메타버스 안정성 강화를 위한 Introspection 기반 MEC App 보안 모니터링 시스템에 관한 연구

조여름*, 박기웅†
세종대학교 시스템보안연구실(지능형드론 융합전공)*
세종대학교 정보보호학과†
sshh802@gmail.com*, *woongbak@sejong.ac.kr†

A Study on the Introspection based MEC App Security Monitoring System for Metaverse Stability

Jo Yeo Reum*, Park Ki Woong†
SysCore Lab. (Convergence Engineering for Intelligent Drone), Sejong University*
Dept. of Information Security, Sejong University†

요 약

메타버스는 초고속·초저지연 데이터 처리를 보장하기 위해 엣지 클라우드 기반의 Multi-Access Edge Computing(MEC) 기술을 사용한다. 메타버스 서비스가 구동되는 가상화 기반의 MEC 애플리케이션(MEC App)은 보안 위협의 주요 대상이 될 수 있다. 따라서, MEC 플랫폼의 무결성 및 안정성을 보장하기 위해 보안 위협에 대응할 수 있는 모니터링 및 탐지 기술이 요구된다. 하지만, 기존의 에이전트 방식의 모니터링 시스템은 인스턴스 내부에서 구동되기 때문에 서비스 지연을 유발할 수 있다. 또한, 사용자 레벨에서 수집된 데이터는 보안 위협에 대한 유의미한 데이터를 확보하는데 어려움이 있다. 본 논문에서는 Introspection 기술을 MEC app 보안 모니터링 시스템에 적용시켜 메타버스의 초고속·초저지연 성능을 보장하면서, 호스트 레벨에서 모니터링을 수행할 수 있는 시스템의 구조와 구현 요구사항을 제시한다.

Ⅰ. 서 론

Covid-19 의 장기화는 사회 기반 인프라가 디지털로 전환되는 디지털 트랜스포메이션을 가속화시켰다. 이러한 과정에서 메타버스가 5G 기술과 함께 급속도로 성장하고 있다. 메타버스의 대용량 데이터의 초고속·초저지연 전송을 구현하기 위해서는 클라우드 및 가상화 기술 기반의 MEC 이 필수적이다. <그림 1>의 MEC 참조구조의 MEC 시스템에서 메타버스 서비스는 MEC 호스트의 가상화 인프라에서 구동되는 MEC app 에서 실행된다 [1]. 그러나, 이와 같은 구조는 공격 표면을 증가시켜 공격의 주요 대상이 될 수 있다. 예를 들어, 공급망 공격으로 손상된 MEC app 으로 많은 수의 서비스 사용자를 악성코드에 감염될 수 있다. 따라서, 이러한 MEC app 의 잠재적인 보안 위협을 모니터링하고 탐지하는 기술은 메타버스 서비스의 안정성을 강화시키기 위해 필수적이다 [2].

기존의 가상화 인프라의 보안 기술은 가상머신 및 컨테이너와 같은 인스턴스 내부에 에이전트를 설치하여 모니터링하는 기술이 주로 제안되어 왔다 [3]. 그러나, 이 방식은 에이전트가 서비스와 동시에 실행되기 때문에 서비스의 성능 저하를 유발하고 인스턴스마다 개별적인 에이전트는 하나의 통합된 가시성을 제공하지 못한다. 또한, 권한이 낮은 사용자 공간에서 모니터링을 하기 때문에 유의미한 데이터를 확보하는데 어려움이 있다.

이러한 접근 방식을 대신하여 가상화 인프라의 호스트 레벨에서 인스턴스를 모니터링할 수 있는 Introspection 기술이 제안되어 왔다 [4-8]. Introspection 은 인스턴스의 런타임 상태에 대한 정보를 실시간으로 모니터링하여 실행 코드를 분석하고 검사하는데 사용되는 기술이다.

본 논문에서는 먼저 Introspection 기술의 특징과 활용된 연구를 살펴보고 이러한 Introspection 기술을 MEC app 보안 모니터링 시스템에 적용시켜 초고속·초저지연 성능을 보장하면서, 호스트 레벨에서 모니터링을 수행할 수 있는 시스템의 구조와 구현 요구사항을 제시한다.

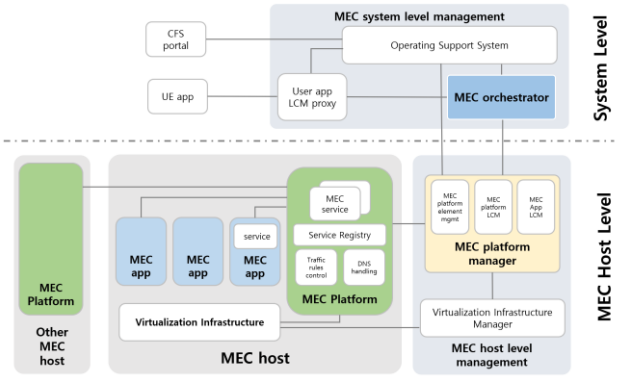

그림 1. MEC 참조 구조[1]

Ⅱ. 가상화 환경의 Introspection 기술

가상화 환경에서의 Introspection 은 호스트의 가상화 인프라에서 구동되는 인스턴스의 런타임 정보를 인스턴스 외부에서 모니터링하는 기술이다. 수집되는 데이터는 실행 중인 애플리케이션을 직접적으로 인스펙션(Inspection)할 수 있는 메모리 및 CPU state 와 같은 로우(raw) 데이터로 전체 시스템에 대한 통찰과 투명성을 제공하며 분석 목적에 따라 다양하게 활용될 수 있다.

Introspection 기술이 가지는 확장성과 활용성으로 인해 보안 분야에서 가상 시스템의 안정성 강화를 위한 연구로 활발하게 활용되고 있다. <표 1>은 Introspection 기술을 활용하여 가상머신과 컨테이너 내부의 보안 위협에 대응하기 위해 수행된 연구이다.

표 1. 가상화 환경의 Introspection 기술 활용 연구

관련 연구		연구 내용
가상머신	Hyperlink[4]	커널 소스 없이 가상머신을 Introspection 하고 메모리 포렌식을 적용하여 루트킷 탐지
	VMI IDS[5]	가상머신 상태 및 이벤트를 Introspection 및 분석하여 침입 탐지 시스템 설계
	Ether[6]	시스템 내부의 시스템 콜을 모니터링하여 malware 를 탐지하는 프레임워크 제안
컨테이너	Docker Container[7]	실행 중인 컨테이너를 Introspection 하는 도구로 감염된 컨테이너 식별
	Container[8]	컨테이너를 외부에서 유연하게 Introspection 할 수 있는 프레임워크제안

Ⅲ. Introspection 기반 MEC app 모니터링 시스템

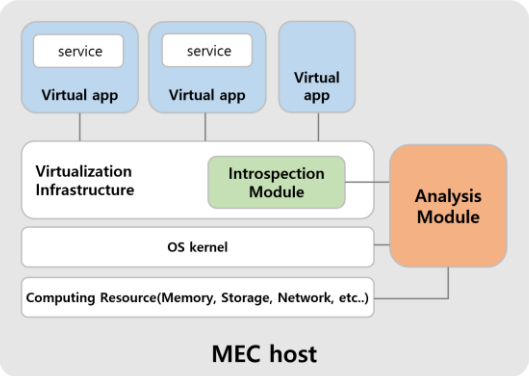

그림 2. MEC App 보안 모니터링 시스템 구조

본 논문에서는 Introspection 기반 MEC app 보안 모니터링 시스템의 전체적인 구조를 <그림 2>와 같이 설계하였다. 기존 MEC 참조 구조를 기반으로 MEC 호스트 내에 두개의 모듈을 추가하였다. Introspection 모듈은 가상화 인프라에서 구동 중인 인스턴스 상태를 모니터링한다. 이 모듈은 인스턴스 보다 더 높은 권한의 호스트 레벨에서 모니터링하기 때문에 인스턴스 내부 실행 상태를 투명하게 관찰할 수 있다. 인스턴스의 Introspection 방법은 메모리, I/O , 시스템 호출(syscall) 및 프로세스 Introspection 등이 있으며 분석 모듈의 접근 방식이나 목적에 따라 달라질 수 있다. 분석 모듈은 모니터링과 동시에 인스턴스 내의 보안 위협을 탐지하는 동작을 수행한다. 이때 Introspection 데이터의 동적 분석을 통해 침입 탐지, 인스턴스 상태 변경 감지, 악성코드 감지 등의 보안 기능을 적용할 수 있다. 시스템의 동작의 예시는 다음과 같다. 인스턴스 내의 프로세스 생성 및 전환을 모니터링하여 숨은 프로세스를 탐지하고 의심스러운 프로세스의 함수 호출을 추적하여 프로세스 동작을 분석할 수 있다.

메타버스 서비스의 안정성을 위한 Introspection 기반 MEC app 을 모니터링 시스템은 다음과 같은 요구사항이 따른다: 1) 메타버스의 초저지연·초고속 특성을 보장하기 위해 Introspection 과정에서 메타버스 서비스의 성능 저하 영향을 최소화해야 한다. 2) 수집 데이터는 가상머신 또는 컨테이너의 환경에 맞게 식별할 수 있는 데이터로 번역되어야 한다. 3) Introspection 모듈 자체 코드의 보안성과 수집된 데이터의 무결성이 보장되어야 한다 [9].

Ⅳ. 결론

본 논문에서는 가상화 환경의 Introspection 기술의 특징과 활용된 연구를 살펴보았다. 또한, Introspection 기술을 MEC app 보안 모니터링 시스템에 적용시켜 메타버스의 초고속·초저지연 성능을 보장하면서, 호스트 레벨에서 모니터링을 수행할 수 있는 시스템 구조와 구현 요구사항을 제시하였다. 향후 이와 같은 모니터링 시스템 연구가 활발히 이루지고 상용 메타버스 서비스에 적용이 된다면 보안 위협 발생을 탐지하고 신속하게 사고 원인을 파악하여 대응할 수 있을 것이라고 기대한다.

ACKNOWLEDGMENT

이 논문은 2021년도 정보통신방송혁신인재양성(ITRC)의 지원 (IITP-2021-0-01816,50%)과2019년도 정부(과학기술정보통신부)의 재원으로 정보통신기획평가원(IITP)의 지원 (No.2019-0-00426, 10%)과 2020년도 한국연구재단(NRF) 연구과제의 지원 (NRF-2020R1A2C4002737, 40%)을 받아 수행된 연구임.

참 고 문 헌

[1] ETSI, "Multi-access Edge Computing (MEC); Framework and Reference Architecture", ETSI GS MEC 003 v2.2.1, Dec, 2020.

[2] 김영수, et al. "5G 환경에서의 MEC 보안위협 및 대응 기술." 정보과학회지 38.9 (2020): 16-24

[3] M. I. Sharif, W. Lee, W. Cui and A. Lanzi, "Secure in-vm monitoring using hardware virtualization", Proceedings of the 16th ACM Conference on Computer and Communications Security CCS '09, pp. 477-487, 2009

[4] J. Xiao, L. Lu, H. Wang and X. Zhu, "HyperLink: Virtual Machine Introspection and Memory Forensic Analysis without Kernel Source Code," 2016 IEEE International Conference on Autonomic Computing (ICAC), 2016

[5] GARFINKEL, Tal, et al. A virtual machine introspection based architecture for intrusion detection. In: Ndss. 2003

[6] DINABURG, Artem, et al. Ether: malware analysis via hardware virtualization extensions. In: Proceedings of the 15th ACMconference on Computer and communications security. 2008. p. 51-62.

[7] Watts, Thomas, et al. "Insight from a docker container introspection." Hawaii International Conference on System Sciences 2019.

[8] Zhan, D, Tan, K, Ye, L, Yu, H & Liu, H 2021, 'Container Introspection: Using External Management Containers to Monitor Containers in Cloud Computing', Computers, Materials & Continua, vol. 69, no. 3, pp. 3783-3794

[9] More, Asit, and Shashikala Tapaswi. "Virtual machine introspection: towards bridging the semantic gap." Journal of Cloud Computing 3.1 (2014): 1-14.