블룸 필터를 이용한 SSD-Insider 알고리즘에 대한 성능 향상

김정현*, 정창훈*, 양대헌*, 이경희**

*인하대학교 컴퓨터공학과, **수원대학교 전기공학과

Enhancing SSD-Insider Algorithm with Bloom Filter

Jeong-Hyeon Kim*, Chang-Hun Jung*,
Dae-Hun Nyang*, Kyung-Hee Lee**

*Computer Science Engineering, Inha University,
**Electrical Engineering, The University of Suwon

요 약

랜섬웨어[1]란 사용자 데스크탑에 존재하는 파일들을 암호화하여 복호화 비용을 요구하는 악성 프로그램이다. SSD-Insider[2]는 SSD 내부에서 최근 10초간의 블록 I/O를 관찰하여 랜섬웨어를 탐지할 수 있는 알고리즘이며, 본 논문에서는 SSD-Insider 알고리즘에 사용되는 해시테이블을 블룸 필터[3]로 변경하여 성능을 향상시킨 알고리즘을 제안한다. 결과적으로 동일한 전체 데이터셋에 대해 평균적으로 메모리 사용량을 65% 감소시키면서 동일한 정확도를 얻었으며, 또한 데이터셋에 대해 SSD-Insider 알고리즘이 요구하는 최대 메모리 사용량과 동일한 메모리양을 사용하여 10배 긴 시간을 관찰할 수 있었다. 그리고 이를 이용해 탐지하기 어려웠던 랜섬웨어에 대해서도 정상적으로 탐지하는 결과를 보였다.

I. 서론

랜섬웨어[1]란 사용자 데스크탑에 존재하는 파일들을 암호화하여 복호화 비용을 요구하는 악성 프로그램이다. 랜섬웨어 공격의 빈도는 점점 증가하는 추세이며, 개인용 PC 및 서버뿐만 아니라 IoT 디바이스 및 다양한 컴퓨팅 시스템에 대한 랜섬웨어 공격이 확산되고 있다. 따라서 철저한 관리를 통해 랜섬웨어를 예방하고 탐지하며 암호화된 파일을 복구하는 시스템이 필요하다.

SSD-Insider[2]는 SSD 내부에서 최근 10초간의 블록 I/O를 관찰하여 랜섬웨어를 탐지할 수 있는 알고리즘을 개발하였다. SSD-Insider 알고리즘은 SSD 내부에 구현되어야 하기 때문에, 메모리 사용량을 최대로 줄이는 것이 중요하다. 또한 최근에는 다양한 랜섬웨어가 개발됨에 따라 10초보다 더 긴 시간동안의 블록 I/O를 관찰하여, 다양한 특징을 가진 랜섬웨어를 탐지 할 수 있어야 한다.

본 논문에서는 SSD-Insider에서 사용되는 자료구조 중 해시테이블을 블룸 필터[3]로 대체하는 알고리즘을 제안한다. 결과적으로 동일한 탐지 정확도를 유지하면서, 평균적으로 메모리 사용량을 65% 감소시킬 수 있다. 또한 데이터셋에 대하여 SSD-Insider 알고리즘이 요구하는 최대 메모리 사용량과 동일한 메모리양으로 10배 긴 시간을 관찰할 수 있으며, 이를 이용해 탐지하기 어려웠던 랜섬웨어를 탐지할 수 있다.

II. SSD-Insider

SSD-Insider[1]에서는 랜섬웨어가 사용자의 파일을 암호화하기 위해, 특정 LBA를 읽은 후 동일한 LBA에 쓰는 행위를 덮어쓰기라고 정의하였다. 그리고 최근 10초 이내 덮어쓰기의 발생 빈도를 관찰하고 ID3 알고리즘을 이용하여 랜섬웨어를 탐지하는 알고리즘을 개발하였다. SSD-Insider 알고리즘에서는 덮어쓰기 판단을 하기 위하여 Read LBA를 10초 동안 해시테이블에 유지한다. 그리고 Write LBA 요청이 발생하면, 해시테이블을 탐색하고 동일한 Read LB

A가 존재할 시에 덮어쓰기라고 판단한다. 1초 동안에 발생한 모든 Read LBA는 1개의 해시테이블에 저장되고, 따라서 10초 동안의 Read LBA를 저장하기 위하여 10개의 해시테이블을 사용한다. 본 논문에서는 이것을 윈도우 사이즈 10의 해시테이블이라고 정의한다.

SSD에는 FTL, GC 등을 위해 DRAM이 존재하며, 제조사가 추가적인 알고리즘을 탑재하기 위해서는 더 많은 DRAM이 요구된다. 이는 SSD 단가상승의 원인이 되어 사용자의 부담으로 이어질 수 있다. 본 논문에서는 랜섬웨어 탐지 알고리즘을 SSD에 탑재할 때, 사용자의 금전적인 부담을 최소화하기 위하여 SSD-Insider의 메모리 사용량을 줄이는 알고리즘을 제안한다.

III. 블룸 필터를 이용한 SSD-Insider

블룸 필터는 원소를 추가하거나 검사할 때 비트 단위로 수행하기 때문에 메모리를 절약할 수 있고, 메모리 사용량에 비해 충돌 가능성을 줄일 수 있다. 따라서 본 논문에서는 해시테이블을 블룸 필터로 대체한 알고리즘을 제안한다. 블룸 필터를 이용한 알고리즘은 SSD-Insider 알고리즘에서 사용하는 10개의 해시테이블을 10개의 블룸 필터로 교체한다. 각 블룸 필터는 m KB 크기로, (8 x 1024 x m)개의 비트들을 가지고 있는 블룸 필터이다. 각 블룸 필터는 원소 추가와 원소 검사 시 서로 다른 k개의 해시 함수를 사용한다.

Read LBA가 발생하면 원소 추가를 수행하기 위해 k개의 해시 함수를 사용하여 나온 해시 값들에 대응하는 비트들을 모두 1로 설정한다. Write LBA가 발생하면 원소 검사를 수행하기 위해 각 블룸 필터마다 k개의 해시 함수로 나온 해시 값들에 대응하는 비트들이 모두 1인지, 10개의 블룸 필터에 대해 모두 검사한다. 원소 검사 시 각 블룸 필터에서 k개의 비트들 중 단 한 개의 비트라도 1이 아니라면, 해당 LBA는 블룸 필터에 추가되지 않은 원소라고 판단한다.

〔표 1〕 데이터셋 리스트

항 목	이 름	개 수
일반 응용 프로그램	file compress	5
	dropbox	12
	install	10
랜섬웨어	globeImposter	3
	shield	4
	usps	1
	wannacry	4

IV. 실험

4.1 실험 환경 및 방법

실험은 SSD-Insider 논문에서 사용한 데이터셋을 이용하여 진행하였다. 데이터셋은 Linux 시스템에서 가상환경으로 윈도우를 구성하고 윈도우에서 랜섬웨어 및 일반 응용 프로그램을 실행시킨 다음, Linux에서 Blktrace로 가상 윈도우 환경에서 발생하는 Read/Write 요청을 수집하였다. 데이터셋의 포맷은 csv이며, 시간, Read/Write, LBA, 크기로 이루어져 있다. 수집하여 실험에 사용한 랜섬웨어 및 일반 응용 프로그램의 데이터셋은 [표 1]과 같다.

블룸 필터의 파라미터로는 블룸 필터의 크기 m과 사용할 해시 함수의 개수 k가 있다. m과 k를 정하기 위해서는, 블룸 필터에 추가될 원소의 개수 n과 충돌 확률 p가 필요하다. 10개의 블룸 필터 각각에 추가될 원소 개수 n의 최대치를 확인하기 위해, 데이터셋 전체에 대해 1초간 발생하는 Read LBA의 최대 개수를 측정하였다. 측정 결과 1초간 최대 279,639개의 Read LBA가 존재하였고, 따라서 추가될 원소의 개수 n은 279,639로 설정하였다. 충돌 확률 p는 원소 개수 n과 비슷한 1/300,000의 확률로 설정하였고, 이에 따라 m은 1 MB로, k는 10으로 설정하였다.

실험은 SSD-Insider 논문의 기준인 윈도우 사이즈 10일 때와, 이보다 긴 시간 동안 블록 I/O를 관찰하기 위하여 윈도우 사이즈를 100으로 늘렸을 때를 기준으로 진행하였고, 각각에 대하여 메모리 사용량과 수행시간을 확인하였다. 그리고 윈도우 사이즈 10일 때 탐지하기 어려웠던 gryphon 랜섬웨어에 대해 윈도우 사이

〔표 2〕 윈도우 사이즈 10 일 때의 평균 메모리 사용량 비교

일반 App. / RW	Algo. with HT	Algo. with BF
file compress	50 MB	10 MB
dropbox	24 MB	
install	39 MB	
wannacry	26 MB	10 MB
usps	23 MB	
globeImposter	33 MB	
shield	8 MB	
평균	29 MB	10 MB

〔표 3〕 윈도우 사이즈 10 일 때의 평균 수행시간 비교

일반 App. / RW	Algo. with HT	Algo. with BF
file compress	32.91 sec.	38.82 sec.
dropbox	16.59 sec.	17.42 sec.
install	8.91 sec.	15.2 sec.
wannacry	11.96 sec.	13.09 sec.
usps	19.26 sec.	24.52 sec.
globeImposter	9.51 sec.	12.57 sec.
shield	5.24 sec.	6.89 sec.
평균	14.91 sec.	18.35 sec.

즈를 100으로 늘렸을 때 탐지할 수 있는지와 그에 따른 메모리 사용량을 확인하였다.

4.2 윈도우 사이즈 10의 실험 결과

[표 2]는 윈도우 사이즈가 10일 때 데이터셋에 대하여 SSD-Insider 알고리즘과 논문에서 제안하는 알고리즘의 메모리 사용량을 비교한 결과이다. 실험 결과, shield 랜섬웨어를 제외한 대부분의 데이터셋들은 많은 Read를 발생시키기 때문에 저장해야할 LBA가 많아서 상당량의 메모리를 요구한다. 결과적으로 데이터셋에 대하여 블룸 필터를 사용하는 알고리즘이 SSD-Insider 알고리즘보다 전체 데이터셋의 평균 메모리 사용량에 대해 약 65%, 각 데이터셋의 평균 메모리 사용량에 대해 최대 80% 적은 메모리양을 사용하였다. 그리고 file compress 데이터셋에 대해서 최대 100 MB를 사용하는 경우도 존재하였고, 이 경우 90% 적은 메모리를 사용하여 동일한 정확도를 얻는 것을 확인할 수 있었다.

[표 3]은 윈도우 사이즈가 10일 때 데이터셋에 대하여 SSD-Insider 알고리즘과 논문에서 제안하는 알고리즘의 수행시간을 비교한 결과이다. 블룸 필터는 원소 추가와 원소 검사 시 상당량의 해시 연산을 요구하기 때문에, 해시테이블을 사용한 SSD-Insider 알고리즘보다 수행시간이 느려짐을 볼 수 있었다. 실험 결과 평균적으로 1.2배 느려짐을 확인하였다.

4.3 윈도우 사이즈 100의 실험 결과

[표 4]는 윈도우 사이즈가 10일 때보다 더 긴 시간동안 블록 I/O를 관찰하기 위하여, 윈도우 사이즈를 10배 증가시켜서 메모리 사용량을 측정한 결과이다. 실험 결과, dropbox, install 그리고 shield 랜섬웨어를 제외한 데이터셋들에 대하여, 본 논문에서 제안하는 알고리즘이 상대적으로 적은 메모리양을 사용하는 것을 확인할 수 있었다. SSD-Insider 알고리즘은 100초를 관찰하기 위해 전체 데이터셋의 평균 메모리 사용량에 대해 약 14%, 각 데이터셋의 평균 메모리 사용량에 대해 최대 55% 적은 메모리양을 사용한 것을 확인할 수 있었다.

[표 5]는 윈도우 사이즈가 100일 때 데이터셋에 대하여 SSD-Insider 알고리즘과 논문에서 제안하는 알고리즘의 수행시간을 비교한 결과이다. 실험 결과, 윈도우 사이즈 10일 때와 마찬가지로, 원소 추가와 검사 시 k개의 해시 함수를 수행해야하기 때문에 SSD-Insider 알고리즘보다 느린 수행시간을 보이며, 평균적으로 1.38배 느린 수행시간을 보였다.

4.4 gryphon 랜섬웨어 실험

이전에 탐지하기 어려웠던 gryphon 랜섬웨어에 대해서는 윈도우 사이즈 10의 SSD-Insider 알고리즘을 이용하면 8개의 데이터셋 중 모두를 탐지하지 못했다. 그리고 윈도우 사이즈를 100으로 증가시켰을 때에는 8개 중에 6개에 대해서는 탐지를 하지만, 메모리 사용량은 최대 424 MB를 사용하였다.

본 논문에서 제안하는 블룸 필터를 이용하는

〔표 4〕 윈도우 사이즈 100 일 때의 평균 메모리 사용량 비교

일반 App. / RW	Algo. with HT	Algo. with BF
file compress	226 MB	100 MB
dropbox	46 MB	
install	59 MB	
wannacry	139 MB	100 MB
usps	138 MB	
globeImposter	155 MB	
shield	49 MB	
평균	116 MB	100 MB

〔표 5〕윈도우 사이즈 100일 때의 평균 수행시간 비교

일반 App. / RW	Algo. with HT	Algo. with BF
file compress	166.89 sec.	280.85 sec.
dropbox	35.82 sec.	87.84 sec.
install	8.19 sec.	28.70 sec.
wannacry	207.15 sec.	254.70 sec.
usps	1220.34 sec.	1613.23 sec.
globeImposter	230.53 sec.	318.22 sec.
shield	99.72 sec.	135.21 sec.
평균	281.23 sec.	388.32 sec.

알고리즘을 사용했을 때에는 윈도우 사이즈를 100으로 설정하여 실험하였을 때, gryphon 랜섬웨어 8개 중 6개에 대해서 정상적으로 탐지하여 SSD-Insider 알고리즘과 동일한 정확도를 보였다. 하지만 메모리 사용량에 대해서는 424 MB와 비교하여, 약 76% 감소한 100 MB를 사용하였다.

V. 결론

본 논문에서는 SSD-Insider 알고리즘에서 해시테이블을 블룸 필터로 교체하여 구현한 알고리즘을 제안하였다. 결과적으로 블룸 필터를 사용하는 알고리즘이 윈도우 사이즈 10으로 실험을 진행하였을 때, SSD-Insider 알고리즘보다 평균 메모리 사용량이 65% 감소하였다. 또한 SSD-Insider 알고리즘을 사용하였을 때 file compress 데이터셋에서 최대 100 MB의 메모리를 요구하였는데, 블룸 필터를 이용한 알고리즘은 동일한 100 MB의 메모리양으로 10배 긴 시간을 관찰할 수 있었다. 그리고 이를 이용해 탐지하기 어려웠던 gryphon 랜섬웨어를 약 76% 적은 메모리 사용량으로 탐지할 수 있었다. 하지만 각각의 데이터셋에 따라 메모리 사용량에 대해 손해 본 경우와 수행시간에 대해 손해 본 경우가 존재하였다.

향후 연구로는 블룸 필터를 이용하는 알고리즘을 모든 데이터셋에 대해 메모리와 수행시간에서 이득을 보기 위해서 모든 블룸 필터에 동일한 해시 함수를 사용하고, 정수 해싱을 활용하는 방식을 이용하여 최적화 시킬 계획이며, Open-Channel SSD에서 구현하여 실효성을 검증할 것이다. 이후에는 탐지 윈도우 사이즈를 미리 정하여 탐지하는 알고리즘과 다르게, I/O 요청이 적은 경우 몇 시간 단위로도 관찰할 수 있는 새로운 알고리즘을 연구하려고 한다.

[Acknowledgements]

이 논문은 2018년도 정부(과학기술정보통신부)의 재원으로 정보통신기획평가원의 지원을 받아 수행된 연구임(No. 2018-0-00391, I/O 분포를 이용한 행위 기반의 랜섬웨어 탐지 기술).

[참고문헌]

[1] 조성준, 강승용, 노봉남. (2018). 랜섬웨어 유형별 특징분석 및 위협에 대한 연구. 한국정보기술학회 종합학술발표논문집, (), 472-475.

[2] Baek, SungHa, et al. "SSD-insider: Internal defense of solid-state drive against ransomware with perfect data recovery." 2018 IEEE 38th International Conference on Distributed Computing Systems (ICDCS). IEEE, 2018.

[3] Bloom, Burton H. "Space/time trade-offs in hash coding with allowable errors." Communications of the ACM 13.7 (1970): 422-426.