# 리눅스 운영체제에서 자원 관리 효율성 극대화를 위한 보안 프로세스 Multi-State 제어 기술

김성진†○, 이병용†, 장은태†, 최상훈†, 박기웅‡
세종대학교 시스템보안 연구실
{sys.ryan0902,yibyeongyong,euntaejang,csh0052}@gmail.com†, woongbak@sejong.ac.kr‡

# Security Process Multi-State Orchestration Technology for Maximizing the Efficiency of Resource Management in Linux Operating System

Kim Seong-Jin, Yi Byeong-Yong, Eun-Tae Jang, SangHoon-Choi, Ki-Woong Park
Department of Information Security, Sejong University, Seoul, South Korea

**요 약**

컴퓨터에서는 사용할 수 있는 컴퓨팅 자원은 물리적으로 제한적이기 때문에, 컴퓨팅 성능을 극대화하기 위해서는 이 자원을 가능한 한 효율적으로 사용해야 한다. 악성코드를 실시간 감시하는 보안 프로세스, 리눅스 데몬 등과 같이 항상 실행 상태를 유지하기 위해 메모리를 계속 점유(Always-on)하고 있는 프로세스는 단순 대기 상태일 때 많은 메모리를 사용할 필요가 없다는 특징을 가지고 있다. 이러한 프로세스에 대한 메모리 할당량을 필요에 따라 동적으로 조절함으로써 컴퓨팅 자원을 더욱 효율적으로 사용할 수 있다.

본 논문에서는 Always-on 방식의 보안 프로세스들의 자원을 효율적으로 제어/관리하기 위해 컨테이너 기술을 통해 보안 프로세스를 가상화하여 구동하고, 보안 프로세스가 필요로 하는 메모리 공간의 변화에 따라 Hot, Warm, Cold 상태로 구분하여 프로세스 상태에 따른 적절한 메모리 할당량을 동적으로 제어하는 기술 Thermometer를 제안한다. 우리의 실험결과에 따르면 1시간 동안 3개의 보안 프로세스(Clam AV, Snort, Kaspersky)들의 메모리 누적 사용량을 측정해본 결과 본 논문에서 제안한 프로세스 Multi-State 관리 기술을 활용할 경우 Always-on 방식과 비교하여 약 44%의 메모리 자원을 절약할 수 있다는 것을 확인하였다.

## 1. 서 론

컴퓨터에서는 사용할 수 있는 컴퓨팅 자원(메모리)이 물리적으로 제한적이기 때문에, 이 자원을 효율적으로 사용하는 것은 최적의 컴퓨팅 성능을 얻기 위한 매우 중요한 요소이다. 컴퓨터가 동작할 때 메모리를 점유하여 사용하는 주체는 프로세스이고, 프로세스는 다음과 같은 유형들이 있을 수 있다. (1) 정의된 기능을 수행하기 위해 메모리를 점유했다가 기능 수행을 마친 후 메모리를 반환하고 종료되는 프로세스. (2) 항상 실행 상태를 유지하기 위해 메모리를 계속 점유(Always-on)하고 있는 프로세스. 예를 들면, 텍스트 에디터, 웹 브라우저, terminal 등은 주로 (1) 유형의 프로세스에 해당하고, 악성코드를 실시간 감시하는 보안 프로세스, 리눅스 데몬 등은 대체로 (2) 유형의 프로세스에 해당할 것이다. 특히, (2) 유형의 프로세스는 단순히 대기하고 있을 때와 특정 작업을 수행할 때의 메모리 사용량이 크게 차이나고, 단순 대기 상태일 때는 많은 메모리를 사용할 필요가 없다는 특징을 가지고 있다. 따라서 이러한 유형의 프로세스들에 대한 메모리 할당량을 필요에 따라 적절히 조절한다면 - 대기 상태일 때는 더 적[1]은 메모리를 할당하고, 특정 작업을 수행하기 위해 더 많은 메모리 공간이 필요할 때는 더 많은 메모리를 할당한다면 - 컴퓨팅 자원을 더욱 효율적으로 사용할 수 있을 것이고 이로 인해 사용자에게 더 높은 컴퓨팅 성능을 제공할 수 있을 것이다.

본 논문에서는 컴퓨팅 자원(메모리)의 효율성 극대화를 실현하기 위해 위에서 언급한 (2) 유형에 해당하는 프로세스의 메모리 할당량을 필요에 따라 동적으로 조직하여 프로세스의 상태를 관리할 수 있는 Thermometer를 제안한다. Thermometer는 리눅스 운영체제 환경에서 사용자의 행위분석을 통해 보안 프로세스들의 자원을 효율적으로 관리해주는 메커니즘이다.

본 논문의 구성은 다음과 같다. 2장에서는 Always-on 방식의 프로세스의 문제점을 제기한 연구들에 대해 정리한다. 3장에서는 Always-on 방식의 보안 프로세스들의 자원을 효율적으로 관리하기 위해 설계한 Thermometer

1) †공동 제1저자: (가나다순), ‡교신저자: 박기웅 (세종대학교 정보보호학과 교수)

* 본 연구는 2019년도 과학기술정보통신부의 재원으로 정보통신기획평가원의 지원(No.2018-0-00420) 및 한국연구재단 연구과제(NRF-2020R1A2C4002737)의 지원을 받아 수행된 연구임.

에 대해 설명한다. 마지막 4장에서는 결론과 향후연구에 대해 기술한다.

## 2. 관련 연구

메모리 공간은 물리적으로 제한된 자원이기 때문에, 컴퓨팅 성능을 개선하기 위해서는 이 자원을 가능한 한 효율적으로 사용해야 한다. 메모리를 사용하는 주체는 프로세스인데, 특히 보안 프로세스는 악성코드로부터 컴퓨터를 보호하기 위해 항상 실행 상태를 유지하며 메모리를 점유하고 있는 Always-on 방식의 프로세스이다. 이러한 특성 때문에, 보안 프로세스와 관련하여 더욱 효율적인 메모리 자원 사용을 위한 여러 가지 연구들이 진행됐다.

Ali Hamzah는 시스템 자원의 효율성 관점에서 바라본다면, 보안 프로세스는 메모리를 지속적으로 점유하고 있는 프로세스이므로, 필요에 따라 한정된 컴퓨팅 자원인 메모리를 최적으로 할당하는 것이 보안 프로세스가 상주하고 있는 인프라에 있어서 매우 중요한 문제라고 지적한다 [1].

“Cloud antivirus cost model using machine learning”의 저자 Hamzah은 클라우드 기반의 안티바이러스 소프트웨어의 자원할당 문제를 해결하기 위해 안티바이러스의 총 파일 크기, 코드 세그먼트의 크기, 실행 파일 내의 임베디드 파일 수 등에 따라 안티바이러스 소프트웨어의 비용 모델을 연구하고 머신러닝을 적용한 의사 결정 트리 분류기를 사용하여 최적화된 클라우드 기반의 안티바이러스 소프트웨어에 대한 비용 모델을 제안하였다 [2]. 스마트 폰과 같은 모바일 장치에서 항상 백그라운드에서 동작(Always-on)하는 안티바이러스는 상당한 자원을 소모하기 때문에 배터리 수명에 가장 큰 영향을 미치며 [3], “Selectively -Taming Background Android Apps to Improve Battery Lifetime”의 저자 Martins과 Marcelo는 백그라운드 작업을 하는 안드로이드 앱에서 배터리 소모를 줄일 수 있는 운영체제 메커니즘 Tamer를 제안하였다 [4]. 기존의 여러 연구를 통해 알 수 있듯이, 보안 프로세스에 대한 메모리 자원 할당량 관리는 자원 효율성 측면에 있어서 매우 중요한 문제이다.

## 3. 효율적인 자원 제어를 위한 Thermometer 설계

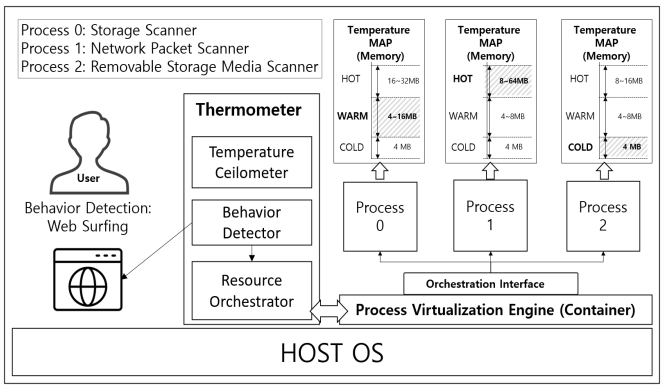


그림 1. 보안 프로세스 자원 제어를 위한 Thermometer 구조

우리는 Always-on 방식의 보안 프로세스들의 자원을 효율적으로 제어/관리하기 위해 Thermometer를 디자인하였다. 우리가 제안한 Thermometer의 구조는 그림 1과 같다. Thermometer는 리눅스 운영체제 환경에서 사용자의 행위분석을 통해 보안 프로세스들을 효율적으로 관리해주는 메커니즘이다. Thermometer의 내부는 크게 3가지(Temperature Ceilometer, Behavior Detector, Resource Orchestrator) 핵심모듈로 구성되어 있다.

첫째, Temperature Ceilometer는 보안 프로세스의 자원 사용량 분석을 통해 3가지의 상태(Cold, Warm, Hot)를 측정해주고 MAP으로 관리한다. Cold 상태는 보안 프로세스가 구동될 수 있는 최소의 자원(메모리 할당량)을 의미한다. 일반적으로 컨테이너 기술을 통해 보안 프로세스를 가상화하여 구동시키면, 최소 4MB의 메모리가 할당되어 있으면 프로세스가 Terminate 되지 않고 Running 상태를 유지한다. Warm 상태는 일반적으로 보안 프로세스가 Idle 상태일 때 소모하고 있는 메모리의 사용량의 40~60%를 의미한다. Warm 상태에서는 실시간으로 악의적인 행위를 탐지할 수 있는 상태이지만, 보안 프로세스를 통해 급격하게 소모되는 자원을 방지할 수 있다. Hot 상태는 보안 프로세스가 Burning 상태일 때 소모하였던 메모리의 최대 사용량을 의미한다. Hot 상태에서는 보안 프로세스가 Burning 상태에서 자원을 제한 없이 쓸 수 있다.

둘째, Behavior Detector는 사용자의 행위분석을 통해 필요한 보안 프로세스의 우선순위를 선정할 수 있도록 도와준다. Behavior Detector는 사용자의 일반적인 행위 정보를 분석하기 위해 구동 중인 프로세스 리스트(Web Brower, Editor, etc)들을 모니터링하며, 특정 이벤트 행위 정보(File Copy, File Drop, Injection, etc)등을 탐지하기 위해 eBPF [5, 6, 7]를 활용하여 시스템콜 정보를 추적 및 분석한다.

마지막으로, Resource Orchestrator는 보안 프로세스의 메모리 사용량을 Scale-Up/Down 해주는 기능을 지원한다. 보안 프로세스를 컨테이너 엔진을 통해 가상화하여 Privilege Mode로 구동시키면, 호스트 운영체제의 사용자 레벨 영역을 모두 모니터링할 수 있다. 또한, 컨테이너 엔진이 통해 구동된 프로세스는 Running 상태에서 메모리 할당량을 동적으로 제어할 수 있다. 본 논문에서는 자원 제어의 범위를 전체 메모리로 한정하여 설명하고 있지만, CPU Core, 블록 I/O, 커널 메모리,스왑 메모리 크기 등의 다양한 자원을 제어할 수 있다.

결과적으로 Thermometer는 Temperature Ceilometer를 통해 리눅스 운영체제 환경에 설치된 보안 프로세스들의 Temperature MAP을 생성하고, Behavior Detector와 Resource Orchestrator를 통해 상황에 맞는 보안 프로세스의 자원 할당량을 동적으로 제어하면서 불필요한 자원 사용을 최소화한다.

## 4. 성능 평가

본 장에서는 보안 프로세스의 자원을 효율적으로 관리하기 위해 설계한 Thermometer의 성능을 평가한다.

### 4.1 실험 환경

본 논문의 실험 환경은 가상화 환경(Virtual Box)에서 진행되었다. 구축된 가상화 환경은 vCPU-Intel Core i7 9700(프로세서 2개)를 사용하였으며, 16GB(DDR4)의 메모리를 할당하였다. 운영체제는 Ubuntu 18.04(64bit)를 사용하였으며, 보안 프로세스를 가상화하기 위한 컨테이너 엔진은 docker 19.03.13을 사용하였다.

### 4.2 메모리 누적 사용량 비교

우리는 Thermometer의 성능을 평가하기 위해 보안 프로세스(Clam AV, Snort, Kaspersky)가 설치된 환경에서 사용자가 웹 서핑을 1시간 진행하였을 때 일반적인 환경(Always-on)과 Thermometer가 적재된 환경에서 소모되는 누적 메모리 사용량(1분 주기 측정)을 비교 측정하였다.

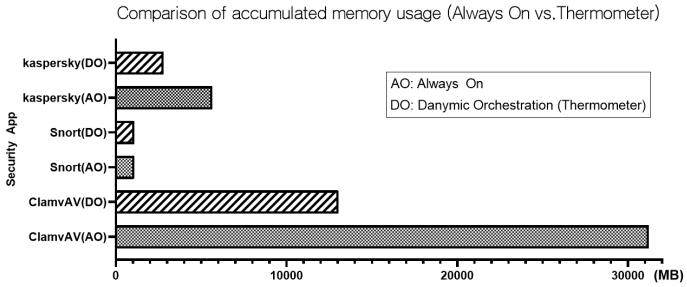


그림 2. 보안 프로세스 관리 기법에 따른 메모리 누적 사용량 비교

실험 결과는 그림 2와 같다. ClamAV, Kaspersky의 경우 스토리지 영역을 스캐닝하여 악성코드를 탐지하는 보안 프로그램이다 [8, 9]. 따라서 사용자의 행위가 웹 서핑일 경우 웹 브라우저를 통해 파일 다운로드 이벤트가 발생할 수 있으므로 파일 스캐닝과 관련 있는 ClamAV, Kaspersky는 Warm 상태 유지한다. Snort의 경우 스니퍼, 패킷 로거, 네트워크 침입 탐지 기능을 제공해준다 [10]. 따라서 웹 서핑 액션에서는 네트워크 관련 보안 프르세스인 Snort는 Hot 상태를 유지하게 된다.
결과적으로 1시간 동안 3개의 보안 프로세스(Clam AV, Snort, Kaspersky)들의 메모리 누적 사용량을 측정해본 결과 본 논문에서 제안한 프로세스 Multi-State 관리 기술을 활용할 경우 AO(Always-on)와 비교하여 약 44%의 메모리 자원을 절약할 수 있다는 것을 확인하였다.

## 5. 결론

본 논문에서는 Always-on 방식 프로세스의 자원을 효율적으로 제어/관리하기 위한 메커니즘을 제안하였다. 우리가 제안한 Thermometer는 리눅스 운영체제 환경에서 사용자의 행위분석을 통해 보안 프로세스들을 효율적으로 관리하여 컴퓨팅 리소스 효율성을 개선하는 기술이다. Thermometer는 컨테이너 기술을 통해 보안 프로세스를 가상화하여 구동하고 보안 프로세스의 상태에 따라 적절한 메모리 할당량을 동적으로 조직한다. 프로세스의 상태는 보안 프로세스가 필요로 하는 메모리 공간의 변화(할당량)에 따라 Hot, Warm, Cold 상태로 구분되며, Thermometer가 상황에 따라 보안 프로세스 컨테이너의 메모리 할당량을 동적으로 제어함으로써 보안 프로세스들에 의해 사용되는 자원 사용량을 최소화한다.
본 논문에서는 알려진 특정 행위 정보와 이벤트 정보를 기반으로 보안 프로세스를 관리한다는 한계점이 있지만, 향후 연구에서는 사용자 행위에 따른 시스템 콜 시퀀스 정보 학습을 통해 사용자의 다양한 행위 정보를 분석하여 프로세스를 인지적으로 관리할 수 있는 연구를 수행할 예정이다.

## 참고문헌

[1] Hamzah, SK Ali Abdullah, Sherif Khattab, and Salwa S. El-Gamal. “Resource allocation for antivirus cloud appliances.“ IOSR Journal of Computer Engineering,(IOSR-JCE) 10 (2013): 2278-0661.

[2] Hamzah, Ali Abdullah, Sherif M. Khattab, and Salwa S. El-Gamal. “Cloud antivirus cost model using machine learning.“ 2014 9th international conference on informatics and systems. IEEE, 2014.

[3] Polakis, Iasonas, et al. “Powerslave: Analyzing the energy consumption of mobile antivirus software.“ International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment. Springer, Cham, 2015.

[4] Martins, Marcelo, Justin Cappos, and Rodrigo Fonseca. “Selectively taming background android apps to improve battery lifetime.“ 2015 {USENIX} Annual Technical Conference ({USENIX}{ATC} 15). 2015.

[5] S. Miano, M. Bertrone, F. Risso, M. Tumolo and M. V. Bernal, “Creating Complex Network Services with eBPF: Experience and Lessons Learned,“ 2018 IEEE 19th International Conference on High Performance Switching and Routing (HPSR), Bucharest, Romania, 2018.

[6] D. Scholz, D. Raumer, P. Emmerich, A. Kurtz, K. Lesiak and G. Carle, “Performance Implications of Packet Filtering with Linux eBPF,“ 2018 30th International Teletraffic Congress (ITC 30), Vienna, 2018.

[7] Deri, Luca, Samuele Sabella, and Simone Mainardi. “Combining System Visibility and Security Using eBPF.“ ITASEC. 2019.

[8] ClamAV, https://www.clamav.net

[9] Kaspersky, https://www.kaspersky.com

[10] Snort, https://www.snort.org