(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2020년05월04일
(11) 등록번호 10-2105885
(24) 등록일자 2020년04월23일

(51) 국제특허분류(Int. Cl.)
G06F 21/56 (2013.01)
(52) CPC특허분류
G06F 21/56 (2013.01)
(21) 출원번호 10-2018-0152655
(22) 출원일자 2018년11월30일
심사청구일자 2018년11월30일
(56) 선행기술조사문헌
SungHa Baek, "SSD-Insider: Internal Defense of Solid-State Drive against Ransomware with Perfect Data Recovery", 2018 IEEE 38th International Conference on Distributed Computing Systems(2018.07.)
(뒷면에 계속)

(73) 특허권자
주식회사 심플한
인천광역시 미추홀구 소성로 192, 401호 (학익동, 남광빌딩)
인하대학교 산학협력단
인천광역시 미추홀구 인하로 100(용현동, 인하대학교)
(72) 발명자
백성하
서울특별시 구로구 디지털로27나길 12, 2층동 2호 (구로동)
양대헌
서울특별시 서초구 서초중앙로 200, 17동 901호 (서초동, 삼풍아파트)
(74) 대리인
특허법인남촌

전체 청구항 수 : 총 10 항 심사관 : 정성훈

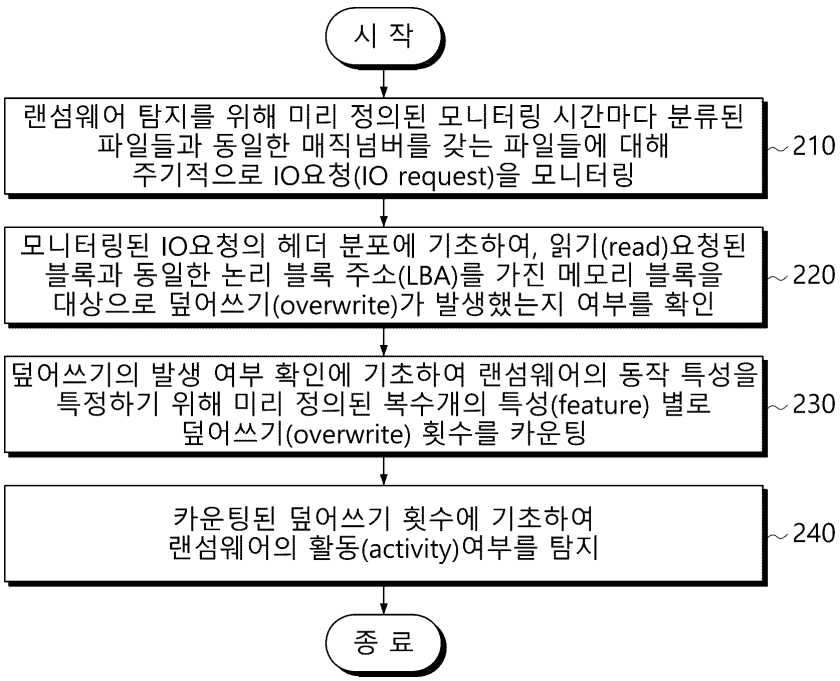
(54) 발명의 명칭 **랜섬웨어 탐지 방법 및 랜섬웨어 탐지 시스템**

(57) 요 약

본 발명은 NAND 플래시 메모리 기반 SSD(Solid State Drive) 내부에서 랜섬웨어(ransomeware)의 활동을 탐지(detect)하여 랜섬웨어를 방어할 수 있는 랜섬웨어 탐지 방법 및 랜섬웨어 탐지 시스템에 관한 것으로, 더욱 상세하게는, 랜섬웨어에 감염되는 파일들을 분류하고 랜섬웨어 탐지를 위해 미리 정의된 모니터링 시간마다 상기

(뒷면에 계속)

대 표 도 - 도2



분류된 파일들과 동일한 매직넘버(magic number)를 갖는 파일들에 대해 주기적으로 IO 요청(IO request)을 모니터링하는 단계; 모니터링된 IO 요청의 헤더(IO request header)의 분포에 기초하여, 읽기(read) 요청된 블록과 동일한 논리 블록 주소(LBA)를 가진 메모리 블록을 대상으로 덮어 쓰기(overwrite)가 발생했는지 여부를 확인하는 단계; 상기 덮어 쓰기의 발생 여부 확인에 기초하여 상기 랜섬웨어의 동작 특성을 특정하기 위해 미리 정의된 복수 개의 특성(feature) 별로 덮어 쓰기(overwrite) 횟수를 카운팅하는 단계; 및 카운팅된 상기 덮어 쓰기 횟수에 기초하여 랜섬웨어의 활동(activity) 여부를 탐지하는 단계;를 포함하는 랜섬웨어 탐지 방법 및 랜섬웨어 탐지 시스템에 관한 것이다.

(56) 선행기술조사문헌
KR100632204 B1
KR1020180001941 A
KR1020180060819 A
KR1020180030328 A
KR1020170119903 A

이 발명을 지원한 국가연구개발사업
과제고유번호 2018-0-00391
부처명 과학기술정보통신부
연구관리전문기관 정보통신기술진흥센터(IITP)
연구사업명 정보보호핵심원천기술개발사업
연구과제명 I/O 분포를 이용한 행위 기반의 랜섬웨어 탐지 기술
기 여 율 1/1
주관기관 인하대학교 산학협력단
연구기간 2018.04.01 ~ 2019.12.31

명 세 서

청구범위

청구항 1

NAND 플래시 메모리에서 동작하는 랜섬웨어 탐지 방법에 있어서,

랜섬웨어에 감염되는 파일들을 분류하고 랜섬웨어 탐지를 위해 미리 정의된 모니터링 시간마다 상기 분류된 파일들과 동일한 매직넘버(magic number)를 갖는 파일들에 대해 주기적으로 IO 요청(IO request)을 모니터링하는 단계;

모니터링된 IO 요청의 헤더(IO request header)의 분포에 기초하여, 읽기(read) 요청된 블록과 동일한 논리 블록 주소(LBA)를 가진 메모리 블록을 대상으로 덮어 쓰기(overwrite)가 발생했는지 여부를 확인하는 단계;

상기 덮어 쓰기의 발생 여부 확인에 기초하여 상기 랜섬웨어의 동작 특성을 특정하기 위해 미리 정의된 복수 개의 특성(feature) 별로 덮어 쓰기(overwrite) 횟수를 카운팅하는 단계; 및

카운팅된 상기 덮어 쓰기 횟수에 기초하여 랜섬웨어의 활동(activity) 여부를 탐지하는 단계;를 포함하는 랜섬웨어 탐지 방법.

청구항 2

제1항에 있어서,

상기 복수 개의 특성은, 미리 정의된 타임 슬라이스(times slice) 동안 발생한 덮어 쓰기(overwrite)의 횟수를 나타내는 OWIO, 상기 타임 슬라이스를 기반으로 하는 타임 윈도우(time window) 동안 발생한 쓰기 요청(write request)의 총 수 대비 덮어 쓰기(overwrite)가 발생한 블록의 비율을 나타내는 OWST, 타임 윈도우 동안 발생한 덮어 쓰기(overwrite)의 수를 나타내는 PWIO, 현재 타임 윈도우에서 연속적으로 덮어 쓰기가 발생한 블록(continuously overwritten block)의 평균 길이(length)를 나타내는 AVGWIO, 이전 타임 윈도우에서의 평균 덮어 쓰기(overwrite)의 개수 대 현재 타임 슬라이스(time slice)에서의 덮어 쓰기(overwrite)의 개수 간의 비율을 나타내는 OWSLOPE, 이전 타임 슬라이스에서의 평균 쓰기 횟수 대 현재 타임 슬라이스에서의 덮어 쓰기 횟수의 비율을 나타내는 IO를 포함하는 랜섬웨어 탐지 방법.

청구항 3

제1항에 있어서,

상기 NAND 플래시 메모리는 SSD(Solid State Drive)인 랜섬웨어 탐지 방법.

청구항 4

제1항에 있어서,

상기 랜섬웨어의 활동(activity) 여부를 탐지하는 단계에서는,

서로 다른 다양한 종류의 랜섬웨어의 활동 기간(active period) 동안 수집된 복수 개의 IO 요청을 상기 복수 개의 특성(feature) 별로 학습됨에 따라 생성된 이진 의사 결정 트리(binary decision tree)를 기반으로 모니터링된 상기 IO 요청에 해당하는 블록에 발생한 덮어 쓰기(overwrite)가 랜섬웨어의 활동(activity)에 의한 것인지 여부를 탐지하는 랜섬웨어 탐지 방법.

청구항 5

제4항에 있어서,

상기 랜섬웨어의 활동(activity) 여부를 탐지하는 단계는,

상기 복수 개의 특성(feature) 각각에 해당하는 특징값을 카운팅 테이블에 기초하여 계산하는 단계;

계산된 상기 복수 개의 특성 별 특징값을 상기 이진 의사 결정 트리의 입력 파라미터로 설정하여 상기 랜섬웨어의 활동 여부를 나타내는 결과값을 획득하는 단계; 및

획득된 결과값을 기반으로 상기 랜섬웨어가 활성 상태인지 여부를 결정하는 단계;를 포함하는 랜섬웨어 탐지 방법.

청구항 6

제5항에 있어서,

상기 결정하는 단계에서는,

미리 지정된 타임 윈도우 동안 획득된 복수의 결과값을 합산하고, 합산된 결과값이 미리 지정된 임계값(threshold)보다 크면, 랜섬웨어가 활성 상태인 것으로 결정하는 랜섬웨어 탐지 방법.

청구항 7

제1항에 있어서,

상기 랜섬웨어의 활동(activity) 여부를 탐지하는 단계에서는,

상기 모니터링된 IO 요청 별로 시간 정보, 논리 블록 주소(LBA) 정보, IO 모드(IO Mode) 정보, 및 길이(length) 정보가 연관하여 저장된 카운팅 테이블을 이용하여 상기 랜섬웨어의 활동 여부를 탐지하는 랜섬웨어 탐지 방법.

청구항 8

제7항에 있어서,

상기 카운팅 테이블은, 타임 슬라이스 동안 카운트된 상기 모니터링된 IO 요청의 횟수 및 상기 논리 블록 주소(LBA)에 기초하여 업데이트되는 랜섬웨어 탐지 방법.

청구항 9

제7항에 있어서,

상기 카운팅 테이블은, 미리 정의된 타임 윈도우 동안 상기 모니터링된 IO 요청 중 읽기(read) IO 요청과 동일한 논리 블록 주소(LBA)에 해당하는 쓰기(write) IO 요청을 선택적으로 저장하는 랜섬웨어 탐지 방법.

청구항 10

NAND 플래시 메모리에서 동작하는 랜섬웨어 탐지 시스템에 있어서,

랜섬웨어에 감염되는 파일들을 분류하고 랜섬웨어 탐지를 위해 미리 정의된 모니터링 시간마다 상기 분류된 파일들과 동일한 매직넘버(magic number)를 갖는 파일들에 대해 주기적으로 IO 요청(IO request)을 모니터링하는 모니터링부;

모니터링된 IO 요청의 헤더(IO request header)의 분포에 기초하여, 읽기(read) 요청된 블록과 동일한 논리 블록 주소(LBA)를 가진 메모리 블록을 대상으로 덮어 쓰기(overwrite)가 발생했는지 여부를 확인하는 확인부;

상기 덮어 쓰기의 발생 여부 확인에 기초하여 상기 랜섬웨어의 동작 특성을 특정하기 위해 미리 정의된 복수 개의 특성(feature) 별로 덮어 쓰기(overwrite) 횟수를 카운팅하는 카운팅부; 및

카운팅된 상기 덮어 쓰기 횟수에 기초하여 랜섬웨어가 활성(activity) 상태인지 여부를 탐지하는 탐지부;를 포함하는 랜섬웨어 탐지 시스템.

발명의 설명

기 술 분 야

[0001] 본 발명은 랜섬웨어 탐지 방법 및 랜섬웨어 탐지 시스템에 관한 것으로, 특히, NAND 플래시 메모리 기반 SSD(Solid State Drive) 내부에서 랜섬웨어(ransomeware)의 활동을 탐지(detect)하여 랜섬웨어를 방어할 수 있는 랜섬웨어 탐지 방법 및 랜섬웨어 탐지 시스템에 관한 것이다.

배 경 기 술

[0003] 금전적 이익을 위해 사용자의 데이터를 인질로 보유하는 악성 소프트웨어 유형인 랜섬웨어로 인한 피해가 증가하고 있으며, 랜섬웨어(ransomeware)는 토르(Tor)와 같은 익명 통신 시스템을 사용하는 복잡한 명령 및 제어(C&C) 네트워크를 사용하고 추적하기 어려운 비트 코인과 같은 가상화폐를 사용하여 금전적 이득을 수집함으로써 표준 멀웨어(malware) 방어를 파괴하려고 시도하고 있다. 랜섬웨어는 공격자에게 잠재적으로 높은 금전적 이득을 주고 랜섬웨어에 대한 방어를 어렵게 함으로써 사이버 범죄자들에게 "수익성 높은 사업"을 제공하고 있어, 랜섬웨어로 인한 피해 사례가 상당하다. 예를 들어 한국의 Nayana 웹 호스팅 회사는 2017년 Erebus라는 암호화된 랜섬웨어에 의해 공격 받았으며, 한국 인터넷 진흥원(KISA)에 따르면 3,400개 이상의 웹사이트가 랜섬웨어의 영향을 받았다. Nayana는 해커에게 111만 달러를 3회에 나누어서 제공하여 키를 받고 해킹된 데이터를 복구하였으나, 데이터를 완전히 복구 할 수 없었다. 미국에서는 2016년 Hollywood Presbyterian Medical Center가 암호화된 랜섬웨어에 감염되어 의료 기록에 액세스 할 수 없게 되었고, 이로 인하여 수술이 불가능하게 된 날로부터 10일 후에 40 비트 코인(당시 약 17,000달러)을 공격자에게 지불한 사례가 있다. 2015년에 100건 이상의 랜섬웨어가 보고되었으며, 2016년 말에 평균 갈취액은 679달러로 전년보다 두 배가 증가하였다. 모든 경제 부문이 랜섬웨어의 타겟이며, 2015년에 기관 감염이 38%를 차지하였고 제조 부문 감염이 17 %를 차지하였다. 이처럼, 랜섬웨어로 인한 피해가 증가함에 따라 안티 랜섬웨어의 필요성이 증가하고 있다.

[0004] 일반적으로 랜섬웨어는 락커(locker) 랜섬웨어와 암호화 랜섬웨어로 분류된다. 락커 랜섬웨어는 사용자가 감염된 시스템에 액세스하는 것을 방지하는 것이고, WannaCry, CryptoWall, TeslaCrypt(a.k.a. AlphaCrypt), Locky 등의 암호화 랜섬웨어는 사용자의 데이터 액세스를 방지하기 위해 데이터를 암호화하여 사용자의 데이터를 잠그는 악성 소프트웨어이다. 2016년에 새로 발견된 랜섬웨어의 대부분은 암호화 랜섬웨어로서, 암호화 랜섬웨어는 2015년랜섬웨어의 80%에 달한다. 이와 같이, 암호화 랜섬웨어의 비율이 증가하므로, 암호화 랜섬웨어를 방어하기 위한 기술이 요구된다.

[0005] 아래의 비특허 문헌 [1] G. Canfora, F. Mercaldo, and C. A. Visaggio, "An hmm and structural entropy based detector for android malware: An empirical study," Computers & Security, vol. 61, pp. 1-18, 2016.에서는 엔트로피와 히든 마코브 모델(hidden markov model)을 이용하여 랜섬웨어의 행동 분석에 초점을 맞추어 대량의 암호화 시스템을 사용하는 기술을 제시하고 있으나, 어플리케이션 계층에서 소프트웨어 툴을 실행하며, 파일 이름, 파일 크기, 매직 번호, 파일 내용 변경과 같은 파일의 메타 데이터 변경을 모니터링하는 수준에 불과하다. 아래의 비특허 문헌 [2] N. Scaife, H. Carter, P. Traynor, and K. Butler, "Cryptolock (and drop it): Stopping ransomware attacks on user data," in Proceedings of Distributed Computing Systems (ICDCS). 2016 IEEE International Conference on, 2016, pp. 303-312.과 [3] A. Kharaz, S. Arshad, C. Mulliner, W. Robertson, and E. Kirda, “Unveil: A large-scale, automated approach to detecting ransomware," in Proceedings of the 8th ACM International Conference on Embedded Software, ser. USENIX Security '16. USENIX, 2016, pp. 757-772.에 제시된 파일 모니터링 방법은 랜섬웨어의 의심스러운 행동을 조기에 탐지하나, 응용 프로그램 레벨에서의 모니터링으로서 여러가지 제한을 가지고 있다. 즉, 응용 프로그램 레벨에서 동작하는 랜섬웨어 탐지 기술의 경우, 사용자가 백신 소프트웨어를 설치하지 않는 한 랜섬웨어의 활동을 막을 수 없으며, 랜섬웨어의 공격을 탐지하기 전에 랜섬웨어에 의해 이미 암호화된 파일은 복구될 수 없어 데이터 손실이 불가피하다.

[0006] 또한, 응용 프로그램 레벨의 모니터링은 모든 IO 및 파일 시스템 수정을 모니터링해야 하므로 많은 리소스

(resource)를 소비하게 되어 시스템 성능을 저하시킨다. 뿐만 아니라, 대부분의 랜섬웨어는 마이크로소프트 윈도우(Microsoft Windows) 운영 체제를 대상으로 개발되었기 때문에 리눅스(Linux) 및 맥OS(MacOS)와 같은 다른 운영 체제에 적합한 방어 기술은 아직 잘 연구되지 않았다.

[0007] 따라서 각 플랫폼(파일 시스템, 운영 체제, 하드웨어 구성 등)의 고유 기능을 고려하여 여러 버전의 방어 시스템이 요구된다. 다시 말해, 사용자가 백신 소프트웨어를 설치하지 않았더라도, 운영 체제의 종류에 관계없이 랜섬웨어를 탐지하여 랜섬웨어의 공격에 방어할 수 있는 기술이 요구된다.

[0008] 한국등록특허 제10-1685014호는 컴퓨터 시스템의 랜섬웨어 행위에 대한 선제적인 탐지 차단 방법 및 그 장치에 관한 것으로, 운영체제 상에서 최초로 접근할 수 밖에 없는 경로 상에 함정 파일을 생성하여 두고, 함정 파일에 접근하여 암호화 등의 변경 시도를 행하는 랜섬웨어 행위를 탐지하여 차단하는 기술을 개시하고 있다.

선행기술문헌

특허문헌

[0010] (특허문헌 0001) (비특허문헌 1)[1] G. Canfora, F. Mercaldo, and C. A. Visaggio, "An hmm and structural entropy based detector for android malware: An empirical study," Computers & Security, vol. 61, pp. 1-18, 2016.

(특허문헌 0002) (비특허문헌 2)[2] N. Scaife, H. Carter, P. Traynor, and K. Butler, “Cryptolock (and drop it): Stopping ransomware attacks on user data,” in Proceedings of Distributed Computing Systems (ICDCS). 2016 IEEE International Conference on, 2016, pp. 303-312.

(특허문헌 0003) (비특허문헌 3)[3] A. Kharaz, S. Arshad, C. Mulliner, W. Robertson, and E. Kirda, “Unveil: A large-scale, automated approach to detecting ransomware,” in Proceedings of the 8th ACM International Conference on Embedded Software, ser. USENIX Security ’16. USENIX, 2016, pp. 757-772.

(특허문헌 0004) (비특허문헌 4)[4] J. R. Quinlan, "Induction of decision trees," Mach. Learn., vol. 1, no. 1, pp. 81-106, 1986.

발명의 내용

해결하려는 과제

[0011] 본 발명이 해결하고자 하는 기술적 과제는, 응용 프로그램 레벨이 아닌 NAND 플래시 메모리 내부에서 랜섬웨어의 활동으로 인한 SSD(Solid State Drive)의 비정상적인 특성을 신속히 탐지하여 랜섬웨어의 공격에 방어하고, SSD의 지연 삭제 특성을 기반으로 랜섬웨어로 인해 삭제된 파일을 복구하는 랜섬웨어 탐지 방법 및 랜섬웨어 탐지 시스템을 제공하는데 있다.

[0012] 본 발명이 해결하고자 하는 다른 기술적 과제는, 랜섬웨어에 감염될 수 있는 이미지나 문서 등과 같은 파일들을 분류하고 파일 형식을 지정하는 매직넘버(magic number)를 통해 분류된 파일들과 동일한 매직넘버를 갖는 파일들에 대해 신속히 랜섬웨어 감염 여부를 탐지할 수 있는 랜섬웨어 탐지 방법 및 랜섬웨어 탐지 시스템을 제공하는데 있다.

과제의 해결 수단

[0014] 상기 기술적 과제를 달성하기 위한 본 발명의 특징에 따르면, 본 발명인 랜섬웨어 탐지 방법은, NAND 플래시 메모리에서 동작하는 랜섬웨어 탐지 방법에 있어서, 랜섬웨어에 감염되는 파일들을 분류하고 랜섬웨어 탐지를 위해 미리 정의된 모니터링 시간마다 상기 분류된 파일들과 동일한 매직넘버(magic number)를 갖는 파일들에 대해 주기적으로 IO 요청(IO request)을 모니터링하는 단계; 모니터링된 IO 요청의 헤더(IO request header)의 분포에 기초하여, 읽기(read) 요청된 블록과 동일한 논리 블록 주소(LBA)를 가진 메모리 블록을 대상으로 덮어 쓰기(overwrite)가 발생했는지 여부를 확인하는 단계; 상기 덮어 쓰기의 발생 여부 확인에 기초하여 상기 랜섬웨어의 동작 특성을 특정하기 위해 미리 정의된 복수 개의 특성(feature) 별로 덮어 쓰기(overwrite) 횟수를 카운팅하는 단계; 및 카운팅된 상기 덮어 쓰기 횟수에 기초하여 랜섬웨어의 활동(activity) 여부를 탐지하는 단계;를

포함하는 것을 특징으로 한다.

[0015] 또한, 상기 복수 개의 특성은, 미리 정의된 타임 슬라이스(times slice) 동안 발생한 덮어 쓰기(overwrite)의 횟수를 나타내는 OWIO, 상기 타임 슬라이스를 기반으로 하는 타임 윈도우(time window) 동안 발생한 쓰기 요청(write request)의 총 수 대비 덮어 쓰기(overwrite)가 발생한 블록의 비율을 나타내는 OWST, 타임 윈도우 동안 발생한 덮어 쓰기(overwrite)의 수를 나타내는 PWIO, 현재 타임 윈도우에서 연속적으로 덮어 쓰기가 발생한 블록(continuously overwritten block)의 평균 길이(length)를 나타내는 AVGWIO, 이전 타임 윈도우에서의 평균 덮어 쓰기(overwrite)의 개수 대 현재 타임 슬라이스(time slice)에서의 덮어 쓰기(overwrite)의 개수 간의 비율을 나타내는 OWSLOPE, 이전 타임 슬라이스에서의 평균 쓰기 횟수 대 현재 타임 슬라이스에서의 덮어 쓰기 횟수의 비율을 나타내는 IO를 포함하는 것을 특징으로 한다.

[0016] 또한, 상기 NAND 플래시 메모리는 SSD(Solid State Drive)를 나타내는 것을 특징으로 한다.

[0017] 또한, 상기 랜섬웨어의 활동(activity) 여부를 탐지하는 단계는, 서로 다른 다양한 종류의 랜섬웨어의 활동 기간(active period) 동안 수집된 복수 개의 IO 요청을 상기 복수 개의 특성(feature) 별로 학습됨에 따라 생성된 이진 의사 결정 트리(binary decision tree)를 기반으로 모니터링된 상기 IO 요청에 해당하는 블록에 발생한 덮어 쓰기(overwrite)가 랜섬웨어의 활동(activity)에 의한 것인지 여부를 탐지하는 것을 특징으로 한다.

[0018] 또한, 상기 랜섬웨어의 활동(activity) 여부를 탐지하는 단계는, 상기 복수 개의 특성(feature) 각각에 해당하는 특징값을 카운팅 테이블에 기초하여 계산하는 단계; 계산된 상기 복수 개의 특성 별 특징값을 상기 이진 의사 결정 트리의 입력 파라미터로 설정하여 상기 랜섬웨어의 활동 여부를 나타내는 결과값을 획득하는 단계; 및 획득된 결과값을 기반으로 상기 랜섬웨어가 활성 상태인지 여부를 결정하는 단계;를 포함하는 것을 특징으로 한다.

[0019] 또한, 상기 결정하는 단계는, 미리 지정된 타임 윈도우 동안 획득된 복수의 결과값을 합산하고, 합산된 결과값이 미리 지정된 임계값(threshold)보다 크면, 랜섬웨어가 활성 상태인 것으로 결정하는 것을 특징으로 한다.

[0020] 또한, 상기 랜섬웨어의 활동(activity) 여부를 탐지하는 단계는, 상기 모니터링된 IO 요청 별로 시간 정보, 논리 블록 주소(LBA) 정보, IO 모드(IO Mode) 정보, 및 길이(length) 정보가 연관하여 저장된 카운팅 테이블을 이용하여 상기 랜섬웨어의 활동 여부를 탐지하는 것을 특징으로 한다.

[0021] 또한, 상기 카운팅 테이블은, 타임 슬라이스 동안 카운트된 상기 모니터링된 IO 요청의 횟수 및 상기 논리 블록 주소(LBA)에 기초하여 업데이트되는 것을 특징으로 한다.

[0022] 또한, 상기 카운팅 테이블은, 미리 정의된 타임 윈도우 동안 상기 모니터링된 IO 요청 중 읽기(read) IO 요청과 동일한 논리 블록 주소(LBA)에 해당하는 쓰기(write) IO 요청을 선택적으로 저장하는 것을 특징으로 한다.

[0023] 또한, 본 발명인 램섬웨어 탐지 시스템은, NAND 플래시 메모리에서 동작하는 랜섬웨어 탐지 시스템에 있어서, 랜섬웨어에 감염되는 파일들을 분류하고 랜섬웨어 탐지를 위해 미리 정의된 모니터링 시간마다 상기 분류된 파일들과 동일한 매직넘버(magic number)를 갖는 파일들에 대해 주기적으로 IO 요청(IO request)을 모니터링하는 모니터링부; 모니터링된 IO 요청의 헤더(IO request header)의 분포에 기초하여, 읽기(read) 요청된 블록과 동일한 논리 블록 주소(LBA)를 가진 메모리 블록을 대상으로 덮어 쓰기(overwrite)가 발생했는지 여부를 확인하는 확인부; 상기 덮어 쓰기의 발생 여부 확인에 기초하여 상기 랜섬웨어의 동작 특성을 특정하기 위해 미리 정의된 복수 개의 특성(feature) 별로 덮어 쓰기(overwrite) 횟수를 카운팅하는 카운팅부; 및 카운팅된 상기 덮어 쓰기 횟수에 기초하여 랜섬웨어가 활성(activity) 상태인지 여부를 탐지하는 탐지부;를 포함하는 것을 특징으로 한다.

발명의 효과

[0025] 이상에서 상술한 본 발명에 의한 랜섬웨어 탐지 방법 및 랜섬웨어 탐지 시스템은 다음과 같은 효과가 있다.

[0026] 먼저, SSD 내부에 펌웨어(firmware) 형태로 구현되어, SSD의 비정상적인 특성을 기반으로 랜섬웨어의 활동(activity)을 탐지함에 따라, 백신 등의 랜섬웨어 응용 프로그램을 실행하지 않아도 랜섬웨어를 탐지할 수 있을 뿐만 아니라, OS(또는 미들웨어)가 랜섬웨어 모니터링 응용 프로그램을 실행하지 않더라도 랜섬웨어의 활동을 탐지할 수 있다.

[0027] 또한, SSD 내부에 펌웨어 형태로 구현되어 동작함에 따라, OS의 종류에 관계없이, 즉, 알려지지 않은 OS, 파일 시스템 및 응용 프로그램에서도 랜섬웨어의 활동을 탐지할 수 있다.

[0028] 또한, 랜섬웨어의 활동(activity)을 특정하는 6가지의 특성(feature)을 기반으로 랜섬웨어의 활동 여부를 탐지함으로써, 현재까지 알려진 랜섬웨어 이외에 알려지지 않은 랜섬웨어의 활동 여부를 탐지할 수 있다.

[0029] 또한, 랜섬웨어에 감염될 수 있는 파일들을 정의하고, 정의된 파일들과 동일한 매직넘버를 갖는 파일들에 대해서 랜섬웨어 감염 여부를 확인하므로 신속한 검사 및 탐지가 가능하다.

도면의 간단한 설명

[0031] 도 1은 본 발명의 일실시예에 있어서, 랜섬웨어 탐지 시스템의 내부 구성을 설명하기 위한 블록도이다.

도 2는 본 발명의 일실시예에 있어서, 랜섬웨어 탐지 방법을 도시한 흐름도이다.

도 3은 본 발명의 일실시예에 있어서, 랜섬웨어를 특정하는 6가지 특성 중 OWIO관련 그래프이다.

도 4는 본 발명의 일실시예에 있어서, 랜섬웨어를 특정하는 6가지 특성 중 OWIO를 제외한 5가지 특성관련 그래프이다.

도 5는 본 발명의 일실시예에 있어서, 학습된 6가지 특성(feature)을 기반으로 랜섬웨어의 활동(activity)을 탐지하는 세부 동작을 도시한 흐름도이다.

도 6은 본 발명의 일실시예에 있어서, 카운팅 테이블을 도시한 도면이다.

도 7은 본 발명의 일실시예에 있어서, 기본 함수(basic function)를 이용하여 업데이트된 카운팅 테이블을 도시한 도면이다.

도 8은 본 발명의 일실시예에 있어서, 트리 결과값을 합산하여 최종적으로 랜섬웨어의 활동 여부를 결정하기 위한 스코어(score)를 계산하는 동작을 설명하기 위해 제공되는 도면이다.

발명을 실시하기 위한 구체적인 내용

[0032] 이하, 본 발명의 일부 실시례들을 예시적인 도면을 통해 상세하게 설명한다. 각 도면의 구성요소들에 참조부호를 부가함에 있어서, 동일한 구성요소들에 대해서는 비록 다른 도면상에 표시되더라도 가능한 한 동일한 부호를 가지도록 하고 있음에 유의해야 한다. 또한, 본 발명의 실시례를 설명함에 있어, 관련된 공지구성 또는 기능에 대한 구체적인 설명이 본 발명의 실시례에 대한 이해를 방해한다고 판단되는 경우에는 그 상세한 설명은 생략한다.

[0033] 또한, 본 발명의 실시례의 구성 요소를 설명하는 데 있어서, 제 1, 제 2, A, B, (a), (b) 등의 용어를 사용할 수 있다. 이러한 용어는 그 구성 요소를 다른 구성 요소와 구별하기 위한 것일 뿐, 그 용어에 의해 해당 구성 요소의 본질이나 차례 또는 순서 등이 한정되지 않는다. 어떤 구성 요소가 다른 구성요소에 "연결", "결합" 또는 "접속"된다고 기재된 경우, 그 구성 요소는 그 다른 구성요소에 직접적으로 연결되거나 접속될 수 있지만, 각 구성 요소 사이에 또 다른 구성 요소가 "연결", "결합" 또는 "접속"될 수도 있다고 이해되어야 할 것이다.

[0035] 먼저, 랜섬웨어는 신속한 탐지가 관건이라 할 수 있다. 즉, 일정 시간(평균 10초) 이내에 랜섬웨어를 감지하고 이에 대해 대처하지 못한다면, 일정 시간이 경과한 후에는 랜섬웨어에 대한 감염 여부를 파악하더라도 이미 대부분의 파일들이 랜섬웨어에 감염된 상태일 수 있다.

[0036] 따라서, 본 발명인 랜섬웨어 탐지 방법 및 랜섬웨어 탐지 시스템에서는 모든 파일들에 대해 랜섬웨어 감염 여부를 확인하지 않고 랜섬웨어에 감염될 수 있는 이미지나 문서 등과 같은 파일들을 정의한 후 정의된 파일들에 대해서만 기설정된 시간마다 주기적으로 랜섬웨어 감염 여부를 확인한다. 즉, 정의(분류)된 파일들과 동일한 매직넘버(magic number)를 갖는 파일들에 대해서만 랜섬웨어 감염 여부를 확인함으로써, 랜섬웨어 감염 여부에 대한 신속한 탐지 및 대처가 가능하다.

[0037] 매직넘버(magic number)란 파일 형식이나 프로토콜을 식별하는데 사용되는 상수 또는 텍스트값 파일 서명, 파일의 내용을 확인하거나 확인하는데 사용되는 데이터의 목록이다. 이러한 서명은 마술 번호라고도 하며, 파일 서명(file signature)과 유사한 개념으로 사용될 때도 있다.

[0038] 매직 번호는 여러 운영 체제에서 공통적으로 나타나는 프로그램이다. 매직 번호는 강력한 형식의 데이터를 구현하며 프로그램 실행 시간에 데이터 형식을 읽는 제어 프로그램에 대한 대역 내 신호 형식이다. 많은 파일에는 포함된 데이터를 식별하는 상수가 있다. 파일에서 이러한 상수를 감지하는 것은 많은 파일 형식을 구분하는 간

단하고 효과적인 방법이며 추가 런타임 정보를 산출할 수 있다.

[0039] 매직 번호는 일례로 다음과 같이 표현될 수 있다.

[0040] 컴파일된 Java 클래스 파일(바이트 코드) 및 Mach-O 바이너리는 16진수 'CAFEBABE'로 시작하며, 'Pack200'로 압축하면 바이트가 'CAFED00D'로 바뀐다. GIF 이미지 파일에는 "GIF89a"(47 49 46 38 39 61) 또는 "GIF87a"(47 49 46 38 37 61)에 대한 ASCII 코드가 있다. JPEG 이미지 파일은 'FF D8'로 시작하여 'FF D9'로 끝나며, PDF 파일은 "% PDF"(16진수 25 50 44 46)로 시작한다. ZIP 파일의 헤더는 DOS 압축 유틸리티 'PKZIP'의 필자인 Phil Katz의 머리 글자 "PK"(50 4B)로 시작한다.

[0041] 하기 [표 1]은 랜섬웨어에 감염될 수 있는 대표적인 파일 형식에 따른 헤더(Header)와 풋터(Footer)의 매직넘버를 나타낸 표이다.

표 1

파일 포맷 (extension/format)	시작 값 (Header)	마지막 값 (Footer)
JPEG	FF D8 FF E0 00 10 4A 46 49 46 00 01 01	
GIF	47 49 46 38 39 61 4E 01 53 00 C4	21 00 00 3B 00
PNG	89 50 4E 47 0D 0A 1A 0A	
TIFF (TIF)	49 49 2A 00 4D 4D 00 2A	
DOC	D0 CF 11 E0 A1 B1 1A E1	57 6F 72 64 2E 44 6F 63 75 6D 65 6E 74 2E
XLS	D0 CF 11 E0 A1 B1 1A E1	FE FF FF FF 00 00 00 00 00 00 00 00 57 00 6F 00 72 00 6B 00 62 00 6F 00 6F 00 6B 00
PPT	D0 CF 11 E0 A1 B1 1A E1	50 00 6F 00 77 00 65 00 72 00 50 00 6F 00 69 00 6E 00 74 00 20 00 44 00 6F 00 63 00 75 00 6D 00 65 00 6E 00 74
ZIP	50 4B 03 04 14	50 4B 05 06 00
PCAP	D4 C3 B2 A1	

[0042]

[0043] 상기 [표 1]과 같이 각각의 파일 포맷에 따른 매직넘버는 다르게 표현된다. 따라서, 파일 형식에 따른 고유 식별 번호(부호)라 할 수 있는 매직넘버를 통해 정의된 파일들에 대해서 기설정된 주기마다 검사를 수행함으로써 랜섬웨어에 대한 탐지 효율성을 향상시킬 수 있다. 한편, 랜섬웨어에 감염되는 파일에 따라서 미리 분류된 파일 이외에도 해당 파일의 매직넘버가 추가될 수 있음은 당연하다 할 것이다.

[0044] 본 발명은 NAND 플래시 기반의 SSD 내부에서 동작하도록 SSD에 펌웨어 형태로 구현되어 랜섬웨어(ransomeware)의 활동(activity)을 탐지하는 기술에 관한 것으로서, 특히, 어플리케이션 계층이나 OS 계층에서 랜섬웨어를 탐지하는 것이 아니라 IO 요청(IO request) 및 암호화 랜섬웨어를 특정하는 6가지 특성(feature)를 기반으로 SSD 내부에서 랜섬웨어의 활동을 탐지하고, 삭제된 파일을 복구하는 기술에 관한 것이다. 즉, 본 실시예들에서, 랜섬웨어 탐지 방법은 SSD 펌웨어의 일부로 동작할 수 있으며, 제한된 리소스(예컨대, CPU 전원 및 메모리)만으로 랜섬웨어를 탐지할 수 있다.

[0045] 본 실시예들에서, 랜섬웨어 탐지 방법 및 시스템은 전체 페이로드(payload) 대신 IO 요청 헤더의 분포만을 확인하여 랜섬웨어의 활동(activity)을 인식하므로, 경량이고, 변하지 않는 랜섬웨어의 독특한 동작 특성을 이용하여 랜섬웨어를 탐지할 수 있다. 이에 따라, 상기와 같은 매직넘버 이외에 랜섬웨어의 독특한 동작 특성인 덮어쓰기(overwrite)를 특정하는 6가지 특성(feature)들이 랜섬웨어 탐지를 위해 이용될 수 있다.

[0046] 본 실시예들에서, 암호화 랜섬웨어를 특정하는 '6가지 특성(feature)'은 사용자의 원래 파일을 삭제하고, 삭제된 위치에 새로운 데이터(예컨대, 임의의 랜덤 데이터 또는 해커에 의해 암호화된 원본 데이터)를 기록하여 사용자가 원본 데이터를 복구하지 못하도록 덮어 쓰기(overwrite)하는 랜섬웨어의 특성들을 나타내는 것으로서, 미리 정의된 타임 슬라이스(times slice) 동안 발생한 덮어 쓰기(overwrite)의 횟수를 나타내는 OWIO, 타임 슬라이스를 기반으로 하는 타임 윈도우(time window) 동안 발생한 쓰기 요청(write request)의 총 수 대비 덮어 쓰기(overwrite)가 발생한 블록의 비율을 나타내는 OWST, 타임 윈도우 동안 발생한 덮어 쓰기(overwrite)의 수를 나타내는 PWIO, 현재 타임 윈도우에서 연속적으로 덮어 쓰기가 발생한 블록(continuously overwritten block)의 평균 길이(length)를 나타내는 AVGWIO, 이전 타임 윈도우에서의 평균 덮어 쓰기(overwrite)의 개수 대

현재 타임 슬라이스(time slice)에서의 덮어 쓰기(overwrite)의 개수 간의 비율을 나타내는 OWSLOPE, 이전 타임 슬라이스에서의 평균 쓰기 횟수 대 현재 타임 슬라이스에서의 덮어 쓰기 횟수의 비율을 나타내는 IO를 포함할 수 있다.

[0047] 본 실시예들에서, 'IO 요청(IO request)'은 파일 이름, 파일 크기, 프로세스 ID, 프로세스 이름 등 많은 양의 정보를 포함하는 것이 아니라 쓰기 및 읽기 요청(write request, read request) 만을 포함할 수 있다. 즉, IO 요청만을 확인하여 랜섬웨어의 활동(activity) 여부를 탐지할 수 있다.

[0048] 본 실시예들에서, '덮어 쓰기(overwrite) '는 메모리 블록을 읽은 후 동일한 논리 블록 주소(LBA)를 가진 메모리 블록을 업데이트하여 해당 블록의 데이터를 제거/삭제하는 것을 나타낼 수 있다.

[0049] 본 실시예들에서, '타임 슬라이스(time slice)'는 미리 지정된 타임 윈도우(window)를 복수 개의 구간으로 구분한 것으로서, 예컨대, 타임 윈도우가 10초인 경우, 타임 슬라이스는 1초 등으로 구분될 수 있다. 즉, 10초의 타임 윈도우가 1초의 타임 슬라이스 10개로 분할될 수 있다.

[0050] 도 1은 본 발명의 일실시예에 있어서, 랜섬웨어 탐지 시스템의 내부 구성을 설명하기 위한 블록도이고, 도 2는 본 발명의 일실시예에 있어서, 랜섬웨어 탐지 방법을 도시한 흐름도이다.

[0051] 본 실시예에 따른 랜섬웨어 탐지 시스템(100)은 SSD의 내부에 펌웨어 형태로 구현되어 컴퓨터 프로그램의 명령을 처리하도록 구성된 컴퓨팅 시스템으로서, 모니터링부(110), 확인부(120), 카운팅부(130) 및 탐지부(140)를 포함할 수 있다. 예를 들어, 랜섬웨어 탐지 시스템(100)는 컴퓨팅 시스템의 프로세서로서, 모니터링부(110), 확인부(120), 카운팅부(130) 및 탐지부(140)를 위한 프로그램 코드를 실행하도록 구성될 수 있다.

[0052] 210 단계에서, 모니터링부(110)는 랜섬웨어에 감염될 수 있는 분류된 파일들에 대한 랜섬웨어 탐지를 위해 미리 정의된 모니터링 시간마다 상기 분류된 파일들과 동일한 매직넘버(magic number)를 갖는 파일들에 대해 미리 정의된 모니터링 시간마다 주기적으로 IO 요청(IO request)를 모니터링할 수 있다.

[0053] 예를 들어, 모니터링부(110)는 타임 윈도우(예컨대, 10초) 동안 IO 요청을 모니터링하여 수집할 수 있다. 즉 10초동안 발생한 읽기(read) 요청 및 쓰기(write) 요청을 모두 가지고 있을 수 있다.

[0054] 220 단계에서, 확인부(120)는 모니터링된 IO 요청의 헤더의 분포에 기초하여 읽기(read) 요청된 블록과 동일한 논리 블록 주소(Logical Block Address, LBA)를 가진 메모리 블록을 대상으로 덮어 쓰기(overwrite)가 발생했는지 여부를 확인할 수 있다.

[0055] 230 단계에서, 카운팅부(130)는 덮어 쓰기의 발생 여부 확인에 기초하여 랜섬웨어의 동작 특성을 특정하는 6개의 특성 별로 덮어 쓰기 횟수를 카운팅할 수 있다. 여기서, 랜섬웨어의 동작 특성(즉, 활동)을 특정하는 6개의 특성(feature)은 OWIO, OWST, PWIO, AVGWIO, OWSLOPE 및 IO를 포함할 수 있다.

[0056] OWIO는 미리 정의된 타임 슬라이스(times slice) 동안 발생한 덮어 쓰기(overwrite)의 횟수를 나타내고, OWST는 타임 슬라이스를 기반으로 하는 타임 윈도우(time window) 동안 발생한 쓰기 요청(write request)의 총 수 대비 덮어 쓰기(overwrite)가 발생한 블록의 비율을 나타낼 수 있다. PWIO는 타임 윈도우 동안 발생한 덮어 쓰기(overwrite)의 수를 나타내고, AVGWIO는 현재 타임 윈도우에서 연속적으로 덮어 쓰기가 발생한 블록(continuously overwritten block)의 평균 길이(length)를 나타내고, OWSLOPE는 이전 타임 윈도우에서의 평균 덮어 쓰기(overwrite)의 개수 대 현재 타임 슬라이스(time slice)에서의 덮어 쓰기(overwrite)의 개수 간의 비율을 나타낼 수 있다. 그리고 IO는, 이전 타임 슬라이스에서의 평균 쓰기 횟수 대 현재 타임 슬라이스에서의 덮어 쓰기 횟수의 비율을 나타낼 수 있다.

[0057] 240 단계에서, 탐지부(140)는 카운팅된 덮어 쓰기 횟수에 기초하여 랜섬웨어의 활동(activity) 여부를 탐지할 수 있다.

[0058] 이때, 랜섬웨어의 활동 여부를 탐지하기 위해, 다양한 종류의 복수의 랜섬웨어들과 관련하여 각 랜섬웨어들의 활동 기간(active period) 동안 수집된 복수 개의 IO 요청이 상기 6가지 특성 별로 학습될 수 있다. 예컨대, 머신 러닝 알고리즘, 이진 의사 결정 트리(binary decision tree) 알고리즘 등을 이용하여 6가지 특성별로 학습이 수행될 수 있다. 그리고, 탐지부(140)는 학습을 통해 생성된 트리를 기반으로 타임 윈도우 동안 모니터링된 IO 요청을 이용하여 현재 타임 윈도우에서 랜섬웨어의 활동이 존재하는지 여부를 탐지할 수 있다. 여기서, 랜섬 웨어의 활동을 탐지하는 동작은 도 5를 참조하여 아래에서 자세히 설명하기로 한다.

[0059] 랜섬웨어 탐지 시스템(100)은 전체 페이로드(payload) 대신 IO 요청 헤더의 분포만을 확인하여 랜섬웨어의 활동

(activity)을 인식하므로, 경량이면서, 암호화된 랜섬웨어의 덮어 쓰기에 따른 6가지 특성을 기반으로 랜섬웨어를 탐지할 수 있다. 이하에서는 도 3 및 도 4를 참고하여 6가지 특성에 대해 상세히 설명하기로 한다.

[0060] 도 3은 본 발명의 일실시예에 있어서, 랜섬웨어를 특정하는 6가지 특성 중 OWIO관련 그래프이다.

[0061] 도 3을 참고하면, 310은 6가지 특성(feature) 중 랜섬웨어 별 OWIO를 도시한 그래프이고, 320은 랜섬웨어와 일반 어플리케이션의 OWIO를 도시한 그래프이다.

[0062] 그래프 310은, 덮어쓰기(overwrite) 빈도가 다를 때 WannaCry와 Mole 랜섬웨어가 1초의 타임 슬라이스동안 얼마나 오래 활동하는지를 나타낼 수 있다. 310을 참고하면, 덮어 쓰기(overwrite)가 자주 발생할수록(즉, 덮어 쓰기의 빈도수가 증가할수록) WannaCry와 Mole 랜섬웨어의 활동 시간이 길어짐을 확인할 수 있다.

[0063] 그래프 320은 4개의 랜섬웨어(WannaCry, Mole, Jaff, Cryptooshield)와 4개의 일반적인 어플리케이션(즉, 비랜섬웨어 어플리케이션으로서, 데이터 삭제, P2P 다운로드, 클라우드 스토리지 동기화 및 압축)에 대한 덮어 쓰기 요청 수의 누적 그래프를 나타낼 수 있다. 320을 참고하면, 랜섬웨어의 경우, 일반적인 어플리케이션 보다 시간이 지날수록 상대적으로 높은 성장률(growth) 가짐을 확인할 수 있다. 320에서, WannaCry, Mole은 Jaff, Cryptooshield보다 성장률(growth)이 높지만, Jaff, Cryptooshield은 상대적으로 낮은 성장률을 보이므로 랜섬웨어를 특정하는 6가지 특성을 기반으로 랜섬웨어를 탐지하면, Jaff, Cryptooshield를 일반 어플리케이션과 구분하여 탐지 가능할 수 있다.

[0064] OWIO는 짧은 시간 동안 문서 파일의 동일한 블록을 읽고(read), 암호화하며 덮어 쓰는 속성을 나타내는 특성(feature)으로서, 그래프 310을 참고하면, OWIO의 특징값을 변화시키는 1초 동안의 타임 윈도우에서 WannaCry와 Mole 랜섬웨어가 얼마나 오래 활동/행동하는지 확인할 수 있다. 즉, 덮어 쓰기(overwrite)가 자주 발생할수록 랜섬웨어의 활동 시간이 길어짐을 확인할 수 있다. 320을 참고하면, 일반 어플리케이션이 덮어쓰기(overwrite) 빈도는 데이터 와이핑 어플리케이션을 제외하고는 랜섬웨어의 덮어쓰기 빈도만큼 높지 않음(예컨대, 100K 미만)을 알 수 있다. 그래프 320에 도시한 바와 같이, OWIO 특성은 일반 어플리케이션에서의 데이터 와이핑, 클라우드 저장, P2P 다운로드 등으로 인해 발생 가능하며, 누적된 데이터 와이핑 프로그램의 수는 랜섬웨어의 수만큼 높을 수 있다. 이에 따라, 도 4와 같이 OWIO 이외에 랜섬웨어의 덮어 쓰기(overwrite) 활동으로 인한 5가지 특성이 추가적으로 특정될 수 있다.

[0065] 도 4는 본 발명의 일실시예에 있어서, 랜섬웨어를 특정하는 6가지 특성 중 OWIO를 제외한 5가지 특성관련 그래프이다.

[0066] 도 4에서 그래프 410 및 420은 OWST, 그래프 430 및 440은 PWIO, 그래프 450 및 460은 AVGWIO, 그래프 470은 IO, 그래프 480은 OWSLOPE를 도시한 그래프이다.

[0067] 도 3의 그래프 320에서와 같이, 랜섬웨어와 구별하기 어려운 일반 어플리케이션 중 하나가 데이터 와이핑(data wiping)이다. 이때, 중요한 특징은 하나의 블록에 대한 중복 덮어 쓰기가 한번만 계산되는 타임 윈도우 내에서, 쓰기 요청(write request) 중 덮어 쓰기가 몇 번 발생하는가 하는 것이다. 데이터를 안전하게 지우기 위해 일반적인 데이터 와이핑 어플리케이션은 단일 블록에 여러 번 덮어 쓰기를 필요로 하므로 랜섬웨어의 값에 비해 OWST 값이 낮게 발생할 수 있다.

[0068] 예를 들어, 동일한 메모리 블록에 대해 하나의 읽기(read) IO 당 7번의 덮어 쓰기(overwrite)가 필요할 수 있다. 420을 참고하면, OWST가 랜섬웨어 운영 중에 발생하는 쓰기(write) IO에서 높은 덮어 쓰기(overwrite) 속도를 포착함을 확인할 수 있으며, 410을 참고하면, 랜섬웨어의 활동 기간(active period)과 OWST 사이의 강한 상관 관계를 확인할 수 있다.

[0069] 그래프 430 및 440을 참고하면, CPU 사용량이 많은 작업이나 IO 처리량이 많은 작업이 IO 처리량이 많은 작업이 랜섬웨어가 활동 중일 때 실행될 수 있다. 그러면, 랜섬웨어의 속도가 느려지므로, 랜섬웨어의 IO 요청은 오랜 기간동안 분산될 수 있다. 예를 들어, 랜섬웨어 Jaff는 OWIO 및 OWST 특성만으로는 탐지되기 어려울 수 있다. PWIO는 타임 윈도우 동안 발생한 덮어 쓰기(overwrite)의 수를 나타내는 것으로서, 그래프 430을 참고하면, OWIO의 단기 슬라이스(1s) 대신 장기간의 윈도우(10s) 동안 누적된 덮어 쓰기(overwrite) 수를 이용 시 Jaff 등의 랜섬웨어 탐지가 가능할 수 있다. 이때, 그래프 440을 참고하면, 누적된 수와 PWIO의 상관 관계를 알 수 있다.

[0070] 그래프 450 및 460을 참고하면, AVGWIO는 랜섬웨어 공격 대상의 런-길이(run-length) 특성을 탐지/캡쳐하기 위해 이용될 수 있다. 랜섬웨어는 문서 및 이미지를 대상으로 공격하므로, 데이터 와이핑(data wiping), 조각 모

음, DB 업데이트와 달리 복수 개의 연속하는 블록들에 덮어 쓰기(overwrite) 작업을 수행하지 않는다. 이러한 특성을 고려하여 그래프 460을 참고하면, 랜섬웨어의 경우, 일반적인 어플리케이션(예컨대, 데이터 와이핑, MySQL)보다 덮어 쓰기가 수행된 연속하는 블록의 길이가 상대적으로 짧음을 확인할 수 있다. 그래프 450에서, 활동 기간(또는 활동 기간, active period)와 AVGWIO 간의 상관 관계를 확인할 수 있다.

[0071] 6가지 특성 중 OWIO, OWST, PWIO, AVWIO는 랜섬웨어의 활동으로 인해 덮어 쓰기 작업 시 발생하는 주요 특성(feature)에 해당하고, OWSLOPE와 IO는 보조 특성(secondary feature)에 해당할 수 있다. OWSLOPE는 덮어 쓰기(overwrite) 볼륨의 급격한 증가로 인한 랜섬웨어의 활동(즉, 동작)을 탐지하기 위해 이용될 수 있다.

[0072] 그래프 470 및 480을 참고하면, 랜섬웨어의 활동 기간(active period)과 IO 또는 OWSLOPE 간의 상관관계를 확인할 수 있다.

[0073] 6가지 특성 중 어느 하나의 특성만 사용하는 경우, 다양한 종류의 랜섬웨어들 중 특정 랜섬웨어의 탐지가 누락될 수 있으나, 머신 러닝 또는 이진 의사 결정 트리 알고리즘을 이용하여 6가지 특성 별로 학습을 진행하고, 학습된 모델을 기반으로 모니터링된 IO 요청과 관련하여 랜섬웨어의 활동 여부를 탐지함으로써, 즉각적으로 다양한 종류의 랜섬웨어의 탐지가 가능하고, 이전에 알려지지 않은 랜섬웨어의 탐지가 가능할 수 있다. 예를 들어, 이진 의사 결정 트리 알고리즘으로 위의 비특허 문헌 [4] J. R. Quinlan, "Induction of decision trees," Mach. Learn., vol. 1, no. 1, pp. 81-106, 1986.에 제시된 ID3 알고리즘이 이용될 수 있으며, ID3 알고리즘을 통해 구축된 학습 모델인 이진 의사 결정 트리(binary decision tree)를 이용하여 모니터링된 IO 요청에 대한 6가지 특성 별 특징값이 계산될 수 있으며, 특징값을 기반으로 랜섬웨어의 활도 여부가 탐지될 수 있다.

[0074] 도 5는 본 발명의 일실시예에 있어서, 학습된 6가지 특성(feature)을 기반으로 랜섬웨어의 활동(activity)을 탐지하는 세부 동작을 도시한 흐름도이다.

[0075] 도 5의 각 단계들(510 내지 530)은 도 2의 240 단계를 구체화한 것으로서, 탐지부(140)에 의해 수행될 수 있다.

[0076] 모니터링부(110)는 랜섬웨어 검색을 위해 모든 IO 요청을 모니터링할 수 있으며, 각 IO 요청은 시간(time) 정보, 논리 블록 주소(LBA) 정보, IO 모드(IO Mode), 덮어 쓰기된 길이(overwritten length) 정보의 네 가지 항목으로 구분될 수 있다. 즉, IO 요청은 시간(time) 정보, 논리 블록 주소(LBA) 정보, IO 모드(IO Mode), 길이(overwritten length) 정보를 포함할 수 있다.

[0077] 여기서, 시간 정보는 IO 요청이 발생한 시점을 나타내고, 논리 블록 주소(LBA) 정보는 메모리 블록에 기록된 데이터가 읽히거나, 메모리 블록에 데이터가 쓰여지는 시작 주소를 나타낼 수 있다. IO 모드 정보는 IO 요청 유형, 예컨대, 읽기(read) 요청인지, 쓰기(write) 요청인지 여부를 나타내고, 길이 정보(length)는 IO 요청에 해당하는 메모리 블록의 수를 나타낼 수 있다.

[0078] 탐지부(140)는 모니터링된 IO 요청과 카운팅 테이블, 이진 의사 결정 트리를 이용하여 랜섬웨어의 활동(activity)을 탐지할 수 있다.

[0079] 510 단계에서, 탐지부(140)는 랜섬웨어를 특정하는 6개의 특성 각각에 해당하는 특징값을 카운팅 테이블(counting)에 기초하여 계산할 수 있다. 여기서, 카운팅 테이블은 도 6을 참조하여 후술하기로 한다.

[0080] 520 단계에서, 탐지부(140)는 계산된 6가지 특성 별 특징값을 이진 의사 결정 트리의 입력 파라미터로 설정하여, 랜섬웨어의 활동 여부를 나타내는 결과값을 얻을 수 있다. 즉, 이진 의사 결정 트리의 출력이 랜섬웨어의 활동 여부를 나타내는 결과값에 해당할 수 있다. 예컨대, 랜섬웨어가 활동하는 상태(즉, 활성 상태)인 경우, 1의 값이 결과값으로 출력되고, 랜섬웨어가 활동하지 않는 상태인 경우, 0의 값이 상기 트리의 결과값으로 출력될 수 있다.

[0081] 530 단계에서, 탐지부(140)는 결과값을 기반으로 스코어링(scoring)을 수행하여 랜섬웨어가 활성 상태인지 여부를 최종 결정할 수 있다.

[0082] 예를 들어, 탐지부(140)는 미리 지정된 타임 윈도우(예컨대, 10초)동안 획득된 복수의 결과값들을 합산하고, 합산된 결과값이 미리 지정된 임계값(threshold)보다 큰지 여부를 비교할 수 있다. 상기 합산된 결과값이 임계값보다 크면, 탐지부(140)는 랜섬웨어가 활성 상태인 것으로 결정할 수 있다. 즉, 랜섬웨어의 활동을 탐지할 수 있다. 그리고, 합산된 결과값이 임계값 보다 크지 않으면, 탐지부(140)는 랜섬웨어가 활성 상태가 아닌 것으로 결정할 수 있다. 즉, 랜섬웨어가 아닌 일반 어플리케이션의 데이터 와이핑 등으로 인한 덮어쓰기가 발생한 것으로 결정할 수 있다.

[0083] 이처럼, 랜섬웨어의 탐지를 위해 카운팅 테이블이 이용되며, 카운팅 테이블에는 모니터링된 IO 요청 별로 시간 정보, 논리 블록 주소(LBA) 정보, IO 모드 정보, 및 길이(length) 정보가 연관하여 저장될 수 있다. IO 모드 정보와 길이 정보는 통합된 형태인 RL(read length) 및 WL(write length)로 표현될 수 있다.

[0084] 도 6은 본 발명의 일실시예에 있어서, 카운팅 테이블을 도시한 도면이다.

[0085] 도 6을 참고하면, 카운팅 테이블(610)은 시간 정보, 논리 블록 주소(LBA) 정보, 모니터링된 IO 요청에 해당하는 RL(read length) 및 WL(write length) 정보를 포함할 수 있다. 도 6에서, IO 요청은 IOReq로 표현될 수 있으며, length 가 1인 경우를 가정할 수 있다.

[0086] 타임 윈도우는 IO 요청을 통해 모니터링이 수행되는 모니터링 기간을 나타낼 수 있으며, 모니터링 기간 동안 주기적으로 랜섬웨어의 활동 여부가 탐지될 수 있다. 예를 들어, 타임 윈도우가 10초로 설정된 경우, 주기적으로 10초동안 모니터링된 IO 요청을 기반으로 SSD가 랜섬웨어에 감염되었는지 여부를 탐지할 수 있다.

[0087] 타임 윈도우는 N개의 타임 슬라이스로 구성될 수 있으며, 모든 체크 포인트에서 타임 슬라이스(예컨대, 1초)만큼 타임 윈도우가 분할될 수 있다. 6가지 특성(feature)의 가치 평가를 위해 IO 요청(IOReq)의 덮어 쓰기 실행 길이를 저장하는 카운팅 테이블이 저장될 수 있다. 타임 슬라이스동안 발생한 IO 요청의 개수가 카운트될 수 있으며, 카운트된 값과 논리 블록 주소(LBA) 정보에 기초하여 카운팅 테이블이 업데이트될 수 있다.

[0088] 예컨대, 카운팅 테이블(610)을 참고하면 시간(tome) 정보는 새로 생성되거나 업데이트되는 타임 슬라이스 번호(즉, 식별 정보)를 나타내고, LBA는 동일 메모리 블록에 대해 연속하여 덮어 쓰기가 발생한 시작 주소를 나타내고, RL은 LBA에서 연속적으로 발생하는 읽기 요청(read IO)의 총 길이를 나타낼 수 있다. 그리고, WL은 읽기 IO가 발생한 후에 발생하는 쓰기 IO의 총 길이를 나타낼 수 있다. 즉, 읽기 IO가 발생한 블록과 동일한 블록에 연속하여 쓰기가 발생한 블록의 총 길이를 나타낼 수 있다.

[0089] 타임 윈도우 동안 발생한 IO 요청과 관련하여 6가지 특성 별 특징값 계산을 위해 카운팅 테이블(610)과 함께 해시 테이블(620)이 이용될 수 있다. 해시 테이블(hash table, 620)은 논리 블록 주소(LBA)를 해시 함수(hash function)의 키(key)로 적용함에 따라 생성될 수 있다.

[0090] 예를 들어, 6가지 특성(feature) 중 IO의 특징값은 현재 시간 슬라이스 동안 발생한 모든 읽기 IO 요청(RIO)과 쓰기 IO 요청(WIO)의 합으로 계산될 수 있다.

[0091] 그리고, OWIO의 특징값은 현재 슬라이스에 해당하는 WL(write length)의 합계로서 계산될 수 있다. PWIO의 특징값은 현재 시간이 t인 경우, t-11에서 t-1까지의 카운팅 테이블(610)에 저장된 모든 WL의 합계로서 계산될 수 있다. AVGWIO의 특징값은 카운팅 테이블(610)에 저장된 WL 중 t-10에서 t까지의 WL들의 합의 평균(즉, 평균 WL)으로서 계산될 수 있다. OWSLOPE의 특징값은 OWIO를 PWIO로 나눔으로써 계산될 수 있다.

[0092] 아래의 [표 2]은 랜섬웨어 탐지를 위한 알고리즘에 해당할 수 있다.

표 2

Algorithm 1 RansomwareDetection

```
Require: N
1: for all req_i do
2:   if the time slice expires then
3:     Calculate 6 attributes for N time slices
4:     ransom_t =DecisionTree_ID3(6 attributes)
5:     Score = Score + ransom_t
6:     Slide TimeWindow by one time slice
7:     Score = Score − ransom_t−10
8:   end if
9:   if req_i is write-req then
10:    if there is a read entry for req_i in the counting table then
11:      UpdateEntryW()
12:      if not all lba before req_i in the entry are overwritten then
13:        SplitEntry()
14:      end if
15:      if there is an entry overlapped then
16:        MergeEntry()
17:      end if
18:    end if
19:  else
20:    if there is no entry of req_i then
21:      if there is an entry overlapped then
22:        UpdateEntryR() and/or MergeEntry()
23:      else
24:        NewEntry()
25:      end if
26:    end if
27:  end if
28: end for
```

[0093]

[0094] 위의 [표 2] 및 도 7을 참고하면, [표 2] 및 도 7에서 사용된 IO 요청은 IO 모드 정보, LBA 정보, 및 길이(length) 정보를 포함하며, 길이 정보는 1로 가정할 수 있다. T=0에서, 카운팅 테이블(710)과 해시 테이블(720)은 초기에 비어있을 수 있다. 이때, 첫번째 IO 요청(IOReq(1,R))이 수신되면, 탐지부(140)는 해시 테이블(720)을 검색하여 카운팅 테이블(720)에서 수신된/모니터링된 IO 요청(IOReq(1,R))에 해당하는 엔트리(entry)를 찾을 수 있다. 이때, 해시 테이블(720)에서 해당 IO 요청과 관련하여 검색된 키 1이 카운팅 테이블(710)의 엔트리에 없으면, [표 2]의 line 24와 같이, 탐지부(140)는 NewEntry 함수를 사용하여 카운팅 테이블(710)에 새 엔트리(entry)를 만들고, 이를 해시 테이블(720)의 키값 1로 연관하여 등록할 수 있다. 이후 NewEntry 함수는 아래와 같이 업데이트될 수 있다.

[0095] 먼저, 시간(time) 정보는 처음에 0(=현재 타임 슬라이스)으로 설정되고, LBA는 1(=IOReq의 LBA)로 설정되고, RL은 1, WL은 0으로 설정될 수 있다. 즉, 첫 번째 IO 요청은 읽기 IO 요청이므로, 첫 번째 IO 요청이 모니터링된 현재 타임 슬라이스(t=0)에서 RL=1, WL=0으로 카운팅 테이블(710)의 엔트리(RL, WL, t)의 값이 설정될 수 있다.

[0096] 모니터링 중인 현재 타임 슬라이스(t=0) 동안 두 번째 IO 요청(IOReq (2, R))이 수신되면(즉, 모니터링되면), 탐지부(140)는 해시 테이블(720)에서 키 값 2가 있는 엔트리를 검색할 수 있다. 이때, 해시 테이블(720)에 키 값이 2인 엔트리가 존재하지 않으나, LBA 2에 인접한 키 1이 있는 엔트리가 존재할 수 있다. 이 경우, 탐지부(140)는 카운팅 테이블(710)에 새로운 항목을 생성하지는 않지만, 두 번째 IO요청(IOReq (2, R))과 관련하여 업데이트 함수(UpdateEntryR 함수)를 이용하여 인접 엔트리를 업데이트할 수 있다. 즉, [표 2]의 line 22를 참고하면, 두 번째 IO요청(IOReq (2, R))은 첫 번째 IO요청(IOReq (1, R))의 연속적인 요청이므로, 인접 엔트리의 업데이트가 수행될 수 있다. 그러면, UpdateEntryR 함수는 카운팅 테이블(710)에서 해당 RL 값을 1씩 증가시킬 수 있다. 세 번째 IO 요청(IOReq (3, R))이 수신된 경우, [표 2]의 line 22와 같이 동일한 방식으로 카운팅 테이블(710)이 업데이트될 수 있다. [표 2]의 line 6을 참고하면, 체크 포인트 1(t=1)에서 타임 윈도우는 타임 슬라이스에 따라 슬라이드되어 6개의 특성(feature)의 특징값을 계산할 수 있다. 예를 들어, 타임 슬라이스동안 발생한 IO 요청(IOReq)이 3개이기 때문에 IO의 특징값은 3으로 계산되고, 덮어 쓰기(overwrite)가 없기 때문에

OWIO의 특징값은 0으로 계산될 수 있다.

[0097] 타임 슬라이스 t=1에서, 네 번째 IO 요청(IOReq (1, R))이 수신될 수 있다. 이때, IOReq (1, R))는 LBA의 엔트리에 이미 존재하므로 삭제될 수 있다. 즉, 네 번째 IOReq (1, R))은 첫 번째 IO 요청과 동일하여 카운팅 테이블(710)에서 이전 타임 슬라이스(t=0)의 LBA 엔트리에 네 번째 IOReq (1, R))에 해당하는 LBA 정보가 이미 저장되어 있으므로, 탐지부(140)는 네 번째 IOReq (1, R))의 LBA 정보는 카운팅 테이블(710)에 저장하지 않고 삭제할 수 있다. 이처럼, 덮어 쓰기(overwrite) 계산을 위해, 카운팅 테이블(710)에는 타임 윈도우 내에서 읽기 IO 요청과 동일한 LBA 정보를 가진 쓰기 IO 요청만 저장될 수 있다. 즉, 타임 윈도우 동안 발생한 모든 쓰기 IO 요청 중 읽기 IO 요청과 동일한 LBA 주소 정보를 가진 쓰기 IO 요청만 선택적으로 카운팅 테이블(710)에 저장될 수 있다.

[0098] 이어, IO 요청(IOReq (3, W))은 카운팅 테이블(710)에 이미 저장되어 있을 수 있다. 상기 IO 요청(IOReq (3, W))은 LBA 1에서 3까지의 읽기 IO가 있음을 나타내지만, IO 요청(IOReq (3, W))의 LBA는 3에 해당할 수 있다. 즉, LBA 1 및 2는 덮어 쓰기(overwrite)되지 않았지만 LBA 3은 덮어 쓰기(overwrite)되어졌을 수 있다. [표 2]의 line 11을 참고하면, SpliteEntry 함수는 엔트리(entry)를 두 개로 구분할 수 있다. 이때, 하나는(Time 0, LBA 1, RL 2, WL 0) 튜플(tuple)을 가지며, 다른 하나는 (Time 1, LBA 3, RL 1, WL 1) 튜플을 가질 수 있다.

[0099] 체크 포인트 2(즉, t=2)에서 OWIO의 특징값은 1로 계산될 수 있다. 즉, 덮어 쓰기가 한 번 발생했음을 의미할 수 있다. 그리고, WIO의 특징값은 2, OWST=OWIO/WIO이므로 OWST의 특징값은 0.5로 계산될 수 있다. 엔트리(entry)의 개수는 2이고, WL의 총합은 1이므로, [표 2]의 line 3과 같이 AVGWIO의 특징값은 1/로 계산될 수 있다. 다음 타임 슬라이스에 대한 IO 요청(IOReqs)은 이전 타임 슬라이스와 동일하게 적용될 수 있다.

[0100] 마지막 타임 슬라이스(t=3s-4s)에서 모니터링된 IO 요청을 처리하는 과정에서 MergeEntry 함수가 이용될 수 있다. [표 2]의 line 11을 참고하면, IOReq (2, W)는 UpdateEntryW 함수에 의해 LBA 1인 항목을 업데이트할 수 있으며, LBA1과 LBA2가 모두 덮어 쓰기(overwrite)되고, WL은 2로 설정될 수 있다. 또한, 인접한 LBA가 있는 엔트리가 존재하므로, [표 2]의 line 16과 같이, MergeEntry 함수는 두 개의 엔트리(711)을 하나로 병합할 수 있다. 결과적으로, 체크 포인트 4(t=3)에서 IO의 특징값은 4이고 WIO의 특징값은 4로 계산될 수 있다. 반면에, 덮어 쓰기가 두 번 발생하므로 OWIO의 특징값은 2이고 OWST의 특징값은 2/4로 계산될 수 있다. 카운팅 테이블(710)에 두 개의 엔트리가 있으므로 AVGWIO의 특징값은 4/2이고, 두 엔트리에서 WL의 합계는 4로서 계산될 수 있다. PWIO의 특징값은 이전 체크 포인트 3의 모든 WL의 합계이므로 2로서 계산되고, 이전 체크 포인트의 PWIO가 현재 OWIO로 나뉘어져 OWSLOPE가 되므로, OWSLOPE의 특징값은 는 1/1로 계산될 수 있다.

[0101] 타임 슬라이스가 만료되면, [표 2]의 line 6과 같이, 카운팅 테이블(710)에서 오래된 엔트리는 타임 윈도우를 이동하여 삭제될 수 있다. [표 2]의 line 7과 같이, 탐지부(140)는 삭제된 엔트리를 빼서 스코어(score)를 조정할 수 있다. 예를 들어, 카운팅 테이블(710)을 이용하여 6개의 특성 별로 특징값이 계산되고, 계산된 특징값은 ID3 알고리즘을 기반으로 생성된 이진 의사 결정 트리(DecisionTreeID3)의 입력 파라미터로 설정될 수 있다. 그러면, 이진 의사 결정 트리의 출력값으로서, 6개의 특성 별 결과값이 획득될 수 있다. 탐지부(140)는 결과값을 합하여 랜섬웨어의 활동 여부를 최종적으로 결정할 수 있다.

[0102] 예를 들어, 상기 이진 의사 결정 트리는 6개의 특성 별 특징값들을 입력으로 하여, 0 또는 1의 결과값을 출력할 수 있다. 1은 현재 체크 포인트에서 랜섬웨어의 공격을 받을 가능성이 높음을 나타내고, 0은 현재 체크 포인트에서 랜섬웨어의 공격을 받을 가능성이 높지 않음을 나타낼 수 있다. 10초 타임 윈도우를 사용하는 경우, 10초 동안 상기 트리의 출력이 모두 합쳐져서 도 8과 같이, 0부터 10까지의 점수(810)가 계산될 수 있다. 이처럼, 타임 윈도우 별로 트리의 출력이 합산된 스코어 점수가 계산되고, 계산된 점수가 미리 정의된 임계값(예컨대, 3)보다 크면, 탐지부(140)는 랜섬웨어가 활동(active, 즉, 활성 상태)하는 것으로 결정할 수 있다. 그리고, 임계값보다 크지 않으면, 랜섬웨어가 활동하지 않는 것, 즉, SSD에 존재하지 않는 것으로 결정할 수 있다.

[0103] 본 발명의 실시예에 따른 방법들은 다양한 컴퓨터 시스템을 통하여 수행될 수 있는 프로그램 명령(instruction) 형태로 구현되어 컴퓨터 판독 가능 매체에 기록될 수 있다.

[0104] 이상에서 설명된 장치는 하드웨어 구성요소, 소프트웨어 구성요소, 및/또는 하드웨어 구성요소 및 소프트웨어 구성요소의 조합으로 구현될 수 있다. 예를 들어, 실시예들에서 설명된 장치 및 구성요소는, 예를 들어, 프로세서, 콘트롤러, ALU(arithmetic logic unit), 디지털 신호 프로세서(digital signal processor), 마이크로컴퓨터, FPGA(field programmable gate array), PLU(programmable logic unit), 마이크로프로세서, 또는 명령

(instruction)을 실행하고 응답할 수 있는 다른 어떠한 장치와 같이, 하나 이상의 범용 컴퓨터 또는 특수 목적 컴퓨터를 이용하여 구현될 수 있다. 처리 장치는 운영 체제(OS) 및 상기 운영 체제 상에서 수행되는 하나 이상의 소프트웨어 애플리케이션을 수행할 수 있다. 또한, 처리 장치는 소프트웨어의 실행에 응답하여, 데이터를 접근, 저장, 조작, 처리 및 생성할 수도 있다. 이해의 편의를 위하여, 처리 장치는 하나가 사용되는 것으로 설명된 경우도 있지만, 해당 기술분야에서 통상의 지식을 가진 자는, 처리 장치가 복수 개의 처리 요소(processing 엔트리) 및/또는 복수 유형의 처리 요소를 포함할 수 있음을 알 수 있다. 예를 들어, 처리 장치는 복수 개의 프로세서 또는 하나의 프로세서 및 하나의 콘트롤러를 포함할 수 있다. 또한, 병렬 프로세서(parallel processor)와 같은, 다른 처리 구성(processing configuration)도 가능하다.

[0105] 소프트웨어는 컴퓨터 프로그램(computer program), 코드(code), 명령(instruction), 또는 이들 중 하나 이상의 조합을 포함할 수 있으며, 원하는 대로 동작하도록 처리 장치를 구성하거나 독립적으로 또는 결합적으로(collectively) 처리 장치를 명령할 수 있다. 소프트웨어는 네트워크로 연결된 컴퓨터 시스템 상에 분산되어서, 분산된 방법으로 저장되거나 실행될 수도 있다. 소프트웨어 및 데이터는 하나 이상의 컴퓨터 판독 가능 기록 매체에 저장될 수 있다.

[0106] 실시예에 따른 방법은 다양한 컴퓨터 수단을 통하여 수행될 수 있는 프로그램 명령 형태로 구현되어 컴퓨터 판독 가능 매체에 기록될 수 있다. 상기 컴퓨터 판독 가능 매체는 프로그램 명령, 데이터 파일, 데이터 구조 등을 단독으로 또는 조합하여 포함할 수 있다. 상기 매체에 기록되는 프로그램 명령은 실시예를 위하여 특별히 설계되고 구성된 것들이거나 컴퓨터 소프트웨어 당업자에게 공지되어 사용 가능한 것일 수도 있다. 컴퓨터 판독 가능 기록 매체의 예에는 하드 디스크, 플로피 디스크 및 자기 테이프와 같은 자기 매체(magnetic media), CD-ROM, DVD와 같은 광기록 매체(optical media), 플롭티컬 디스크(floptical disk)와 같은 자기-광 매체(magneto-optical media), 및 롬(ROM), 램(RAM), 플래시 메모리 등과 같은 프로그램 명령을 저장하고 수행하도록 특별히 구성된 하드웨어 장치가 포함된다.

[0107] 프로그램 명령의 예에는 컴파일러에 의해 만들어지는 것과 같은 기계어 코드뿐만 아니라 인터프리터 등을 사용해서 컴퓨터에 의해서 실행될 수 있는 고급 언어 코드를 포함한다.

[0109] 이상에서, 본 발명의 실시례를 구성하는 모든 구성 요소들이 하나로 결합하거나 결합하여 동작하는 것으로 설명되었다고 해서, 본 발명이 반드시 이러한 실시례에 한정되는 것은 아니다. 즉, 본 발명의 목적 범위 안에서라면, 그 모든 구성 요소들이 하나 이상으로 선택적으로 결합하여 동작할 수도 있다. 또한, 이상에서 기재된 "포함하다", "구성하다" 또는 "가지다" 등의 용어는, 특별히 반대되는 기재가 없는 한, 해당 구성 요소가 내재할 수 있음을 의미하는 것이므로, 다른 구성 요소를 제외하는 것이 아니라 다른 구성 요소를 더 포함할 수 있는 것으로 해석되어야 한다. 기술적이거나 과학적인 용어를 포함한 모든 용어들은, 다르게 정의되지 않는 한, 본 발명이 속하는 기술 분야에서 통상의 지식을 가진 자에 의해 일반적으로 이해되는 것과 동일한 의미가 있다. 사전에 정의된 용어와 같이 일반적으로 사용되는 용어들은 관련 기술의 문맥상의 의미와 일치하는 것으로 해석되어야 하며, 본 발명에서 명백하게 정의하지 않는 한, 이상적이거나 과도하게 형식적인 의미로 해석되지 않는다.

[0110] 이상의 설명은 본 발명의 기술 사상을 예시적으로 설명한 것에 불과한 것으로서, 본 발명이 속하는 기술 분야에서 통상의 지식을 가진 자라면 본 발명의 본질적인 특성에서 벗어나지 않는 범위에서 다양한 수정 및 변형이 가능할 것이다. 따라서, 본 발명에 게시된 실시례들은 본 발명의 기술 사상을 한정하기 위한 것이 아니라 설명하기 위한 것이고, 이러한 실시례에 의하여 본 발명의 기술 사상의 범위가 한정되는 것은 아니다. 본 발명의 보호 범위는 아래의 청구범위에 의하여 해석되어야 하며, 그와 동등한 범위 내에 있는 모든 기술 사상은 본 발명의 권리범위에 포함되는 것으로 해석되어야 할 것이다.

부호의 설명

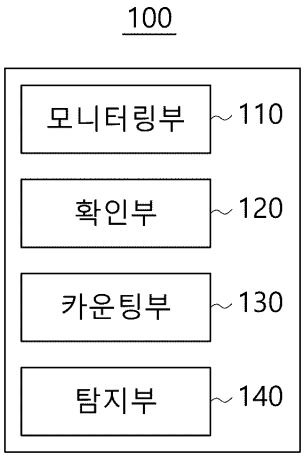
[0112] 100 : 랜섬웨어 탐지 시스템

110 : 모니터링부 120 : 확인부

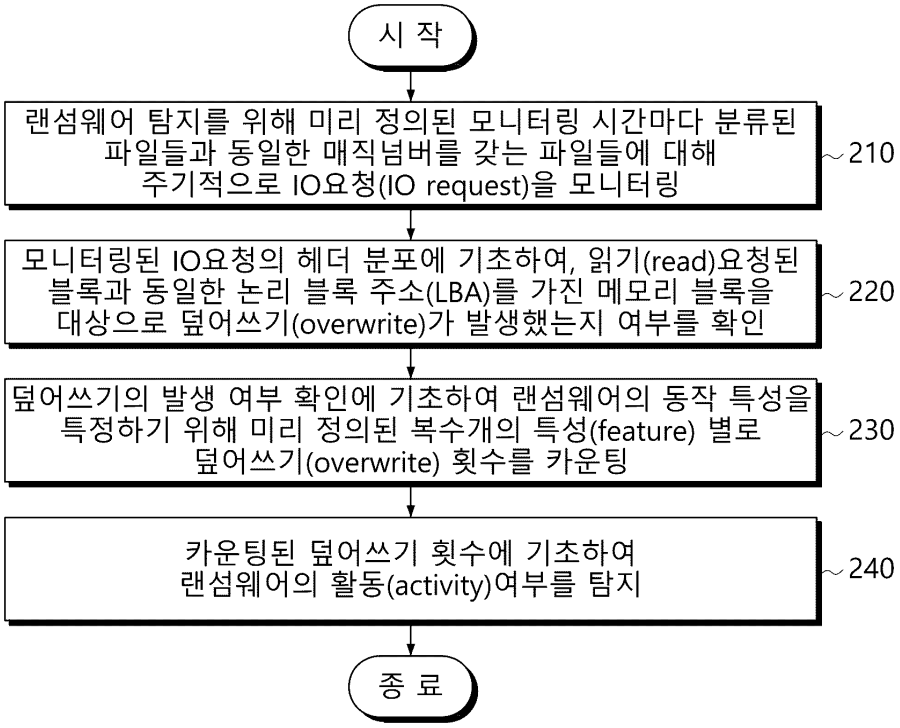
130 : 카운팅부 140 : 탐지부

도면

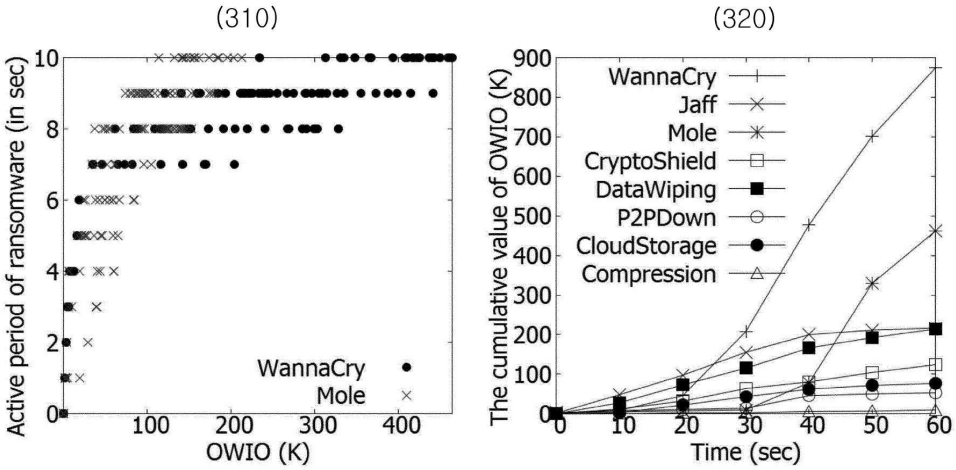
도면1


100
모니터링부
110
확인부
120
카운팅부
130
탐지부
140

도면2


시 작
랜섬웨어 탐지를 위해 미리 정의된 모니터링 시간마다 분류된 파일들과 동일한 매직넘버를 갖는 파일들에 대해 주기적으로 IO요청(IO request)을 모니터링
210
모니터링된 IO요청의 헤더 분포에 기초하여, 읽기(read)요청된 블록과 동일한 논리 블록 주소(LBA)를 가진 메모리 블록을 대상으로 덮어쓰기(overwrite)가 발생했는지 여부를 확인
220
덮어쓰기의 발생 여부 확인에 기초하여 랜섬웨어의 동작 특성을 특정하기 위해 미리 정의된 복수개의 특성(feature) 별로 덮어쓰기(overwrite) 횟수를 카운팅
230
카운팅된 덮어쓰기 횟수에 기초하여 랜섬웨어의 활동(activity)여부를 탐지
240
종 료

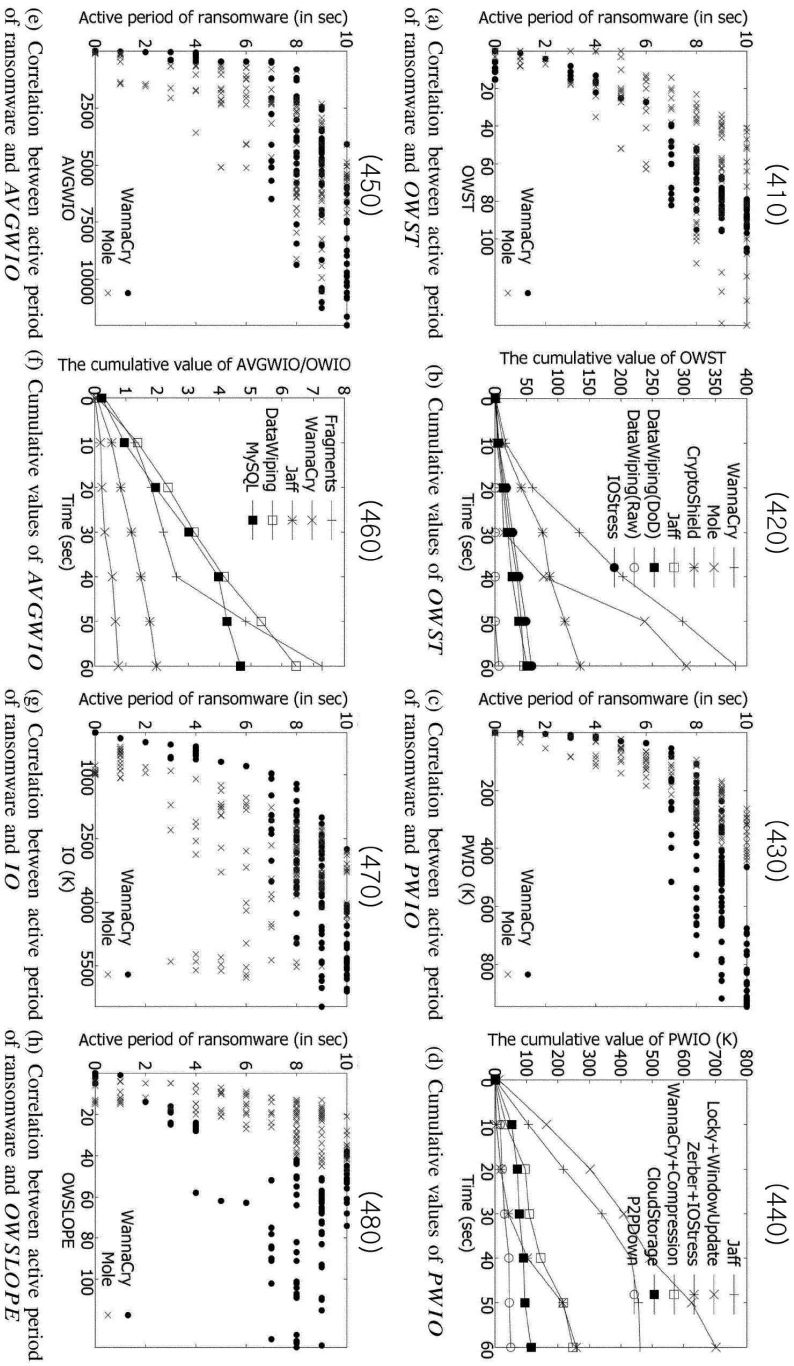
도면3



(a) Ransomware action vs. overwriting frequency.

(b) Ransomware and normal app's cumulative overwriting frequencies

도면4



(a) Correlation between active period of ransomware and $OWST$

(b) Cumulative values of $OWST$

(c) Correlation between active period of ransomware and $PWIO$

(d) Cumulative values of $PWIO$

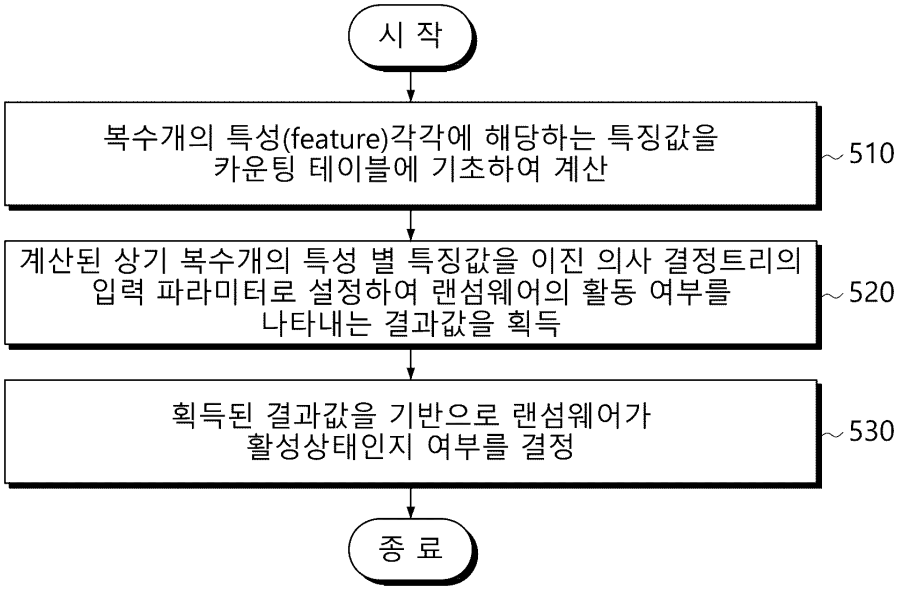
(e) Correlation between active period of ransomware and $AVGWIO$

(f) Cumulative values of $AVGWIO$

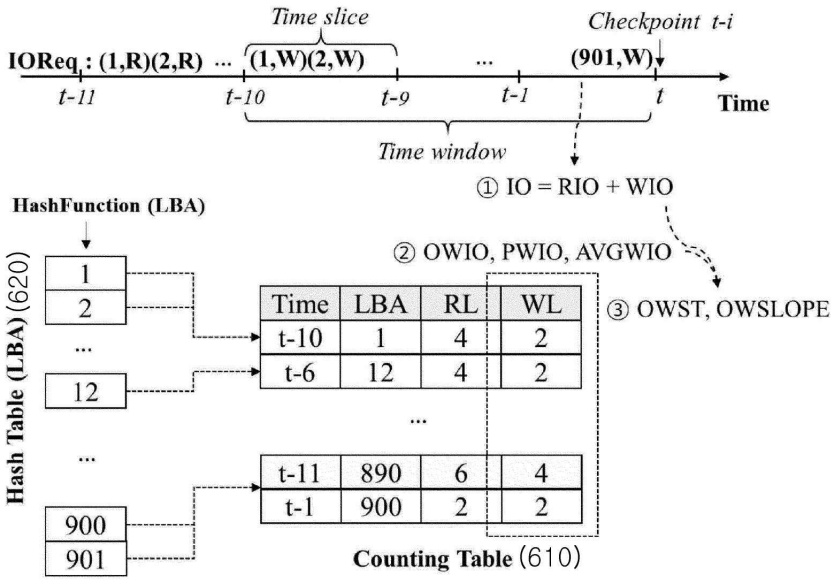
(g) Correlation between active period of ransomware and IO

(h) Correlation between active period of ransomware and $OWSLOPE$

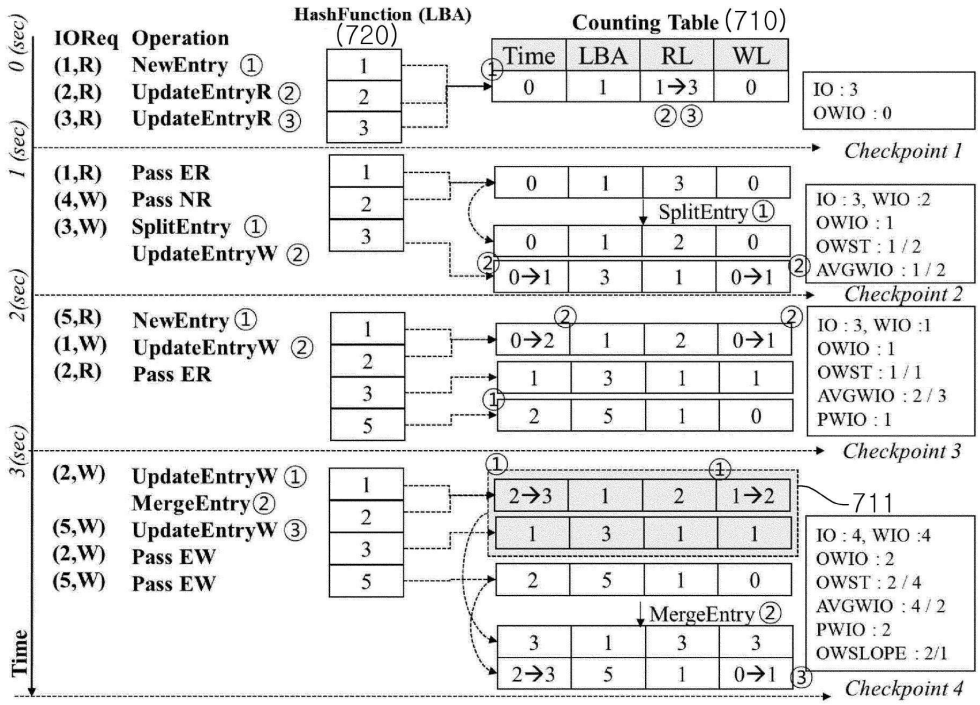
도면5


시 작
복수개의 특성(feature)각각에 해당하는 특징값을 카운팅 테이블에 기초하여 계산
510
계산된 상기 복수개의 특성 별 특징값을 이진 의사 결정트리의 입력 파라미터로 설정하여 랜섬웨어의 활동 여부를 나타내는 결과값을 획득
520
획득된 결과값을 기반으로 랜섬웨어가 활성상태인지 여부를 결정
530
종 료

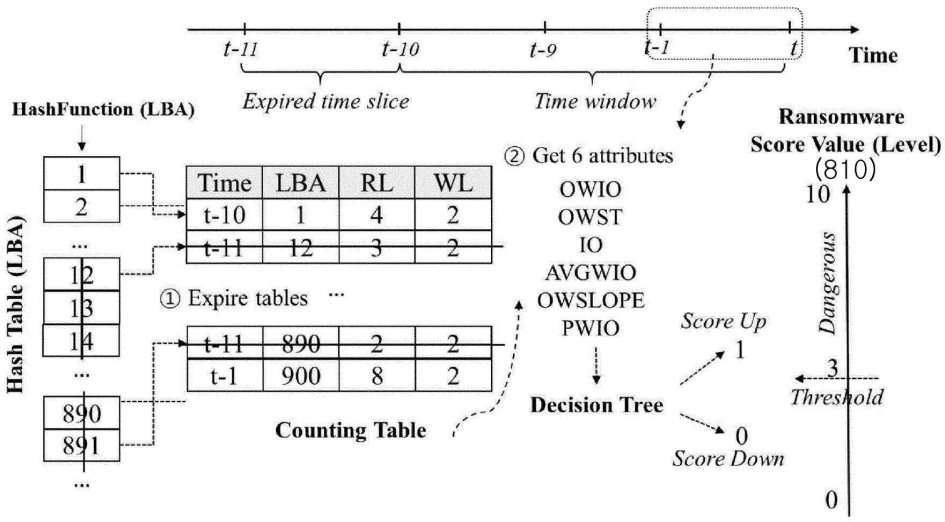
도면6


Time slice
Checkpoint t-i
IOReq : (1,R)(2,R) ... (1,W)(2,W) ... (901,W)
t-11
t-10
t-9
t-1
t
Time
Time window
① IO = RIO + WIO
② OWIO, PWIO, AVGWIO
③ OWST, OWSLOPE
HashFunction (LBA)
Hash Table (LBA) (620)
1
2
12
900
901
Time LBA RL WL
t-10 1 4 2
t-6 12 4 2
t-11 890 6 4
t-1 900 2 2
Counting Table (610)

도면7


HashFunction (LBA)
(720)
Counting Table (710)
IOReq Operation
(1,R) NewEntry ①
(2,R) UpdateEntryR ②
(3,R) UpdateEntryR ③
(1,R) Pass ER
(4,W) Pass NR
(3,W) SplitEntry ①
UpdateEntryW ②
(5,R) NewEntry ①
(1,W) UpdateEntryW ②
(2,R) Pass ER
(2,W) UpdateEntryW ①
MergeEntry②
(5,W) UpdateEntryW ③
(2,W) Pass EW
(5,W) Pass EW
Time
Time LBA RL WL
IO : 3
OWIO : 0
Checkpoint 1
SplitEntry ①
IO : 3, WIO :2
OWIO : 1
OWST : 1 / 2
AVGWIO : 1 / 2
Checkpoint 2
IO : 3, WIO :1
OWIO : 1
OWST : 1 / 1
AVGWIO : 2 / 3
PWIO : 1
Checkpoint 3
711
MergeEntry ②
IO : 4, WIO :4
OWIO : 2
OWST : 2 / 4
AVGWIO : 4 / 2
PWIO : 2
OWSLOPE : 2/1
Checkpoint 4

도면8


t-11
t-10
t-9
t-1
t
Time
Expired time slice
Time window
HashFunction (LBA)
Hash Table (LBA)
Time LBA RL WL
t-10 1 4 2
t-11 12 3 2
① Expire tables
t-11 890 2 2
t-1 900 8 2
Counting Table
② Get 6 attributes
OWIO
OWST
IO
AVGWIO
OWSLOPE
PWIO
Decision Tree
Score Up
1
0
Score Down
Ransomware
Score Value (Level)
(810)
10
Dangerous
3
Threshold
0