는 ransom과 제품을 뜻하는 ware의 합성어로 사용자의 동의 없이 컴퓨터에 불법으로 설치되어 문서나 스프레드시트, 그림 파일등을 암호화하여 열지 못하도록 한 뒤, 암호화된 파일을 원상복구 시켜주는 조건으로 금품을 요구한다. 대표적인 랜섬웨어는 시놀락커(synolocker), 나부터(nsblocker), 크립토락커(cryptolocker), 크립토월(cryptowall), 테슬라크립토(teslacrypto), 랜섬32(ransom32)등이 있다.

[0006] 처음 존재가 알려진 랜섬웨어는 1989년 조셉 팝에 의해 만들어 진 AIDS 트로이목마이며, 소프트웨어의 일부분이 라이선스가 만료되었다고 거짓 알람창을 띄우며 파일들을 암호화 한다. 파일을 해독하려면 'PC cyborg corporation'에 189달러를 지불 할 것을 요구한다.

[0007] 2006년 중반에 발견된 gpcode, archiveus등의 랜섬웨어는 이전의 랜섬웨어보다 복호화가 더 힘들도록 RSA 암호화 알고리즘을 사용하고 확장된 키 길이를 사용했다. GpcodeAG는 660비트 RSA공개키를 이용해서 암호화했고 변종인 Gpcode.AK는 1024비트 RSA키를 이용해 암호화 하였다. 2014년에는 NAS를 대상으로 하는 랜섬웨어인 synolocker가 확산되었다.

[0008] 2013년 7월에는 OS X운영체제에 특화된 랜섬웨어가 발생했다. 이것은 사용자가 불법포르노동영상을 다운로드했기 때문에 사용자를 고발한다는 웹페이지 형식의 알람창을 띄웠다. 윈도우 기반의 유사 제품과 다르게 전체 컴

퓨터 성능을 저해하지 않았고 웹 브라우저의 동작을 이용했다.

[0009] 운영체제 및 웹브라우저에 대한 통계를 제공하는 statcounter에 따르면 2016년 2월 컴퓨터 운영체제 점유율은 1위 Win7(46.16%) 2위 Win10(14.76%), 3위 Win8.1(11.41%), 4위 OS X(9.30%), 5위 WinXP(7.62%), 6위 Win8(3%)로 마이크로소프트사에서 만든 운영체제가 전체의 약 83%점유율을 차지하고 있으며 애플사의 운영체제는 전체의 약 9%점유율을 차지하고 있다. 얼마 전까지만 해도 랜섬웨어는 마이크로소프트사의 운영체제를 사용하는 x86 컴퓨터를 대상으로 감염시켰으나 2015년 자바스크립트 기술의 일종인 NW.js로 만들어진 Ransom32라는 랜섬웨어의 등장으로 리눅스나 OS X운영체제를 사용하는 컴퓨터도 더 이상 안전하지 않게 되었다.

[0010] Ransom32의 감염경로는 스팸메일을 이용하여 자동으로 압축이 풀리는 RAR파일을 보내는 것이다, 스팸메일을 사용자가 열어보는 순간 자동으로 랜섬웨어가 실행되며 "jpe", "mp3", "mov", "docx", "csv", "xml", "dat", "pptx"등의 확장자를 가진 파일들을 128비트 AES방식으로 암호화하고 비트코인을 통해 몸값을 요구한다. 피해자가 일정 기간 동안 몸값을 지불하지 않는 경우 몸값을 더 올리거나 파일을 삭제한다.

[0011] 랜섬웨어에 감염된 경우 피해자는 2가지 선택권이 있다. 제시된 몸값을 지불하거나 파일 접근 권한을 상실 하는 것이다. 암호화된 파일을 복호화 키 없이 해독하는 것이 거의 불가능하므로 암호화된 파일이 꼭 필요한 피해자는 몸값을 지불 할 수밖에 없다. 글로벌 사이버위협연합(CTA)보고서에 따르면 랜섬웨어인 크립토월로 인한 피해액은 전 세계적으로 약 3700억원에 달하는 것으로 추산된다고 한다. 또한 S사가 2015년에 발생한 모바일 랜섬웨어를 분석한 결과 2014년 대비 6배 이상 증가한 약 956억원으로 집계됐다고 밝혔다.

선행기술문헌

특허문헌

[0012] (특허문헌 0001) 한국 등록특허공보 제10-1311702호(2013.09.17)

발명의 내용

해결하려는 과제

[0013] 본 발명의 목적은, 상기 문제점을 해결하기 위한 것으로 최근 기설정된 개수의 파일 입출력(I/O) 이벤트 중 소정 개수 이상의 파일 입출력(I/O) 이벤트가 동일 프로세스인지 판단한 뒤, 소정 개수 이상의 파일 입출력(I/O)가 기설정된 시간 간격보다 짧은 시간 동안 발생한 경우 이를 랜섬웨어로 판단하여 차단하기 위한 것이다.

[0014] 본 발명이 해결하고자 하는 과제는 이상에서 언급한 과제(들)로 제한되지 않으며, 언급되지 않은 또 다른 과제(들)은 아래의 기재로부터 당업자에게 명확하게 이해될 수 있을 것이다.

과제의 해결 수단

[0015] 상기한 목적을 달성하기 위하여, 본 발명의 실시예에 따른 랜섬웨어 공격 차단 장치는 커널(Kernal) 영역에서 파일 입출력(I/O) 간격에 기초하여 랜섬웨어의 공격을 차단하는 장치이며, 파일 입출력(I/O) 이벤트를 실시간으로 수집하고, 최근 기설정된 개수의 상기 파일 입출력(I/O) 이벤트 중 소정 개수 이상의 파일 입출력(I/O) 이벤트가 동일 프로세스인지 판단한 뒤, 소정 개수 이상의 파일 입출력(I/O)가 기설정된 시간 간격보다 짧은 시간 동안 발생하였는지 여부를 판단하는 모니터링 모듈 및 판단 결과에 기초하여, 파일 입출력(I/O) 요청을 전송한 프로그램에 대한 차단 여부를 결정하는 제어 모듈을 포함한다.

발명의 효과

[0016] 본 발명의 일 실시예에 따르면, 최근 기설정된 개수의 파일 입출력(I/O) 이벤트 중 소정 개수 이상의 파일 입출력(I/O) 이벤트가 동일 프로세스인지 판단한 뒤, 소정 개수 이상의 파일 입출력(I/O)가 기설정된 시간 간격보다 짧은 시간 동안 발생한 경우, 랜섬웨어로 판단할 수 있다.

도면의 간단한 설명

[0017] 도 1은 종래의 랜섬웨어 탐지 방법을 설명하기 위한 도면이다.

도 2는 본 발명의 실시예에 따른, 랜섬웨어 공격 차단 장치를 설명하기 위한 구성도이다.

도 3은 본 발명의 실시예에 따른, 랜섬웨어 공격 차단 장치 및 방법을 설명하기 위한 도면이다.

도 4는 커널에서 수집 및 제어가 가능한 파일 오퍼레이션을 설명하기 위한 도면이다.

도 5는 본 발명의 실시예에 따른, 랜섬웨어 공격 차단 장치 및 방법을 설명하기 위한 도면이다.

도 6은 본 발명의 실시예에 따른, 랜섬웨어 공격 차단 장치의 모니터링 모듈을 설명하기 위한 도면이다.

도 7은 본 발명의 실시예에 따른, 랜섬웨어 공격 차단 장치의 제어 모듈을 설명하기 위한 도면이다.

도 8은 본 발명의 실시예에 따른, 랜섬웨어 공격 차단 장치가 동작하는 커널 영역을 설명하기 위한 도면이다.

도 9는 본 발명의 실시예에 따른, 랜섬웨어 공격 차단 장치의 모니터링 모듈을 설명하기 위한 도면이다.

도 10은 본 발명의 실시예에 따른, 랜섬웨어 공격 차단 장치의 모니터링 모듈을 설명하기 위한 도면이다.

도 11은 본 발명의 실시예에 따른, 랜섬웨어 공격 차단 장치의 제어 모듈을 설명하기 위한 도면이다.

도 12는 본 발명의 실시예에 따른, 랜섬웨어 공격 차단 장치의 모니터링 모듈 및 제어 모듈의 관계를 설명하기 위한 도면이다.

도 13 및 도 14는 크립토락커 랜섬웨어가 파일을 암호화하는데 걸리는 시간을 측정한 실험예를 설명하기 위한 도면이다.

도 15는 본 발명의 실시예에 따른, 랜섬웨어 공격 차단 장치 및 방법을 설명하기 위한 도면이다.

도 16은 본 발명의 실시예에 따른, 랜섬웨어 공격 차단 장치 및 방법을 설명하기 위한 도면이다.

도 17 내지 도 20은 크립토락커 랜섬웨어가 파일을 암호화하는데 걸리는 시간을 측정한 실험예를 설명하기 위한 도면이다.

발명을 실시하기 위한 구체적인 내용

[0018] 이하, 본 발명이 속하는 기술분야에서 통상의 지식을 가진 자가 본 발명의 기술적 사상을 용이하게 실시할 수 있을 정도로 상세히 설명하기 위하여, 본 발명의 가장 바람직한 실시예를 첨부 도면을 참조하여 설명하기로 한다. 우선 각 도면의 구성요소들에 참조부호를 부가함에 있어서, 동일한 구성요소들에 대해서는 비록 다른 도면상에 표시되더라도 가능한 한 동일한 부호를 가지도록 하고 있음에 유의해야 한다. 또한, 본 발명을 설명함에 있어, 관련된 공지 구성 또는 기능에 대한 구체적인 설명이 본 발명의 요지를 흐릴 수 있다고 판단되는 경우에는 그 상세한 설명은 생략한다.

[0019] 이하, 본 발명의 실시예에 따른 랜섬웨어 공격 차단 장치 및 방법을 첨부된 도면을 참조하여 상세하게 설명하면 아래와 같다.

[0020] 도 1은 종래의 랜섬웨어 탐지 방법을 설명하기 위한 도면이다. 도 2는 본 발명의 실시예에 따른, 랜섬웨어 공격 차단 장치를 설명하기 위한 구성도이다. 도 3은 본 발명의 실시예에 따른, 랜섬웨어 공격 차단 장치 및 방법을 설명하기 위한 도면이다. 도 4는 커널에서 수집 및 제어가 가능한 파일 오퍼레이션을 설명하기 위한 도면이다. 도 5는 본 발명의 실시예에 따른, 랜섬웨어 공격 차단 장치 및 방법을 설명하기 위한 도면이다. 도 6은 본 발명의 실시예에 따른, 랜섬웨어 공격 차단 장치의 모니터링 모듈을 설명하기 위한 도면이다. 도 7은 본 발명의 실시예에 따른, 랜섬웨어 공격 차단 장치의 제어 모듈을 설명하기 위한 도면이다. 도 8은 본 발명의 실시예에 따른, 랜섬웨어 공격 차단 장치가 동작하는 커널 영역을 설명하기 위한 도면이다. 도 9는 본 발명의 실시예에 따른, 랜섬웨어 공격 차단 장치의 모니터링 모듈을 설명하기 위한 도면이다. 도 10은 본 발명의 실시예에 따른, 랜섬웨어 공격 차단 장치의 모니터링 모듈을 설명하기 위한 도면이다. 도 11은 본 발명의 실시예에 따른, 랜섬웨어 공격 차단 장치의 제어 모듈을 설명하기 위한 도면이다. 도 12는 본 발명의 실시예에 따른, 랜섬웨어 공격 차단 장치의 모니터링 모듈 및 제어 모듈의 관계를 설명하기 위한 도면이다. 도 13 및 도 14는 크립토락커 랜섬웨어가 파일을 암호화하는데 걸리는 시간을 측정한 실험예를 설명하기 위한 도면이다. 도 15는 본 발명의 실시예에 따른, 랜섬웨어 공격 차단 장치 및 방법을 설명하기 위한 도면이다. 도 16은 본 발명의 실시예에 따른, 랜섬웨어 공격 차단 장치 및 방법을 설명하기 위한 도면이다. 도 17 내지 도 20은 크립토락커 랜섬웨어가 파일을 암호화하는데 걸리는 시간을 측정한 실험예를 설명하기 위한 도면이다.

- [0021] 도 1에 도시된 바와 같이, 종래의 랜섬웨어 공격을 차단하는 방법은 크게 시그니처에 기반한 방법, 정기적인 백업을 하는 방법, 보안 폴더를 이용하는 방법이 있다.
- [0022] 시그니처에 기반한 방법은, 랜섬웨어의 특징을 이용하여 랜섬웨어를 탐지하며, 정기적인 백업을 하는 방법은 랜섬웨어 감염 시 백업한 파일로 복구하는 방법이며, 보안폴더를 이용하는 방법은 사용자도 편집이 불가능한 폴더를 생성하는 방법이다.
- [0023] 이러한 랜섬웨어는 보안이 취약한 사이트 및 이메일을 통하여 감염되며, 이메일, 인스턴트 메세지, 웹사이트 등에서 링크를 클릭하면 자동으로 설치를 하고 내부에 잠입한다.
- [0024] 랜섬웨어의 일종인 cryptolocker가 사용하는 알고리즘은 복합 암호화방식(hybrid encryption)으로, 파일암호화는 대칭키 암호화 방식인 AES(Advanced Encryption Standard)를 이용하고 암호화한 내용들을 복호화 하는데 쓰이는 개인키를 다시 RSA비대칭키 암호화 방식으로 암호화 한다.
- [0025] Crptolocker가 이런 방식을 채택한 이유는 폴더를 탐색한 후 파일을 암호화하는 과정에서 비대칭키 암호화 알고리즘(RSA)을 사용하기에는 속도가 너무 느리기 때문이다. 즉, AES암호화방식은 암호화 하는 파일의 수만큼 수행되고 RSA암호화는 한번만 수행된다. 이렇게 RSA로 암호화된 AES키와 피해자를 식별 할 수 있는 컴퓨터 정보 등과 함께 데이터베이스화 하여 관리한다. 일반적으로 256비트 AES, 2048비트 RSA암호화 알고리즘을 사용한다.
- [0026] AES알고리즘으로 파일을 암호화 하면 복호화 키가 생성되는데 이를 이용하면 암호화된 파일을 복호화 할 수 있다. 컴퓨터가 cryptolocker에 감염 된 후에 악성코드를 수집하고 역 공학 분석을 통해서 암호화 하는 과정에서의 개인키 값을 추출 할 수는 있다. 하지만 대칭키 자동 생성 알고리즘의 특성 상 암호화 할 당시와 동일한 키가 생성되지 않는 것이 문제다. 그러므로 파일을 복호화하기 위해서는 처음에 암호화 후 생성된 복호화 키가 반드시 필요하다. RSA암호화 방식 또한 역 공학 분석 또는 공격자서버로 전송되는 키를 수집하더라도, 공격자가 가지고 있는 개인키를 알아내지 못하면 암호화된 파일의 복호화는 불가능하다.
- [0027] 종래의 랜섬웨어 공격을 차단하는 방법 중 백신을 이용하는 방법의 경우, 컴퓨터가 랜섬웨어에 감염 된 경우, 한동안 CPU팬이 매우 빠르게 회전하고 메모리를 과하게 쓰거나 때때로 화면 왼쪽 상단에 이상한 글자가 한 글자씩 생겨난다. 랜섬웨어는 컴퓨터 파일들을 탐색하기 시작하며 성능에 따라 약 5분에서 길게는 1시간정도 암호화 작업을 한다. 이때까지는 컴퓨터가 조금 느려지는 것 외에는 사용자는 특별한 변화를 체감하지 못한다. 파일들의 암호화가 시작되면 시스템 자원을 최대한 끌어서 최대한 빠르게 많은 파일들을 암호화하기 때문에 컴퓨터속도가 매우 느려지는 것을 체감 할 수 있다. 파일의 암호화가 완료된 후 컴퓨터를 재부팅 하게 되면 악성코드가 랜섬웨어에 걸렸음을 나타내는 알림창을 띄운다. 그 후로 감염된 컴퓨터로 일반적인 작업은 불가능하다.
- [0028] 종래의 랜섬웨어 공격을 차단하는 방법 중 특정 확장자 복구를 통한 랜섬웨어 피해 최소화 방법은 대피소라는 기능을 통하여 파일이 암호화 된 경우 자동으로 백업 폴더에서 파일들을 복구하는 방법으로, 대피소는 지정된 확장자명을 가진 파일 전부를 특정 폴더에 백업을 시켜놓는 것이다.
- [0029] 특정 확장자 복구를 통한 랜섬웨어 피해 최소화 방법에서는 랜섬웨어가 파일을 암호화 할 때 파일의 offset에 접근하는데, 이러한 방법에서는 파일의 offset에 접근하는 프로세스를 감지하면, 해당 파일을 백업하며, 랜섬웨어의 파일 암호화를 탐지하면 "랜섬웨어 행위 탐지" 팝업창을 띄우고 악성 프로세스를 차단함과 동시에 암호화된 파일들을 대피소에 저장된 원래 파일로 복구한다.
- [0030] 종래의 랜섬웨어 공격을 차단하는 방법 중 미끼 파일을 이용한 랜섬웨어 사전 탐지 방법은 컴퓨터 하드디스크에서 각 드라이브별로 숨김 속성 값을 가지는 폴더를 생성하며, 폴더 내부에는 4종의 미끼파일인 document.doc, hancm.hwp, image.jpg, text.txt파일을 추가한다. 이러한 방법에서는 랜섬웨어가 미끼 파일을 암호화하면, 이를 탐지하여 알림창을 사용자에게 보여준다.
- [0031] 즉, 종래의 랜섬웨어 공격을 차단하는 방법들은 랜섬웨어를 차단하기 보다는 랜섬웨어가 암호화한 파일을 원래의 파일로 복구를 하거나, 가짜 파일을 생성하여 랜섬웨어가 바꾸는 시점을 탐지하도록 되어 있다. 이는 랜섬웨어의 근본적인 해결책이기보다는 피해 최소화 방안이라고 할 수 있다. 특히 가짜 파일을 이용한 방법은 악성코드가 해당 파일을 암호화 하지 않는 방법으로 얼마든지 우회 할 수 있다.
- [0032] 도 2에 도시된 바와 같이, 본 발명의 실시예에 따른 랜섬웨어 공격 차단 장치(200)는 모니터링 모듈(210) 및 제어 모듈(220)을 포함한다.
- [0033] 예컨대, 본 발명의 실시예에 따른 랜섬웨어 공격 차단 장치(200)는 커널(Kernal) 영역에서 파일 입출력(I/O) 간

격에 기초하여 랜섬웨어의 공격을 차단하는 장치를 의미할 수 있다.

- [0034] 일 실시예에 따르면, 모니터링 모듈(210)은 파일 입출력(I/O) 이벤트를 실시간으로 수집하고, 최근 기설정된 개수의 파일 입출력(I/O) 이벤트 중 소정 개수 이상의 파일 입출력(I/O) 이벤트가 동일 프로세스인지 판단한 뒤, 소정 개수 이상의 파일 입출력(I/O) 이벤트가 기설정된 시간 간격보다 짧은 시간 동안 발생하였는지 여부를 판단할 수 있다.
- [0035] 예컨대, 모니터링 모듈(210)은 최근 10개의 파일 입출력 이벤트 중 5개 이상의 파일 입출력 이벤트가 동일 프로세스인지 판단한 뒤, 5개 이상의 파일 입출력 이벤트가 동일 프로세스인 경우, 5개 이상의 파일 입출력 이벤트가 326.9msec 이내에 발생한 경우, 이를 랜섬웨어에 의한 프로세스로 판단할 수 있다.
- [0036] 예컨대, 기설정된 시간 간격은 사람의 명령에 의해서 발생할 수 없는 시간 간격으로, 랜섬웨어를 비롯한 소프트웨어의 명령에 의해서만 발생할 수 있는 짧은 시간 간격을 의미할 수 있다.
- [0037] 예를 들어, 모니터링 모듈(210)은 윈도우 탐색기, 맥파인더를 비롯하여 OS가 기본적으로 제공하는 프로세스는 예외로 처리하여, 이러한 프로세스가 랜섬웨어로 차단되는 것을 방지할 수 있다.
- [0038] 제어 모듈(220)은 판단 결과에 기초하여, 파일 입출력(I/O) 요청을 전송한 프로그램에 대한 차단 여부를 결정한다.
- [0039] 다시 말해, 본 발명의 실시예에 따른, 랜섬웨어 공격 차단 장치 및 방법은 프로세스가 지나치게 빠르게 여러 파일들에 접근하고 파일을 수정할 경우 랜섬웨어 프로세스로 간주하고 해당 프로세스의 파일 접근 권한을 차단하는 장치 및 방법을 의미할 수 있다.
- [0040] 예컨대, 본 발명의 실시예에 따른, 랜섬웨어 공격 차단 장치 및 방법은 커널 기반의 파일 I/O 모니터링 및 제어 기술을 이용하여, 사용자가 수행할 가능성이 매우 낮은 파일 I/O 행위에 대하여 랜섬웨어로 판단하고 차단하는 장치 및 방법을 의미할 수 있다.
- [0041] 예를 들어, 본 발명의 실시예에 따른, 랜섬웨어 공격 차단 장치 및 방법은 운영체제의 파일 I/O를 제어하는 기술을 이용하여, 랜섬웨어를 포함한 프로세스의 파일 처리 요청을 모니터링하며, 이를 통하여 랜섬웨어 여부를 판단하고 제어할 수 있다.
- [0042] 예컨대, 본 발명의 실시예에 따른, 랜섬웨어 공격 차단 장치 및 방법은 커널영역에서 프로세스들의 파일 입출력 이벤트를 수집하고, 수집한 이벤트를 실시간으로 분석하여 지나치게 빠르게 여러 파일들을 열고 수정한다면 랜섬웨어 프로세스로 간주하고, 랜섬웨어 프로세스의 파일 입출력 요청을 커널에서 승인하지 않고 거부함으로써 랜섬웨어가 파일을 암호화하기 위해 파일을 열고자 하는 행위 자체를 막을 수 있다.
- [0043] 예컨대, 본 발명의 실시예에 따른, 랜섬웨어 공격 차단 장치 및 방법이 랜섬웨어 프로세스를 탐지하는 절차는 프로세스가 문서파일을 열기 위해 FILE OPEN을 커널에 요청하고 커널은 승인한 뒤, 커널에 요청한 프로세스의 FILE OPEN 요청 이벤트를 수집하고, 다시 프로세스는 필요한 FILE OPERATION을 요청한 뒤 커널은 승인하고, 커널은 FILE OPERATION이 FILE WRITE인 경우 요청 이벤트를 수집하며, 할 일을 마친 프로세스는 FILE CLOSE를 커널에 요청하고 커널은 승인하고, 커널에 요청한 프로세스의 FILE CLOSE 요청 이벤트를 수집한다.
- [0044] 상술한 동작을 반복하여, 프로세스가 4개의 파일을 열고 수정하고 닫은 후 5번째로 파일을 열고자 할 때, 모니터링 모듈(210)은 지금까지 소요된 시간을 산출하며, 소요된 시간이 기설정된 시간 이하인 경우 랜섬웨어 프로세스로 판단할 수 있다.
- [0045] 상술한 예시와 같은 상황에서 특정 프로세스가 랜섬웨어로 판단된 경우, 제어 모듈(220)은 랜섬웨어 프로세스의 5번째 파일에 대한 FILE OPEN 요청 이벤트를 거절하며, 커널에서 랜섬웨어 프로세스가 요청하는 모든 FILE OPERATION을 거절하며, 그 결과 랜섬웨어 프로세스는 파일에 대한 아무 OPERATION도 수행하지 못하게 된다.
- [0046] 예컨대, 본 발명의 실시예에 따른, 랜섬웨어 공격 차단 장치 및 방법은 커널영역에서 랜섬웨어 프로세스를 탐지하고 차단하기 때문에 다른 악성코드로부터 조작될 가능성이 낮으며, 랜섬웨어 프로세스로 판단되기 전까지인 최초 4개 파일은 암호화가 되더라도 랜섬웨어 프로세스로 판단이 되고 난 이후부터는 모든 파일 접근 요청을 거부함으로써 파일이 암호화되는 것을 방지할 수 있다.
- [0047] 이때, 도 3에 도시되 바와 같이, 본 발명의 실시예에 따른, 랜섬웨어 공격 차단 장치에 포함된 모니터링 모듈(210)은 커널 영역 중 탐지영역에서 이벤트를 모니터링 할 수 있으며, 모니터링 결과에 따른 명령을 제어 영역에 포함된 제어 모듈(220)에게 전송하여 특정 랜섬웨어 프로세스의 OPERATION 요청을 거부할 수 있다.

- [0048] 이때, 커널에서 수집 및 제어가 가능한 FILE OPERATION의 종류는 도 4에 도시된 바와 같다.
- [0049] 예컨대, 도 5에 도시된 바와 같이, 모니터링 모듈(210)은 파일의 Input/Output 이벤트를 통해 사용자에게 의한 프로세스인지 여부를 확인하고, 최근 기설정된 개수의 이벤트 중 소정 개수의 이벤트가 동일한 프로세스인 경우, 랜섬웨어에 의한 프로세스로 의심하며, 프로세스의 5개 이벤트에 대한 Input/Output 시간을 기준으로 랜섬웨어 프로세스를 판단하며, 제어 모듈(220)은 랜섬웨어 프로세스의 파일 접근 권한을 차단할 수 있다.
- [0050] 예컨대, 도 6에 도시된 바와 같이 모니터링 모듈(210, FILEOP)은 Kernel Authorization에 포함될 수 있으며, 모니터링 모듈(210)은 실제로 File Operation의 인가에는 관여하지 않고 이벤트만 수집할 수 있으며, 여기서 모니터링 모듈(210)이 수집하는 대상 이벤트는 KAUTH_FILEOP_OPEN, KAUTH_FILEOP_CLOSE, KAUTH_FILEOP_RENAME, KAUTH_FILEOP_EXCHANGE, KAUTH_FILEOP_LINK, KAUTH_FILEOP_EXEC 등이 포함될 수 있다.
- [0051] 예컨대, 도 7에 도시된 바와 같이, 제어 모듈(220, VNODE)은 Kernel Authorization에 포함될 수 있으며, 제어 모듈(220)은 File Operation에 대한 인가를 부여할 수 있다. 예컨대, 제어 모듈(220)은 KAUTH_VNODE_WRITE, KAUTH_VNODE_WRITE_DATA, KAUTH_VNODE_DELETE를 비롯한 총 18개의 이벤트의 인가에 관여하며, 그 리턴값으로 KAUTH_RESULT_DEFER, KAUTH_RESULT_ALLOW, KAUTH_RESULT_DENY를 제공할 수 있다.
- [0052] 다시 말해, 도 8에 도시된 바와 같이, 모니터링 모듈(210) 및 제어 모듈(220) 각각은 커널 영역에 포함되어 프로세스를 탐지하고, 프로세스를 인가하는데 관여할 수 있다.
- [0053] 도 9 및 도 10에 도시된 바와 같이, 모니터링 모듈(210)은 Kernel Authorization 기능 중 일부인 FIELOP를 이용하여 프로세스의 파일 접근 요청을 수집할 수 있으며, 최근 10개의 파일 접근 요청 중 5개 이상의 동일한 프로세스(A)가 발생한 경우, 최초로 발생한 프로세스 A의 접근시간인 a초와 마지막으로 발생한 프로세스 A의 접근 시간인 e초의 차이값인 e-a를 소요시간으로 산출하며, 소요시간이 기설정된 시간보다 짧은 경우 이를 랜섬웨어로 판단할 수 있다.
- [0054] 한편 도 11에 도시된 바와 같이, 제어 모듈(220)은 Kernel Authorization 기능 중 일부인 VNODE를 이용하여 랜섬웨어 프로세스의 파일접근 요청을 거부할 수 있으며, 도 9 내지 도 11에 도시된 전체 동작은 도 12에 도시된 바와 같다.
- [0055] 도 13 및 도 14에 도시된 바와 같이, 크립토락커의 파일 암호화 시간은 샘플 1의 경우 5개를 암호화하기 위한 최소시간은 1.75msec로 최대시간은 523.50msec로 나타났으며, 샘플 2의 경우 5개를 암호화하기 위한 최소시간은 1.75msec로 최대 시간은 428.50msec로 나타났다.
- [0056] 상술한 실험예의 결과를 통해, 5개 이상의 파일 입출력(I/O) 이벤트가 기설정된 시간 간격보다 짧은 시간 동안 발생하였는지 여부를 판단하기 위한 기설정된 시간은 크립토락커가 5개의 파일을 암호화하는데 걸리는 최소시간인 1.75msec일 수 있다.
- [0057] 예를 들어, 5개 이상의 파일 입출력(I/O) 이벤트가 기설정된 시간 간격보다 짧은 시간 동안 발생하였는지 여부를 판단하기 위한 기설정된 시간은 가장 빠른 APM을 보유한 프로게이머가 5번의 명령을 입력하는데 걸리는 시간인 326.9msec일 수 있다.
- [0058] 다시 말해, 랜섬웨어가 특정 프로세스가 파일 암호화를 시도 할 때, 프로세스는 운영체제의 커널에 파일 I/O를 요청하게 되며, 본 발명의 실시예에 따른, 랜섬웨어 공격 차단 장치(200)는 이러한 파일 I/O를 탐지하고, 탐지한 파일 I/O 정보를 바탕으로 이상 징후 탐지모델이 파일의 접근이 비정상적인지 판단하며, 파일에 대한 접근이 비정상적인 경우, 랜섬웨어로 판단하고 해당 프로세스의 파일접근권한을 차단하여 파일 암호화를 불가능하게 할 수 있다.
- [0059] 예컨대, 본 발명의 실시예에 따른 랜섬웨어 공격 차단 장치(200)의 커널을 이용한 파일 I/O 모니터링 및 제어 모듈은 도 15에 도시된 바와 같다.
- [0060] 도 15에 도시된 바와 같이 본 발명의 실시예에 따른 랜섬웨어 공격 차단 장치(200)는 프로세스의 행동을 탐지하고 제어하기 위해서 커널의 기능을 활용할 수 있다.
- [0061] 여기서, 랜섬웨어는 파일을 암호화하기 위해서 프로세스를 생성하여 파일을 탐색하고, 파일을 변경(암호화)하게 된다. 심지어 파일의 확장자명까지 변경하는 랜섬웨어도 있는 것으로 확인되었다. 이러한 파일의 변경 행위는 커널에서 탐지 및 제어가 가능하며 이는 마이크로소프트사의 윈도우즈 제품군에서는 드라이버, 애플사의 OSX 제품군에서는 커널 익스텐션(kernel extension)이 있으며, 본 발명의 실시예에 따른 랜섬웨어 공격 차단 장치

(200)가 동작하는 커널은 상술한 예시에 한정되지 않는다.

- [0062] 예컨대, 모니터링 모듈(210)은 커널영역에서 특정한 확장자명을 가진 파일에 대한 처리 요청을 탐지한다. 파일 처리를 요청한 프로세스가 파일내용을 변경하기 위해서는 반드시 파일을 열어야하며, 커널은 파일을 열었을 때 발생하는 신호를 커널 모듈에 전달함으로써 해당 프로세스가 파일에 접근한 시간 및 파일경로를 커널 모듈에서 확인 할 수 있다. 또한 파일 I/O 모니터링 모듈은 프로세스의 파일 접근에 대한 탐지만 아니라 프로세스의 파일 접근 제어가 가능하다. 특정한 프로세스 인식자(process identifier, 이하 pid)값을 통하여 각각의 프로세스를 식별 가능하며, 이를 이용하여 특정 프로세스의 파일접근을 제어 할 수 있다. 그리고 pid와 상관없이 특정한 파일을 기준으로 파일에 접근하려는 프로세스를 제어 할 수 있다.
- [0063] 파일 I/O 모니터링 모듈은 파일에 접근한 프로세스의 정보를 수집하고 제어하는 역할만 수행하며, 파일접근 자체가 합법적인지 불법적인지에 대한 판단은 하지 않는다. 파일 I/O 모니터링 모듈은 정보 송수신 모델로부터 받은 정보를 이용해 랜섬웨어의 프로세스로 인지하고 해당 프로세스의 파일접근권한을 차단한다.
- [0064] 예컨대, 커널에서 동작하는 파일 접근정보 송수신 모듈은 차단해야 할 프로세스에 대한 정보와 파일 처리 요청에 대한 정보를 파일 I/O 모니터링 및 제어모델과 공유한다. 이 정보는 소켓을 이용해 유저영역에서 동작하는 Detector와 주고받는다. 소켓으로 송수신하는 메시지의 내용은 파일 I/O 모니터링 및 제어모델이 특정한 프로세스가 파일에 접근할 경우 탐지했던 pid와 시간, 파일경로가 있다. 소켓은 해당 정보를 보관하고 있다가 Detector와의 연결이 확인되면 Detector에게 메시지를 송신한다.
- [0065] 반대로, 파일 접근정보 송수신 모듈은 Detector로부터 메시지를 수신 할 수도 있다. 최초 정보 송수신 모듈이 Detector에 데이터 메시지를 송신한다. Detector는 수신한 메시지를 열어서 특정 파일에 접근하려는 프로세스를 확인하고 접근이 정상적인지 비정상적인지를 판단한다. 프로세스가 비정상적으로 파일에 접근한다고 판단하면, Detector는 파일 접근 정보를 정보 송수신 모듈에게 송신한다. 정보 송수신 모듈은 수신한 메시지를 파일 I/O 모니터링 및 제어 모델과 공유한다. 결과적으로 파일 I/O 모니터링 및 제어 모델이 랜섬웨어가 사용하는 프로세스에 대한 파일접근을 차단함으로써 파일 암호화를 불가능하게 만든다.
- [0066] 본 발명의 실시예에 따른 랜섬웨어 공격 차단 장치(200)의 비정상적 파일 접근 탐지 모델은 도 16에 도시된 바와 같다.
- [0067] 도 16에 도시된 바와 같이, 특정 프로세스가 커널에게 파일 접근을 요청 할 경우 파일 I/O 모니터링 및 제어모델은 관련 정보를 담은 메시지를 Detector에게 송신한다. 구체적으로, 커널에서 작동하는 파일정보 송수신 모듈은 소켓을 가지고 있다. 이 소켓을 이용하여 사용자 레벨에서 작동하는 Detector의 파일정보 송수신 모듈이 가지고 있는 소켓에게 정보 메시지를 송신한다. 이 모듈은 지속적으로 파일 접근정보 메시지를 수신하고 메시지 내용을 이상징후 탐지모델과 공유한다. 리시브(recv)함수를 통해 수신한 이 정보 메시지는 정보를 문자열타입의 변수를 이용하여 보관한다. 메시지에 담을 정보의 최대 길이는 1024바이트로 설정될 수 있다.
- [0068] 정보 메시지는 pid, 파일접근시간, 파일경로가 들어있다. pid, 파일접근시간 및 파일경로는 가변적인 길이를 가진다. 하지만 pid 와 파일접근시간은 포맷이 정해져 있기 때문에 어느 정도 길이가 정해져 있다. 운영체제마다 파일 최대경로길이가 다르며 윈도우 운영체제 같은 경우에는 폴더 및 파일의 전체 경로길이를 담는 변수의 최대 길이는 260바이트이다. 따라서 일반적인 사용자는 하위폴더를 무수히 많이 만들어 사용한다 하더라도 파일경로의 길이까지 합친 사이즈의 최대값은 1024바이트로 설정할 수 있다.
- [0069] Detector의 정보 송수신 모듈이 수신한 메시지는 텍스트파일에 수신한 내용을 저장한다. 텍스트파일에 저장하지 않고 실시간으로 메시지를 확인해서 비정상적인 파일접근을 판단 할 수도 있지만 로그를 남겨서 파일 접근판단에 대한 근거자료를 가지고 있는 것이 추후 에러가 발생한 경우에 더 능동적으로 대처 할 수 있다.
- [0070] 커널의 정보송수신 모듈이 메시지를 송신하는 시점을 모르기 때문에 Detector의 정보 송수신 모듈은 계속 대기하면서 메시지를 수신 할 때마다 텍스트파일에 메시지를 추가한다.
- [0071] 예컨대, 이상징후 탐지 모델은 앞서 설명한 파일 접근 정보 수신 및 파일 접근 명령 송신 모듈에서 저장한 정보를 이용하여 판단한다. PID, 파일 접근 시간, 파일의 전체 경로에 대한 정보를 구조체 형태로 저장하고 있으며, 이를 이용할 경우 빠른 검색 및 비교가 가능하다.
- [0072] 하지만 파일에 대한 I/O 요청은 랜섬웨어의 프로세스뿐만 아니라 일반적인 프로세스에서도 요청하기 때문에 단순히 한 번의 파일 I/O 요청만으로는 비정상적인 접근을 판단 할 수 없다. 이상징후 탐지 모델은 가장 최근에 저장된 10개의 메시지를 확인한다. 10개의 메시지 중에서 동일한 프로세스가 5개 이상이면 동일한 프로세스의

메세지들을 가져온다. 가져온 메세지 중에서 가장 최근의 메세지와 가장 오래된 메세지의 시간차이가 0.01초 이하일 경우에 비정상적인 접근으로 판단한다. 비정상적인 파일접근으로 판단 할 경우, 정보 송수신 모델과 정보를 공유하고 이 정보를 담은 메세지를 커널의 정보 송수신 모델에 전송한다. 메세지를 수신하면 해당 정보를 파일 I/O 모니터링 및 제어 모델이 확인하고 해당 프로세스의 파일 접근권한을 차단한다.

[0073] 이상징후 탐지 모델은 pid를 기준으로 비정상적으로 파일을 접근한 프로세스의 명령어 또는 경로를 확인하고 pid와 관련 정보를 사용자에게 알려준다. 만약 비정상적인 파일접근을 하는 프로세스가 실제로 랜섬웨어의 프로세스라면 사용자에게 차단 여부를 묻는 동안에도 파일암호화가 진행되기 때문에 프로세스를 우선 차단 한 후에 사용자에게 차단여부를 묻는다. 이상징후 모델은 비정상적으로 파일에 접근하는 프로세스를 사용자에게 알리고 차단여부를 묻는 역할을 한다. 사용자가 차단을 원하지 않은 경우 이상징후 탐지모델은 정보 송수신 모델에게 차단초기화 메세지의 송신을 요청한다. 이 메세지를 커널영역의 정보 송수신 모델이 수신하면 파일 I/O 모니터링 탐지 및 제어 모델과 메세지 내용을 공유하며, 파일 I/O 모니터링 탐지 및 제어 모델은 차단하고 있던 프로세스의 파일접근 권한을 원래 상태로 되돌려 놓는다.

[0074] 도 17에 도시된 바와 같이 크립토락커 샘플1을 활용한 경우, 가상환경의 운영체제에서 파일 암호화 진행시 랜섬웨어의 프로세스는 총 3600번 파일에 접근했다. 한 번의 접근을 한 번의 파일 암호화라고 본다면, 200개의 파일을 암호화 하는데 소모되는 시간은 작게는 0.07초에서 길게는 20.94초 소모된다. 이때, 본 발명의 실시예에 따른, 랜섬웨어 공격 차단 장치(200)는 가장 최근에 발생한 10개의 파일접근 이벤트에서 한 프로세스가 5개 이상의 파일에 접근 한 시간차이를 이용하므로 매 실험 단위마다 5개의 파일을 암호화하는데 걸리는 평균 시간을 측정했다. 측정결과, 가장 작게는 0.00175초가 걸렸고 가장 길게는 0.7075초가 걸렸다. 위의 실험을 기준으로 5개의 파일 접근에 대하여 적어도 0.00175초 차이가 날 경우를 탐지해야만 암호화 할 대상 파일 모두에 대한 탐지가 가능함을 확인할 수 있다.

[0075] 파일접근 이벤트들을 확인한 결과 “txt”의 확장자를 가진 텍스트파일들을 암호화하는 데 걸린 시간이 나머지 확장자를 가진 파일들을 암호화 하는데 걸리는 시간에 비해서 월등히 빠르게 암호화 되는 것을 확인할 수 있었으며, 텍스트 파일을 빼고 다시 동일한 실험을 진행한 결과는 도 18에 도시된 바와 같다.

[0076] 도 18에 도시된 바와 같이, 텍스트 파일을 제외한 100개의 파일을 대상으로 실험을 진행하며, 10개의 파일을 하나의 실험단위로 묶어서 시간을 측정한 결과, 5개의 파일을 암호화 하는데 걸리는 평균 시간은 0.04초에서 최대 10.87초가 소모됐다. 텍스트 파일을 제외하고 나머지 파일들에 대해서만 탐지 할 경우에는 기존의 0.00175의 시간차이를 탐지하는 것에서 0.04초의 시간을 탐지하는 것으로 탐지 요구 시간이 늘어나는 것을 확인할 수 있다.

[0077] 상술한 실험을 크립토락커 샘플2에 대해 수행한 결과는 각각 도 19 및 도 20에 도시된 바와 같다.

[0078] 도 19에 도시된 바와 같이, 텍스트 파일을 포함하여 파일 암호화를 진행했을 때 5개의 파일을 암호화하는데 걸리는 평균시간이 최소 0.00175초에서 최대 0.4685초 소모되는 것을 확인할 수 있었으며, 도 20에 도시된 바와 같이, 텍스트파일을 제외하고 파일 암호화 시간을 측정했을 때, 최소 0.02초에서 최대 6.715초가 소모됨을 확인할 수 있다.

[0079] 도 19 및 도 20에 도시된 바와 같이, 두 번째 cryptolocker 샘플을 사용하면 텍스트파일을 포함했을 경우 최소 0.00175초의 탐지시간이 필요하고 텍스트파일을 제외하면 0.02초의 탐지시간이 필요함을 확인할 수 있다.

[0080] 이상에서 본 발명에 따른 바람직한 실시예에 대해 설명하였으나, 다양한 형태로 변형이 가능하며, 본 기술분야에서 통상의 지식을 가진 자라면 본 발명의 특허청구범위를 벗어남이 없이 다양한 변형예 및 수정예를 실시할 수 있을 것으로 이해된다.

부호의 설명

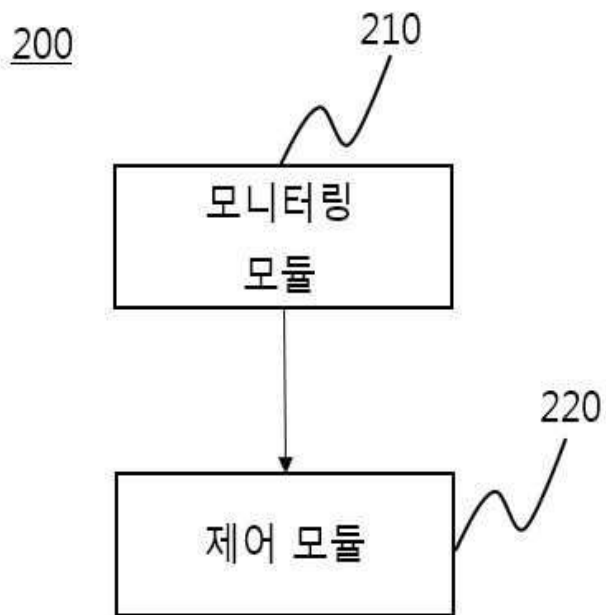
[0081] 200: 랜섬웨어 공격 차단 장치
210: 모니터링 모듈
220: 제어 모듈

도면

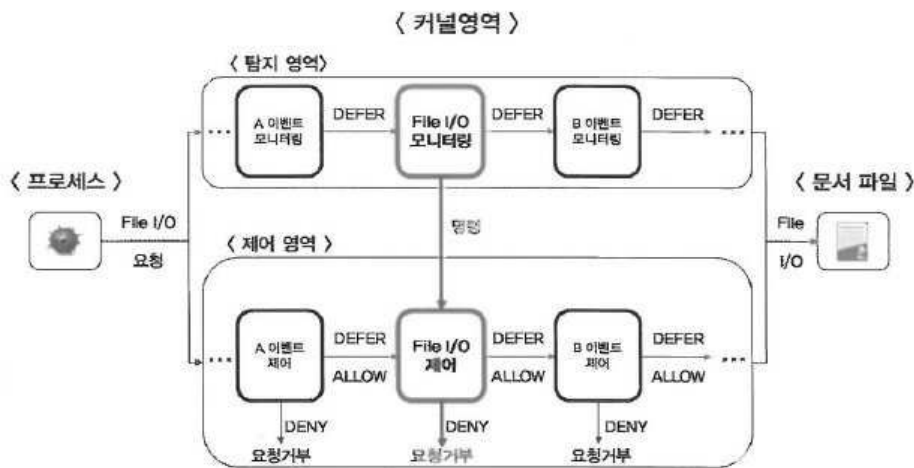
도면1



도면2



도면3



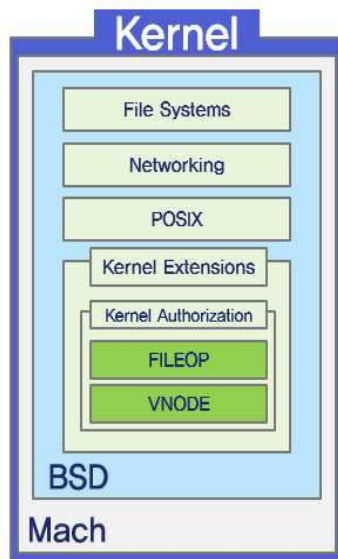
도면4

커널에서 수집 / 제어 가능한 FILE OPERATION	
FILE_OPEN	FILE_CLOSE
FILE_CLOSE_MODIFIED	FILE_RENAME
FILE_EXCHANGE	FILE_LINK
FILE_EXEC	FILE_READ_DATA
FILE_WRITE_DATA	FILE_EXECUTE
FILE_DELETE	FILE_APPEND_DATA
FILE_DELETE_CHILD	FILE_READ_ATTRIBUTES
FILE_WRITE_ATTRIBUTES	FILE_READ_EXTENDED_ATTRIBUTES
FILE_WRITE_EXTENDED_ATTRIBUTES	FILE_READ_SECURITY
FILE_WRITE_SECURITY	FILE_TAKE_OWNERSHIP
FILE_SYNCHRONIZE	FILE_LINKTARGET
FILE_CHECKIMMUTABLE	FILE_ACCESS
FILE_NOIMMUTABLE	

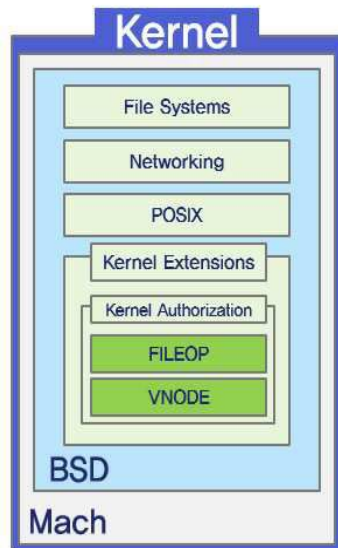
도면5



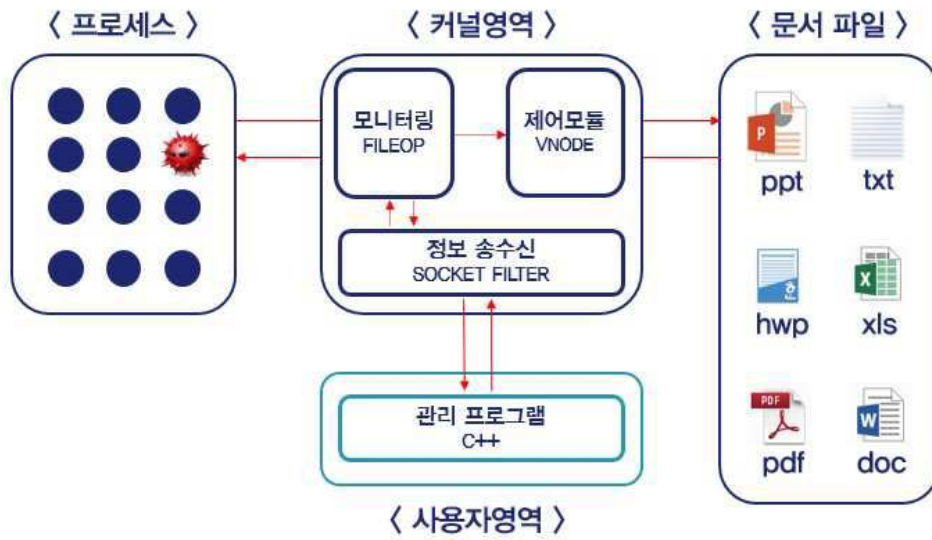
도면6



도면7



도면8

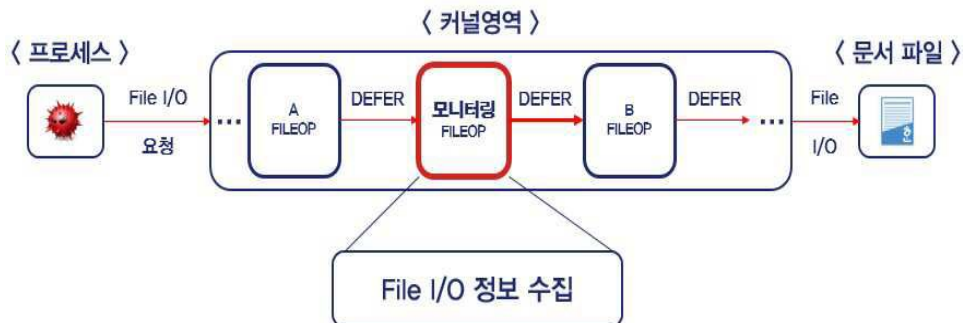


도면9

[모니터링 모듈]

Kernel Authorization기능 중 일부인 FILEOP를 이용한 모니터링 모듈

프로세스의 파일접근 요청 수집



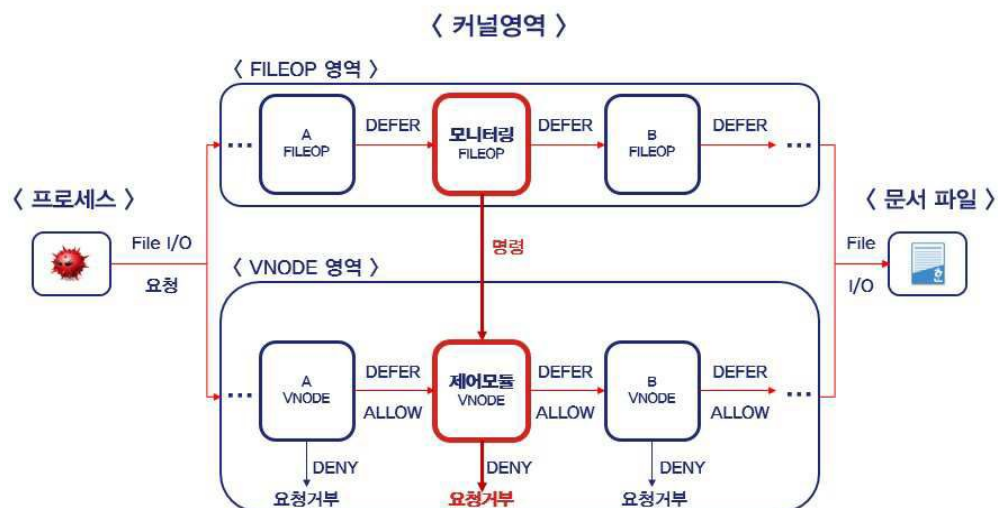
도면10



도면11



도면12



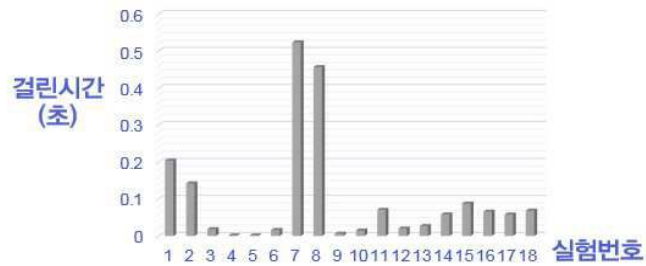
도면13

200개 파일 단위 암호화 실험

실험번호	파일 개수	걸린 시간	5개 파일 평균 암호화 시간
1	200개	8.15초	0.20374초
2	200개	5.69초	0.14225초
3	200개	0.75초	0.01875초
4	200개	0.07초	0.00175초
5	200개	0.07초	0.00175초
6	200개	0.66초	0.0165초
7	200개	20.94초	0.5235초
8	200개	18.27초	0.45675초
9	200개	0.26초	0.0065초
10	200개	0.6초	0.015초
11	200개	2.83초	0.07075초
12	200개	0.81초	0.02025초
13	200개	1.08초	0.027초
14	200개	2.34초	0.0585초
15	200개	3.51초	0.08775초
16	200개	2.64초	0.066초
17	200개	2.32초	0.056초
18	200개	2.75초	0.06875초

[크립토락커 샘플1 파일 암호화시간 측정]

	3600개	200개	5개
최소 시간	99.21 sec	0.07 sec	1.75 msec
최대 시간		20.94 sec	523.50 msec



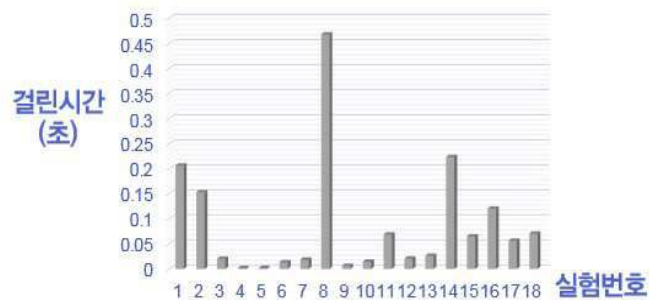
도면14

200개 파일 단위 암호화 실험

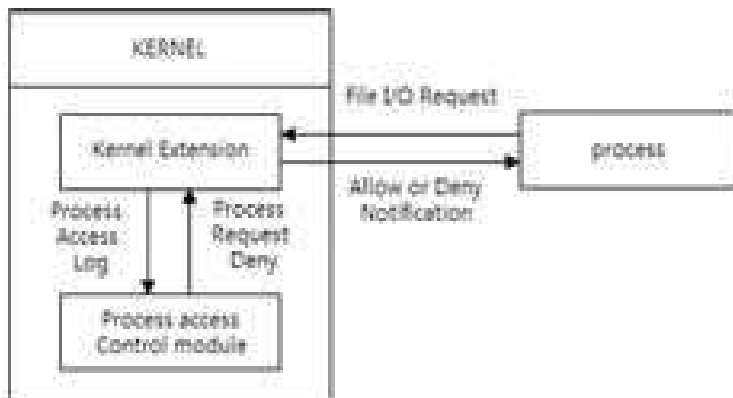
실험번호	파일 개수	걸린 시간	5개 파일 평균 암호화 시간
1	200개	8.26초	0.2095초
2	200개	8.13초	0.15325초
3	200개	0.83초	0.02075초
4	200개	0.07초	0.00175초
5	200개	0.06초	0.002초
6	200개	0.54초	0.0135초
7	200개	0.73초	0.01875초
8	200개	18.74초	0.4685초
9	200개	0.26초	0.0065초
10	200개	0.59초	0.0145초
11	200개	2.76초	0.069초
12	200개	0.84초	0.021초
13	200개	1.06초	0.0265초
14	200개	8.95초	0.22375초
15	200개	2.62초	0.0655초
16	200개	4.83초	0.12075초
17	200개	2.26초	0.0565초
18	200개	2.83초	0.07075초

[크립토락커 샘플2 파일 암호화시간 측정]

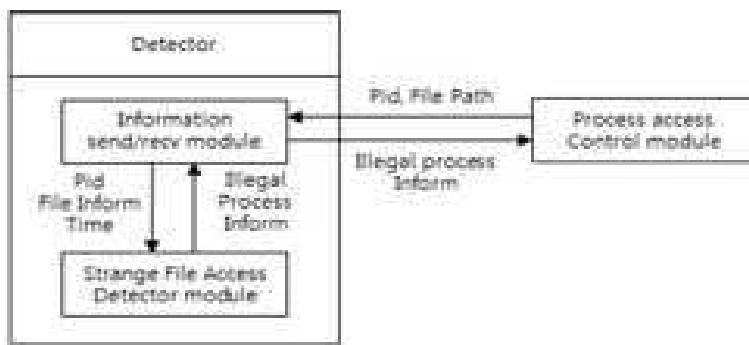
	3600개	200개	5개
최소 시간	62.37 sec	0.07 sec	1.75 msec
최대 시간		18.74 sec	468.50 msec



도면15



도면16



도면17

experiment	lead time	5 file encryption average
1	8.15	0.20374
2	5.69	0.14225
3	0.75	0.01875
4	0.07	0.00175
5	0.07	0.00175
6	0.66	0.0165
7	20.94	0.5235
8	18.27	0.45675
9	0.26	0.0065
10	0.6	0.015
11	2.83	0.7075
12	0.81	0.02025
13	1.08	0.027
14	2.34	0.0585
15	3.51	0.08775
16	2.64	0.066
17	2.32	0.058
18	2.75	0.06875

도면18

experiment	lead time	5 file encryption average
1	1.97	0.985
2	3.01	1.505
3	0.08	0.04
4	0.4	0.2
5	0.07	0.035
6	0.3	0.15
7	0.03	0.015
8	20.45	10.225
9	21.74	10.87
10	0.83	0.415

도면19

experiment	lead time	5 file encryption average
1	8.26	0.2065
2	6.13	0.15325
3	0.83	0.02075
4	0.07	0.00175
5	0.08	0.002
6	0.54	0.0135
7	0.73	0.01875
8	18.74	0.4685
9	0.26	0.0065
10	0.58	0.0145
11	2.76	0.069
12	0.84	0.021
13	1.06	0.0265
14	8.95	0.22375
15	2.62	0.0655
16	4.83	0.12075
17	2.26	0.0565
18	2.83	0.07075

도면20

experiment	lead time	5 file encryption average
1	2.49	1.245
2	13.43	6.715
3	0.07	0.035
4	0.23	0.115
5	0.09	0.045
6	0.2	0.1
7	0.04	0.02
8	0.17	0.085
9	11.99	5.995
10	0.6	0.3