JKIICE Journal of the Korea Institute of Information and Communication Engineering

한국정보통신학회논문지 Vol. 25, No. 10: 1391~1396, Oct. 2021

제로 트러스트 환경의 실시간 파일 접근 이벤트 수집 방법에 관한 연구

한성화[1] · 이후기[2*]

Real-Time File Access Event Collection Methodology for Zero Trust Environment

Sung-Hwa Han[1] · Hoo-Ki Lee[2*]

[1]Professor, Department of Information Security, Tongmyung University, Busan, 48520 Korea
[2*]Professor, Department of Cyber Security, Konyang University, Nonsansi, Chungcheongnam-do, 32992 Korea

요 약

경계 기반 보안체계는 보안 솔루션의 운영 효율성이 높고 관리가 쉬운 장점이 있으며 외부의 보안 위협을 차단하기에는 적합하다. 그러나 신뢰된 사용자를 전제로 운영되기 때문에, 내부에서 발생하는 보안 위협은 차단하기에는 적합하지 않다. 경계 기반의 보안체계의 이러한 문제점을 해결하고자 제로 트러스트 접근통제 모델이 제안되었다. 제로 트러스트 접근통제 모델에서는 실시간 보안 이벤트 모니터링에 대한 보안 요구사항을 만족해야 한다. 본 연구에서는 실시간 모니터링 기능 중 가장 기본적인 파일 접근에 대한 모니터링 방법을 제안한다. 제안하는 모니터링 방법은 kernel level에서 동작하여 사용자의 파일 우회 접근에 의한 모니터링 회피를 원천적으로 방지할 수 있는 장점이 있다. 다만 본 연구는 모니터링 방법에 집중하고 있어, 이를 접근통제 기능까지 확대하기 위한 추가 연구는 계속되어야 한다.

ABSTRACT

The boundary-based security system has the advantage of high operational efficiency and easy management of security solutions, and is suitable for denying external security threats. However, since it is operated on the premise of a trusted user, it is not suitable to deny security threats that occur from within. A zero trust access control model was proposed to solve this problem of the boundary-based security system. In the zero trust access control model, the security requirements for real-time security event monitoring must be satisfied. In this study, we propose a monitoring method for the most basic file access among real-time monitoring functions. The proposed monitoring method operates at the kernel level and has the advantage of fundamentally preventing monitoring evasion due to the user's file bypass access. However, this study focuses on the monitoring method, so additional research to extend it to the access control function should be continued.

키워드 : 실시간, 보안관제, 보안 정책, 제로 트러스트, 접근통제

Keywords : Real-time, Security monitoring, Security policy, Zero-trust, Access control

Received 9 August 2021, **Revised** 3 September 2021, **Accepted** 13 September 2021

*** Corresponding Author** HOO-KI LEE(E-mail:formojo@gmail.com, Tel:+82-41-730-5650)
Professor, Department of Cyber Security, Konyang University, Nonsansi, Chungcheongnam-do, 32992 Korea

Open Access http://doi.org/10.6109/jkiice.2021.25.10.1391 print ISSN: 2234-4772 online ISSN: 2288-4165

Ⅰ. 서 론

대부분의 엔터프라이즈 환경의 기술적 보안체계는네트워크 보안 기술이나 보안 솔루션을 근간으로 하는 경계 기반의 보안체계를 구축하였다. 인프라 및 시스템, 서비스 레벨의 보안체계는, 보호 대상 자산에 대한 보안 요구사항에 맞는 보안 기술을 적용하고 있다.

이 경계 기반의 보안체계는 내부 조직 구성원에 대한 신뢰를 바탕으로 구축된 것으로, 외부의 보안 위협을 탐지하고 차단하는데 적합한 방식이다. 그러나 최근 보안 위협 동향을 본다면, 외부에서 발생한 보안 위협보다는 내부에서 발생한 보안 위협이 더 많다. 이러한 상황에서 많은 엔터프라이즈 환경에 적용되고 있는 경계 기반의 보안체계는 적합하지 않다.

이러한 보안 환경의 한계점을 극복하기 위한 방법으로 제로 트러스트 접근통제 모델이 제안되었다. 제로 트러스트 접근통제 모델에서는 정보 자원을 보호하기 위하여 사용자 인증 및 접근통제, 권한관리 등의 보안 기능을 요구하고 있다. 특히 zero-day attack에 대응하기 위하여, 보안 이벤트의 실시간 모니터링 및 대응 기능을 강조한다.

그러나 대부분의 엔터프라이즈 환경에서는 보안 솔루션에서 수집하고 있는 제한적인 보안 이벤트만을 수집하는 한계가 있다. 이러한 보안 한계점은 조직 내부에서 발생하는 보안 위협을 탐지하기 매우 어려울 뿐만 아니라, 추적하기에도 적합하지 않다.

본 연구에서는 이러한 보안 취약점을 극복하기 위하여, 정보 서비스에 대한 중요 파일에 대한 접근을 모니터링할 수 있는 아키텍처를 제안하고 이를 검증한다.

Ⅱ. 관련 연구

2.1. 보안 환경의 변화

대부분의 엔터프라이즈 환경에서는, 조직의 정보보호 범위를 설정하고 정보보호 특성에 맞게 보안 관리체계를 수립한다. 보안 관리체계를 구축할 때에는 각 조직의 상황에 맞는 보안 환경을 분석 후 보안 모델을 채택한 다음 이에 맞게 보안 관리체계를 구축한다. 그러나 대부분 보안 관리체계의 효율성 추구하며, 이로 인해 결과적으로 경계 기반의 보안체계의 형태를 갖추게 되었다[1].

엔터프라이즈 환경의 정보서비스에는 물리적으로 제한된 위치에서만 접근할 수 있다. Legacy 엔터프라이즈 환경에서는, 보통 조직에서 제공하는 물리적 장소나, 접근 단말, 접근 시스템의 SW 환경 등이 갖추어진 제한된 업무 환경을 만족해야만 접근이 가능했다.

그러나 이러한 환경에서도 보안 위협은 발생한다. 그림 1 는 최근 발생하고 있는 보안 이벤트를 발생시킨 주체의 내·외부 비율을 나타낸다. 내부 구성원에 의해 발생 된 보안 위협 비율이 증가하고 있으므로, 많은 기관에서 적용하고 있는 경계 기반의 보안 관리체계의 한계점이 지적되고 있다[2].

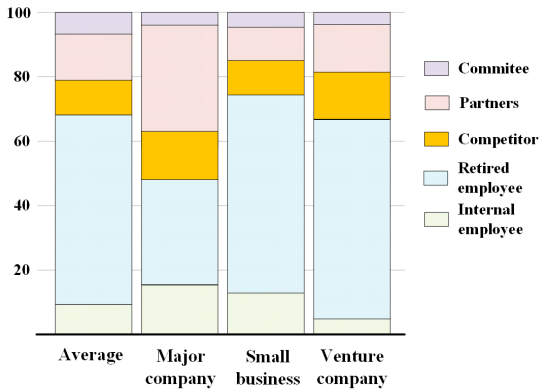

Fig. 1 Security threat occur subject rate

2.2. 제로 트러스트

제로 트러스트는 1994년 Forrester Research의 John Kindervag에 의해 처음 제안된 접근통제 모델이다. 경계 기반의 보안 관리체계의 한계점은 지적되어 왔으나, 이를 개선하기 위한 대안이 없는 보안 환경에서 제로 트러스트 접근통제 모델은 각광받고 있다[3].

제로 트러스트 접근통제 모델에서는 보안 경계를 인정하지 않으며, 물리적인 위치와 상관없이 보호 대상 자원에 접근하는 주체를 기본적으로 공격자로 간주하고 인가되지 않은 모든 접근을 차단한다. 접근통제 모델에서는 파일이나 프로세스뿐만 아니라 시스템, 네트워크 등 다양한 자원이 보호 대상이 될 수 있으며, 주체 또한 사용자 자체나 Application, 시스템 등이 될 수 있다. 보호 대상 자원에는 신원이 검증된 주체의 접근만을 허용한다[4].

많은 공공 기관에서 제안하는 제로 트러스트 접근통제 모델은 다소 차이가 있으나, 공통적인 보안 기능 요구사항은 Table 1과 같다[5, 6].

Table. 1 Zero-trust model's security function requirement

Requirement	Description
MFA	• Multi-Factor Authentication • Subject identification and authentication before grant user privileges.
Authorization	• Grant user privileges to user/process /network subject • All privileges are terminated when subject accomplish its own task.
Access Control	• Access allow/deny subject access to object. • Access control function always enforce, and audit the event.
Security Env. Enhancement	• User security environment enhancement. • If user's security level can't meet the organization's security level, user should to upgrade his security environment via installing security solutions and update user device security configuration.
Audit	• Audit all event for trace security event. • All audit log contain at lease time, subject, object, tools, result.
Real-time Responsibility	• Real-time security event occurs. • When security event is occur, security officer response immediately.

많은 정보보호 전문기관에서는 제로 트러스트 접근통제 모델에 대한 연구를 지속하고 있다. 기관별로 다양한 방식으로 제로 트러스트 모델을 적용하기 위한 전략과 방법론을 제안하고 있으며, 이를 위한 보안 기술 및 솔루션을 출시하고 있다.

2.3. 제로 트러스트 적용을 위한 보안 환경 분석

제로 트러스트 접근통제 모델 기반의 보안 관리체계를 구축하였다고 할지라도, 인증된 사용자에 악의적인 공격행위가 발생할 수 있다.

이러한 사용자에 의한 보안 위협을 차단·방지하기 위하여 제로 트러스트 접근통제 모델에서는 Real-time Responsibility의 보안 요구사항을 제안하였다[7]. 사용자 인증 및 접근통제 기능을 위한 정책이 적용되어 있다고 할지라도, 이미 신뢰 된 사용자는 자의·타의에 의해 다양한 보안 위협을 발생시킬 수 있다. 제로 트러스트의 Real-time Responsibility는, 이러한 상황을 포함하여 다양한 보안 이벤트를 실시간으로 모니터링하고, 필요한 경우 즉시 대응하는 기능이다[8].

Real-time Responsibility를 위해서는 분명 실시간 모니터링 기능이 선행되어야 한다. 많은 보안 위협이 악성코드의 유입으로 시작되는 만큼, 파일 I/O에 대한 실시간 모니터링이 필수적으로 요구된다[9]. 그러나 대부분의 보안 기술 및 보안 솔루션은, 해당 보안 기술 및 보안 솔루션이 제공하는 보안 기능의 메커니즘에 의해 제한적인 파일 접근 이벤트만을 모니터링할 수 있다. 특히 Anti-Virus는 등록된 Signature를 기준으로 동작하게 되므로, Update 되지 않은 경우 일부 악성코드의 동작을 실시간으로 탐지할 수 없다[10].

이러한 상황을 종합한다면, 제로 트러스트 접근통제 모델에서 요구하는 Real-time Responsibility는 만족하기 어려운 상황이다.

Ⅲ. 실시간 파일 접근 모니터링 방법

3.1. 실시간 모니터링 조건

파일 I/O에 대한 실시간 모니터링은, 보안 원칙에 따라, 접근하는 모든 파일의 종류 및 사용자, 접근 Application 종류 등의 접근 조건과 상관없이 모든 파일 I/O에 적용되어야 한다. 또한, 파일 I/O 모니터링 기능을 일체 우회할 수 없어야한다.

이러한 조건을 모두 만족하기 위해서는, 실시간 파일 접근 모니터팅 기능을 운영체제의 kernel level에서 적용해야 한다. 또 사용자의 파일 모니터링 기능 우회를 방지하기 위하여 실시간 파일 접근 모니터링 기능은 단일 지점에 적용해야 한다.

다만 kernel level에서 모니터링 기능을 제공할 때에는, 제로 트러스트 접근통제 모델에서 요구하는 이벤트 기록(Audit)에 대한 보안 요구사항은 만족하기 어렵다. 그러므로 모니터링 기능을 담당하는 커널 모듈과 연동하여 파일에 대한 사용자의 접근 이벤트를 기록할 수 있는 기능은 application level에서 구현해야 한다.

최종적으로 실시간 파일 접근 모니터링 기능에 대한 아키텍처는, kernel level에서 동작하는 module과 application layer에서 동작하는 process를 상호 연동하는 구조가 된다.

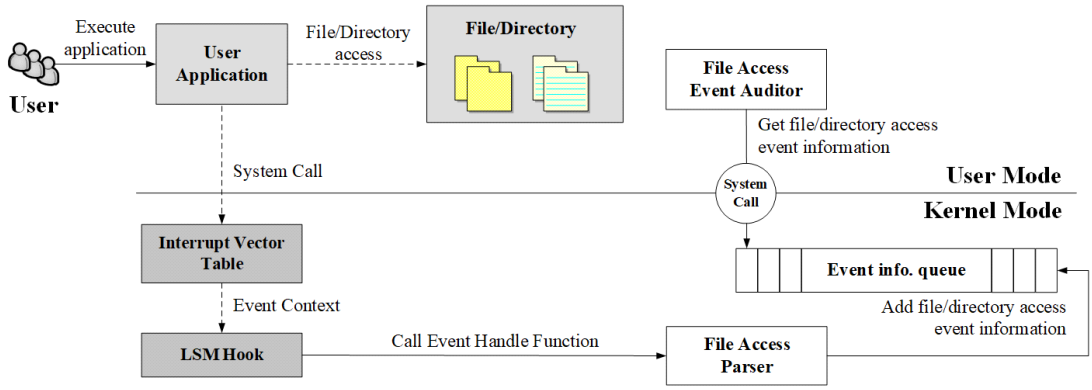

Fig. 2 Zero-Trust based real-time file access monitoring architecture

3.2. 실시간 모니터링 기능 제공 구조

본 연구에서는, 제로 트러스트 환경에서 요구되는 실시간 모니터링 기능 중, 운영체제에서 발생하는 다양한 파일 I/O에 대한 실시간 모니터링 구조를 제안한다.

제안하는 실시간 파일 I/O 모니터링 기능은 Linux 환경을 기준으로 수립하였으며, 그 구조는 그림 2와 같이 크게 2개의 Component로 구성되어 있다.

File Access Parser 는 IRQ(Interrupt Request Queue)로 수집되는 파일 접근 이벤트를 LSM(Linux Security Module)을 통해 수신한다[11, 12]. File Access Parser LSM을 통해 수신한 사용자의 파일 접근 이벤트 정보에서, 파일에 접근하는 주체의 신원을 식별하고, 접근 유형을 식별한다. 또한, 사용자가 접근하려는 파일이 속한 장치 ID와 파일의 절대 경로를 획득하여 Event info. queue에 등록한다. File Access Parser에서 수집하는 정보는 kernel level의 raw data 수준으로, 수집되는 항목은 Table 2와 같다.

Table. 2 Information about file access event

Items	Description
User SID	• File access user SID(Security ID)
Time	• File access event occur time
Device	• Device ID that file registered.
Access Type	• User's access type(RW etc.) • Access scope are different for OS.
Absolute File Path	• File's absolute file path

Event info. queue는, file이나 directory에 대한 접근 이벤트에 대한 정보 일체를 node로 등록하는 자료 구조이다.

File Access Event Auditor는 주기적으로 kernel level의 Event info. queue를 확인한다. 만약 Event info. queue에 새로 등록된 file access event node가 등록된 경우, 이를 application으로 옮기고 Event info. queue에 있는 node를 삭제한다.

File Access Event Auditor는 kernel의 File Access Parser가 전달하는 파일 접근 이벤트에 대한 세부 정보를 수집하여 이를 기록한다. 다만, File Access Event Auditor가 전달하는 SID로는 사용자 계정이나 그룹 계정 정보를 확인할 수 없기 때문에, SID를 바탕으로 사용자 계정 정보나 사용자 그룹 계정에 대한 정보를 추출한다. SID로 사용자 정보를 도출하면, 도출된 정보를 바탕으로 파일 접근 이벤트에 대한 이력을 기록한다.

본 연구에서 제안하는 구조는 kernel level의 단일 지점(IRQ)에 적용하였다. 이 구조적인 특징으로 인하여, 모니터링 기능은 파일에 접근하는 주체의 특성과 상관없이 운영체제에서 발생하는 모든 파일 I/O를 모니터링 할 수 있다. 또한 IRQ는 kernel level의 병목지점이므로, 사용자는 파일 접근에 대한 모니터링 기능을 우회할 수 없다. Application 프로세스와 연동하기 때문에, 파일 I/O에 대한 세부 이력을 기록할 수 있다. 가장 큰 장점으로는 사용자의 파일 접근 유형(Read, Write, Create, Rename, Modify 등)을 식별할 수 있는 장점이 있다.

3.3. 제로 트러스트 기반 실시간 모니터링 동작 절차

사용자 이벤트가 발생한 경우, 발생한 이벤트에 대한

System Call이 호출되며, 이 System Call에 대한 Handler가 IRQ에 등록된다. 만약 사용자 이벤트가 파일 접근인 경우에는 내부적으로 LSM에 등록된 사용자 이벤트 Handler를 먼저 실행한다.

File Access Parser 구동(insmod)으로 LSM에 등록된 사용자 이벤트 Handler가 실행되면, IRQ에 등록된 사용자의 File 접근 이벤트 정보를 확인한다. IRQ에 등록된 파일 접근 이벤트 정보에는 SID와 함께 파일에 대한 절대 경로 등의 정보를 확인한다.

Application Layer에서 동작하는 File Access Event Auditor는 주기적으로 File Access Parser 에 Query하여 사용자의 파일 접근 이벤트 정보를 확인한다. Kernel에서 수집된 정보를 확인하면, 여기의 SID와 Device 장치 ID를 사용자가 인지할 수 있는 User ID와 Group ID, Device Name으로 변환한다. 정보 변환을 완료하면 이를 File에 기록하거나 여타 Process에 전달한다.

Ⅳ. 실시간 모니터링 기능 검증

4.1. 실시간 모니터링 기능 검증 환경

본 연구에서 제안하는 아키텍처의 실효성을 확인하기 위하여, 본 연구에서 제안하는 아키텍처에 따라 실증 구현 후 기능 검증을 수행한다.

기능 검증 환경은 CentOS 7.9에서 구현하였다. 커널 모듈로 File Access Parser를 구현하였으며, Application daemon으로 File Access Event Auditor를 구현하였다. 실시간 모니터링 기능 검증을 위한 테스트 항목은 Table 3과 같다.

Table. 3 Security function verification items

Test ID	Test Object
Func_Test_01	• Check extraction user's file access information
Func_Test_02	• Check covert SID to user ID and device ID to Device Name.

4.2. 실시간 모니터링 기능 검증 결과

Func_Test_01 기능을 검증한 결과, File Access Parser는 그림 3과 같이 사용자의 File I/O 정보를 정확히 수집하는 것을 확인하였다.

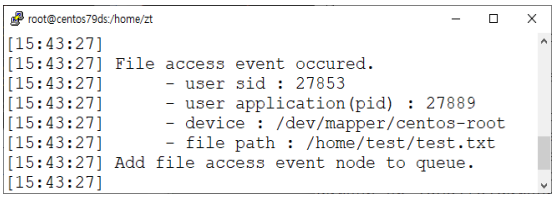

Fig. 3 File access event occur monitoring function verification

Func_Test_02 기능을 검증한 결과, File Access Event Auditor는 그림 4과 같이 File Access Parser가 수집한 Raw data를 사용자가 인지하기 쉬운 형태의 User ID와 Device Name으로 정상 변환됨을 확인하였다.

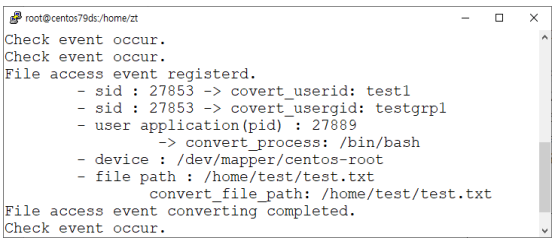

Fig. 4 File access event information converting function verification

Ⅴ. 결 론

엔터프라이즈 정보 기술 환경은 계속 온라인을 통해 확장할 것이며, 이러한 환경의 보안 위협은 계속 증가할 것이다. 특히 COVID 19 이후 원격 재택 업무 방식이 보편화되면서, 엔터프라이즈에 대한 보안 위협은 더욱 증가할 것이다. 이러한 환경에서 제로 트러스트 기반 보안 체계의 구현이나 전환은 사실상 필수적이다.

본 연구에서는 제로 트러스트 기반 보안 체계에서 요구하는 기능 중 가장 기본적인 실시간 파일 I/O 모니터링 기능에 대한 아키텍처를 제안하였다. 본 연구에서 제안하는 아키텍처에 따라 실증 구현 후 기능 검증을 수행한 결과, 본 연구에서 제안한 아키텍처는 충분히 실효성이 있다고 확인되었다.

다만, 본 연구는 Linux 운영체제를 기반의 아키텍처를 제안하였다. 실효성을 수준을 높이고자 한다면, Windows 및 Unix 환경에 대한 아키텍처의 제안 및 실증에 대한 추가적인 연구가 필요하다. 또한, 실시간 모니

터링 기능의 활용도를 높이기 위해서는, 실시간 대응 기능으로 확장되어야 하므로, 이를 위한 연구 확대가 요구된다.

REFERENCES

[1] R. Riccardo and M. Repetto, “Building situational awareness for network threats in fog/edge computing: Emerging paradigms beyond the security perimeter model,” *Future Generation Computer Systems*, vol. 85, pp. 235-249, 2018.

[2] H. B. Chang, “A Study on The Countermeasure by The Types through Case Analysis of Industrial Secret Leakage Accident,” *Convergence security journal*, vol. 15 no. 7, pp. 39-45, 2015.

[3] J. Kindervag, “Build security into your network’s dna: The zero trust network architecture,” *Forrester Research Inc*, pp. 1-26, 2010.

[4] M. Sudakshina, D. A. Khan, and S. Jain, “Cloud-Based Zero Trust Access Control Policy: An Approach to Support Work-From-Home Driven by COVID-19 Pandemic,” *New Generation Computing*, pp. 1-24, 2021.

[5] A. Rastogi and K. E. Nygard, “Software Engineering Principles and Security Vulnerabilities,” in *CATA*, pp. 180-190, Mar. 2019.

[6] X. Hao, W. Ren, R. Xiong, T. Zhu, and K. K. R. Choo, “Asymmetric cryptographic functions based on generative adversarial neural networks for Internet of Things,” *Future Generation Computer Systems*, 2021.

[7] A. Kerman, O. Borchert, S. Rose, and A. Tan, “Implementing A Zero Trust Architecture,” *The MITRE Corporation, Tech. Rep*, 2020.

[8] G. Anil, “A Zero-Trust Security Framework for Granular Insight on Blind Spot and Comprehensive Device Protection in the Enterprise of Internet of Things (E-IOT),” *BMS Institute of Technology*, 2021.

[9] K. D. Uttecht, “Zero Trust (ZT) Concepts for Federal Government Architectures,” *Massachusetts inst of tech lexington United States*, 2020.

[10] M. Al-Asli and T. A. Ghaleb, “Review of signature-based techniques in antivirus products,” *International Conference on Computer and Information Sciences (ICCIS). IEEE,* 2019.

[11] L. Abeni and C. Kiraly, “Investigating the network performance of a real-time Linux Kernel,” *Proc. 15th Real Time Linux Workshop (RTLWS 2013)*, 2013.

[12] C. Wright, C. Cowan, J. Morris, S Smalley, and G. Kroah-Hartman, “Linux security module framework,” in *Ottawa Linux Symposium*, vol. 8032, pp. 6-16, Jun. 2002.

한성화(Sung-Hwa Han)

동명대학교 정보보호학과 교수
숭실대학교 공학박사
SW영향평가 전문위원
※관심분야 : IT융합보안, 시스템 보안, 네트워크 보안, 인공지능, 악성코드 탐지

이후기(Hoo-Ki Lee)

건양대학교 사이버보안학과 교수
숭실대학교 공학박사
창업보육센터장, 정보보호 영재교육원 부원장
※관심분야 : 디지털포렌식, 시스템 보안, 융합 보안, 제로트러스트, 이메일보안