Journal of The Korea Institute of Information Security & Cryptology ISSN 1598-3986(Print)
VOL.34, NO.4, Aug. 2024 ISSN 2288-2715(Online)
https://doi.org/10.13089/JKIISC.2024.34.4.735

소프트웨어 정의 경계를 이용한 네트워크 트래픽 기반 동적 접근 제어*

김 서 이,[1†] 이 일 구[2‡]
[1,2]성신여자대학교 (대학원생, 교수)

Network Traffic-Based Access Control Using Software-Defined Perimeter*

Seo-Yi Kim,[1†] Il-Gu Lee[2‡]
[1,2]Sungshin Women's University (Graduate student, Professor)

요 약

컴퓨터 기술의 급속한 발전은 더 안전한 사용자 환경이 필요하게 되어, 모든 내부 및 외부 네트워크 활동을 검증하는 제로 트러스트 모델의 도입을 촉진했다. 본 논문은 제로 트러스트의 구현 및 지연 문제를 해결하기 위해 소프트웨어 정의 경계 기능을 활용한 효율적인 네트워크 트래픽 데이터 기반 동적 접근 제어 방법을 제안한다. 성능 평가 결과에 따르면 제안한 방법의 탐지 성능은 기존 방식과 유사하게 나타났지만 데이터 셋의 크기는 약 70% 감소했다. 그리고 적응형 제로 트러스트 검증 방식을 제안하여 데이터 셋 크기와 검증 시간을 각각 약 83%, 10% 줄이면서 종래의 방식과 유사한 탐지 성능을 유지했다.

ABSTRACT

The rapid advancement of computer technology has necessitated a safer user environment, prompting the adoption of the zero trust model, which verifies all internal and external network activities. This paper proposes an efficient network traffic data-based dynamic access control method leveraging Software-Defined Perimeter (SDP) capabilities to implement zero trust and address latency issues. According to the performance evaluation results, the detection performance of the proposed scheme is similar to that of conventional schemes, but the dataset size was reduced by 62.47%. Additionally, by proposing an adaptive zero trust verification approach, the dataset size and verification time were reduced by 83.9% and 9.1%, respectively, while maintaining similar detection performance to conventional methods.

Keywords: Zero trust, Software-Defined Perimeter, Dynamic access control

I. 서 론

최근 컴퓨터 기술의 급속히 발전함에 따라, 기존의 텍스트 중심의 사용자 환경에서 그래픽, 이미지, 오디오와 비디오 데이터를 제공하는 멀티미디어 환경으로 변화하고 있다.

제로 트러스트(Zero Trust)란 사이버 보안 모델 중 하나로, "아무도 신뢰하지 않는다"는 원칙에 따라 정보보호체계를 설계, 구축, 운영하는 프레임워크를 의미한다. 제로 트러스트 시스템은 기본적으로 모든 네트워크 활동에 대해 신뢰하지 않으며, 접근을 시도하는 내부와 외부의 모든 요청을 검증하고 인증한다[1]. 제로 트러스트 모델은 기존에 네트워크 경계를 중심으로 보안을 구축하던 경계 보안 모델

Received(06. 12. 2024), Modified(07. 15. 2024), Accepted(07. 17. 2024)
* 본 논문은 2024년도 산업통상자원부 및 한국산업기술진흥원의 산업혁신인재성장지원사업(RS-2024-00415520) 과 과학기술정보통신부 및 정보통신기획평가원의 ICT혁신인재4.0 사업의 연구결과로 수행되었음 (No.IITP-2022-RS-2022-00156310)
† 주저자, cheonrang01@gmail.com
‡ 교신저자, iglee@sungshin.ac.kr(Corresponding author)

(Perimeter Security Model)과 대조적인 구조로 되어 있다.

경계 보안 모델은 네트워크의 내부와 외부 구간을 명확히 설정하고, 경계를 지정한다. 네트워크 내부는 신뢰 구간으로 판단하여 시스템 및 사용자 간의 통신과 관련된 제약 요건이 적고 비교적 자유로운 동작이 가능하다. 네트워크 외부는 비신뢰 구간으로 간주하고 방화벽과 침입 탐지 시스템(Intrusion Detection System) 등의 경계 보안 장치를 통해 동작을 제한한다[2]. 경계 기반 모델은 오랫동안 사이버 보안 모델로써 이용되었지만, 최근 클라우드 컴퓨팅, 모바일 기기, 사물인터넷 등의 사용이 확장되고 원격 근무가 증가함에 따라 네트워크 경계 설정에 어려움을 겪게 되었다. 또한, 사이버 공격이 고도화 및 지능화되어 경계 장비를 우회하는 방법을 활용하여 단일 방어 경계선이 무력화되면 전체 시스템에 위협이 될 수 있다[3]. 최근에는 내부자에 의한 공격 및 위협으로 인한 문제가 화두에 올랐는데, 경계 보안 모델은 내부자를 신뢰 대상으로 판단하기 때문에 공격이 비교적 수월하게 진행될 수 있다. 단일 방어 경계선에 의존하지 않으면서 동적으로 변화하는 네트워크 환경에 유연하게 대처하고, 내부자와 외부자 위협에 대한 고려가 요구됨에 따라 제로 트러스트 모델 도입에 대한 필요성이 제기되었다.

제로 트러스트는 보안성 측면에서 이점을 제공했지만, 인증 과정을 지속적으로 반복하는 과정에서 지연이 발생할 수 있고, 이는 네트워크 성능에 직접적인 영향을 줄 수 있다. 또한 구현 및 운영이 어렵고 시스템이 복잡해지는 문제가 발생한다[4, 5]. 제로 트러스트를 효과적으로 구현하는 동시에 효율적으로 동작할 수 있도록 소프트웨어 정의 경계(Software-Defined Perimeter, SDP)가 주목받게 되었다. SDP는 동적으로 네트워크 경계를 설정하고 비인가자에게 네트워크 리소스를 숨기며, 여러 단계의 인증을 요구하여 제로 트러스트의 원칙을 강화하는 동시에 지연에 따른 성능 저하 문제를 해결하고, 구현 및 관리를 더 효율적으로 수행되도록 한다[6].

본 논문에서는 SDP를 활용하여 네트워크 트래픽 데이터 기반의 저지연 동적 접근 제어 방식을 제안하고자 한다. 제안 방법은 네트워크 트래픽 데이터에 특성 선택(Feature selection)을 적용하여 검증 수행 시 이용하는 특성 수를 줄이고, 머신러닝 알고리즘을 이용하여 빠르고 정확하게 동적 접근 제어를 수행한다. 제안 방법은 단일 및 하이브리드 머신러닝 알고리즘을 적용했을 때, 기존 방식 대비 학습 속도를 평균 64% 개선했다.

그리고 일반적인 상황에 모든 네트워크 트래픽 데이터를 검증하지 않고 적절한 간격을 두어 일부분만 검증하고, 공격이 감지되면 기존의 제로 트러스트 모델과 유사하게 모든 로그를 검증하는 적응형 제로 트러스트 검증 방식을 제안한다. 두 가지 제안 방법의 성능 평가를 통해 제로 트러스트 아키텍처에서 효과적으로 지연을 감소시키는 동시에 탐지 성능을 유지함을 입증한다.

본 논문의 구성은 다음과 같다. 2장에서 제로 트러스트 및 SDP의 개념을 설명하고 이와 관련된 선행 연구를 분석한다. 3장에서는 제안하는 SDP의 특징을 이용한 네트워크 트래픽 기반 접근 제어 방식을 제안한 후, 4장에서 제안 내용의 효과를 입증하기 위한 시뮬레이션 및 결과를 분석한다. 마지막으로 5장에서 결론을 맺는다.

II. 배경지식 및 관련연구

본 절에서는 제로 트러스트 및 SDP에 대한 배경지식을 설명한다. 이후 제로 트러스트 및 SDP와 관련된 선행 연구를 소개한 후, 한계점을 분석한다.

2.1 제로 트러스트

제로 트러스트란 네트워크 내부와 외부의 모든 접근 주체(Entity)에 대한 위협을 항상 동등하게 취급하여 위험을 최소화하는 것을 목표로 하는 보안 모델이다. 제로 트러스트 아키텍처란 제로 트러스트 보안 모델을 실제 환경에 적용하기 위해 이용하는 기술 및 방법론 등을 포함한 구조적 프레임워크를 의미한다. 제로 트러스트 아키텍처는 네트워크 분할과 지속적인 모니터링, 분산 보안 정책, 동적 식별 및 접근 관리 등을 포함한다.

미국 국립표준기술연구소(National Institute of Standards and Technology, NIST)에서는 “NIST SP 800-207”를 통해 제로 트러스트 아키텍처의 기본 원리를 아래의 7가지로 정의했다[7].

1. 모든 데이터 소스 및 컴퓨팅 서비스는 리소스로 간주한다.
2. 네트워크의 위치와 상관없이 모든 통신은 보안이 유지된다.

3. 개별 기업 리소스에 대한 접근은 세션 단위로 부여된다.
4. 자원에 대한 접근은 동적 정책에 의해 결정된다.
5. 기업은 소유 및 연관된 리소스의 무결성과 보안 상태를 모니터링하고 측정한다.
6. 모든 자원 인증 및 권한 부여는 동적이며 접근이 허용되기 전에 엄격하게 시행된다.
7. 기업은 리소스, 네트워크 인프라, 통신 상태에 대한 가능한 많은 정보를 수집하여 보안 상태를 개선하는 데 사용한다.

이와 같은 NIST의 7가지 기본 원리를 통해 알 수 있는 점은 다음과 같다. 제로 트러스트는 경계 기반 보안 모델의 한계를 극복했으며, 지속적인 검증 및 접근 제어를 통해 보안 취약점을 최소화하고 실시간으로 변화하는 네트워크 환경을 반영할 수 있는 새로운 보안 모델을 제공한다. 또한 지속적인 모니터링 및 다양한 데이터 수집을 통해 보안 상태를 유지 및 개선할 수 있으며, 사고 발생 시 빠르게 위협을 탐지하고 대응할 수 있도록 한다.

Fig.1.은 제로 트러스트 아키텍처의 접근 모델을 나타낸다. 제로 트러스트 아키텍처는 제어 영역(Control Plane)과 데이터 영역(Data Plane)의 두 가지 논리 공간으로 구성되어 있다[8]. 제어 영역은 접근 제어 정책이 결정되며, 데이터 영역은 접근 주체가 리소스(Resource)에 접근하는 공간이다. 네트워크의 내부와 외부에 대한 구분 없이 모든 접근 주체를 동등하게 취급한다. 모든 접근 주체는 리소스에 접근을 원할 때, 인증 및 인가 과정을 거치게 된다. 리소스는 데이터, 시스템 등을 포함한다. 접근 제어는 제어 영역의 정책 결정 지점 (Policy Decision Point, PDP)과 데이터 영역의 정책 시행 시점(Policy Enforcement Point, PEP)의 상호작용을 통해 결정된다. 네트워크의 위치와 관계없이 접근이 허용되지 않은 모든 접근 주체를 비신뢰구간(Untrusted)으로 간주하며 접근이 허용되었을 때만 신뢰구간(Trusted)으로 간주한다[9].

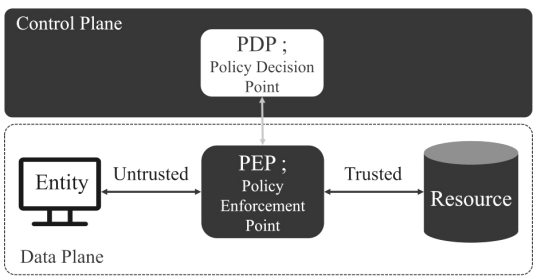

Fig. 1. Zero trust architecture access model

2.2 소프트웨어 정의 경계

소프트웨어 정의 경계는 네트워크 접근을 소프트웨어적으로 제어하여 보안을 강화하는 방법으로 네트워크에 접근하는 모든 주체에 대해 일관된 보안 정책을 적용한다. SDP는 인증이 수행되지 않은 비인가자에 대해 네트워크 리소스를 알 수 없도록 하여 공격자가 해당 네트워크 정보에 대한 수집을 어렵게 한다. 기본적으로 모든 접근 주체에 대해 접근을 거부하고, 인증을 수행한 후에 최소 권한만을 부여하여 불필요한 접근을 차단하고, 세션 설정 시마다 인증이 수행된다. 그리고 세션이 지속되는 동안에도 네트워크의 변동 및 접근 주체의 동작과 위치 조건이 바뀌면 접근 권한을 조정하여 동적 접근 제어를 수행한다. 이러한 이유로 SDP는 모든 접근 주체에 대해 일관되고 수준 높은 보안을 제공하는 동시에 동적으로 변화하는 환경에 맞추어 유연하게 대처할 수 있다.

SDP는 제로 트러스트 아키텍처를 구현할 수 있는 하나의 기술적 방법론으로 이용되고, 제로 트러스트의 기본 원칙을 따르는 동시에 운영 효율성을 향상할 수 있다. 제로 트러스트에 SDP를 적용함으로써 모든 접근 주체의 상태, 위치, 동작 등에 대한 정보를 수집하고 모니터링하며 실시간으로 접근 권한을 부여하고 조정할 수 있는 동적 정책을 시행할 수 있다.

2.3 머신러닝 알고리즘을 이용한 동적 접근 제어

최근에 여러 연구에서 특성 선택 및 머신러닝 알고리즘을 이용한 빠르고 정확한 동적 접근 제어 방식을 제안함에 따라 머신러닝 알고리즘을 이용하여 접근 제어 및 침입 탐지를 수행하는 기존 연구에 대한 분석을 수행했다.

Ananya외 3인[10]은 랜덤 포레스트(Random Forest) 머신러닝 알고리즘 기반의 네트워크 침입 탐지 및 접근 차단 방식을 제안했다. 데이터 특성의 중요도 분석을 수행하여 특성 선택을 수행했고, 원 핫 인코딩(One-hot Encoding) 방식을 이용하여 데이터 품질을 유지하는 기법을 제안했다. 의사 결정 트리(Decision Tree), 다층 퍼셉트론(Multi-layer perceptron) 등의 모델과 비교를 수행했으며, 제안

모델은 가장 좋은 탐지 성능을 보였다. 특히 의사 결정 트리에 비해 약 68% 감소한 평균 제곱 오차(Mean Square Error)를 보였다. 하지만 정확도 측면에서는 큰 개선을 보이지 못했고, 지연 시간을 고려하지 못한 한계점이 있다.

Qigui 외 3인[11]은 주기적으로 사용자 행동 및 속성을 묘사하여 신뢰도를 계산하는 제로 트러스트 아키텍처 기반의 동적 접근 제어 및 권한 부여 시스템을 제안했다. 리소스에 대한 조회, 다운로드, 추가, 수정, 삭제 등의 동작에 대해 각기 다른 임계값을 설정하여 해당 조건을 만족하는 경우에 동적으로 권한을 부여한다. 본 논문은 접근 주체가 접근 시도를 요청한 동작에 따라 보안 수준을 다르게 설정했다는 점에서 이점이 있지만, 제안 방법의 성능을 입증하지 못했고, 효율성 측면을 고려하지 못했다는 점에서 한계가 있다.

Qiuqing 외 1인[12]은 트래픽 자원 스케줄링 관점에서 접근한 제로 트러스트 기반 분산형 동적 접근 제어 방식을 제안했다. 중앙 제어자가 사용자 행동을 지속적으로 모니터링하여 스케줄링 정책을 조정함으로써 내부자 신뢰 프로필을 동적으로 업데이트한다. LSTM 알고리즘을 이용하여 모니터링 과정에서 수집된 데이터의 행동 패턴을 분석한다. 이 연구에서는 트래픽 스케줄링 및 보안성에 대한 평가가 수행되었으나, 지속적인 모니터링 수행 과정에서 지연 시간 및 효율성 측면에 대한 고려가 부족했다.

제로 트러스트 환경은 사용자 및 디바이스의 행동을 지속적으로 모니터링하고 분석해야 할 필요가 있으므로, 대량의 데이터를 지연 없이 처리하고, 비정상적인 행동을 식별해야 한다. 또한 다양한 위협 패턴에 대응하기 위해 규칙 기반의 접근 제어 방식보다 변화하는 환경에 동적으로 대응해야 한다. 보안 위협 및 공격 상황에 대한 실시간 대응 능력을 개선하고, 변화하는 보안 환경에 유연하게 대처하기 위해 머신러닝을 이용한 접근 제어 방식이 필수적이다. 하지만 기존의 머신러닝 기반 동적 접근제어 방식은 지연 시간을 고려하지 못한 한계가 있다. 지연 시간이 길어질 경우, 사용자 인증 및 리소스 접근 요청 등에 많은 시간이 소요된다. 이는 사용자 불만을 초래하고 업무 효율성 감소 및 생산성 저하를 유발할 수 있다. 본 연구에서는 머신러닝을 이용한 동적 접근 시, 지연을 최소화할 수 있는 방법에 대해 논의한다.

III. 네트워크 트래픽 기반 동적 접근 제어

3.1 특성 선택 (Feature selection) 기법

본 논문에서 제안하는 네트워크 트래픽 기반 동적 접근 제어 방식은 제로 트러스트 적용에 따른 지연 문제를 완화할 수 있다. 제로 트러스트는 모든 접근 주체에 대해 지속적인 인증을 수행하기 때문에 기존의 경계 기반 방식에 비해 검증 수행 시간 및 자원 이용량이 증가하게 된다. 인증 과정이 신속하게 수행되지 않으면 지연이 발생하고 이는 네트워크 성능 저하에 치명적인 요인으로 작용할 수 있으므로 빠르고 정확한 인증 및 접근 제어가 필요하다[4]. 본 논문에서 제안하는 네트워크 트래픽 기반 동적 접근 제어 방식은 데이터 전처리 과정에서 특성 선택을 수행하여 검증 데이터의 크기를 감소시키는 동시에 머신러닝 알고리즘을 이용하여 높은 정확도를 유지하는 것을 목표로 한다.

특성 선택은 모델 최적화를 위한 방법의 하나이다. 다음의 세 가지 방법을 활용하여 불필요하거나 중요도가 낮은 데이터를 제거하여 최소한의 핵심 데이터만 이용할 수 있다[13].

1. 필터 방법 (Filter Methods): 데이터의 통계적 특성을 이용하여 중요도를 평가하고 이를 기반으로 특성을 선택하는 방식이다. 이 방법은 분산 임계값, 카이 제곱 검정, 상관계수 분석 방법으로 구현된다.

2. 래퍼 방법 (Wrapper Methods): 다양한 특성 조합을 시도하여 가장 좋은 성능을 보이는 특성 조합을 선택한다. 이 방법은 전진 선택, 후진 제거 방법으로 구현된다.

3. 임베디드 방법 (Embedded Methods): 필터 방법과 래퍼 방법의 장점을 결합한 방식으로 모델 훈련 과정에 특성 선택 과정을 포함한다. 이 방법은 릿지 회귀, 트리 기반 방법, 엘라스틱넷 방법으로 구현된다.

본 연구에서는 최소한의 데이터를 이용하여 빠른 검증을 수행하기 위해 임베디드 방식의 평균 불순도 감소 (Mean Decrease in Impurity, MDI) 방식을 활용하여 효과적인 특성 선택을 통해 데이터를 축소했다. MDI 방식은 트리 기반 알고리즘을 이용하여 수행되며, 모델에서 각 특성이 트리의 불순도를 감소시키는 정도에 따라 중요도를 측정한다. 이 방법은 특성이 모델에 기여하는 정도를 명확히 제시하며 비교적 빠른 계산이 가능한 장점이 있다.

3.2 네트워크 트래픽 기반 저지연 동적 접근 제어

Fig.2.는 특성 선택 및 머신러닝 알고리즘을 이용한 네트워크 트래픽 기반 동적 접근 제어 방식의 순서도이다. 네트워크에 접근을 시도하는 모든 접근 주체에 대해 네트워크 트래픽 데이터를 수집한 후, 데이터 전처리 과정을 거친다. 전처리 과정은 특성 선택을 위해 중요도 분석이 수행된 후, 각 특성의 중요도가 임계값 이상인 경우에만 데이터를 유지하고, 임계값 이하의 특징은 삭제한다. 남은 데이터를 이용하여 데이터 인코딩 및 정규화를 수행한다. 전처리가 완료된 데이터는 머신러닝 알고리즘을 이용하여 정상 접근 여부를 판단한다. 정상 접근으로 판단된 경우에는 접근을 허용하고, 비정상 접근으로 판단된 경우에는 차단한다.

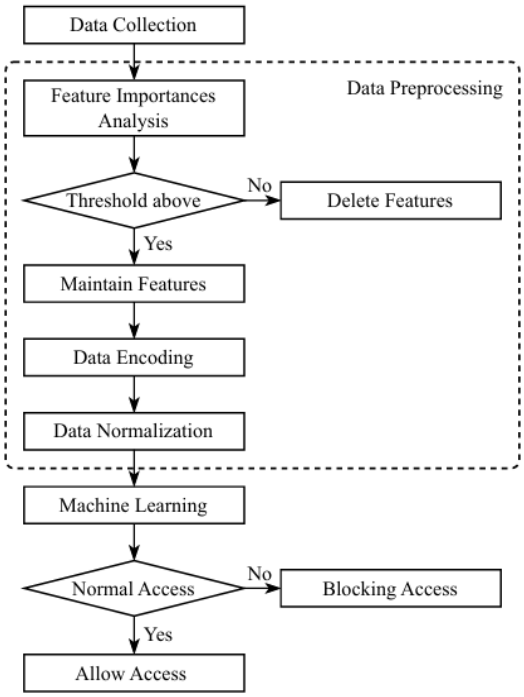

Fig. 2. Flowchart of a dynamic access control method based on network traffic data

3.3 적응형 제로 트러스트 검증 방식

제로 트러스트 모델의 경우, 경계 기반 모델과 같은 기존의 보안 모델보다 보안 수준을 강화할 수 있으나, 모델 도입 및 운영 비용이 상대적으로 비싸고 구현이 어렵다. 특히 사물인터넷과 같은 경량, 저전력, 저가 장치가 동작하는 환경에서는 자원 제약에 따른 데이터 처리 능력 및 저장 공간이 문제된다. 따라서 본 논문에서는 공격 상황이 의심되는 경우에만 제로 트러스트 보안 모델을 적용하는 적응형 제로 트러스트 검증 방식을 제안한다.

적응형 제로 트러스트 검증 방식은 공격이 탐지되지 않는 일반적인 상황에서는 일부 로그 데이터에 대해서만 검증을 수행하고, 공격 상황으로 의심되는 경우에는 제로 트러스트 모델을 적용하여 모든 로그 데이터를 검증한다. 적응형 제로 트러스트 검증 방식은 제로 트러스트 도입에 대한 장벽을 낮출 수 있으며, 적은 비용으로 빠르고 정확한 공격 탐지를 수행할 수 있다.

Fig.3.은 적응형 제로 트러스트 방식에 대한 순서도이다. 수집된 로그 데이터를 공격 상태에 따라 다르게 처리한다. 공격이 발생하지 않은 일반적인 상황에서는 일부 로그 데이터만 이용한다. 본 논문에서는 10개의 로그 데이터가 발생할 때, 그중 1개의 로그 데이터만 이용했다. 그리고 공격 상황이 의심되면 모든 로그 데이터를 이용한다. 이후에 전처리 과정부터는 3.3절에서 제안한 동적 접근 제어 방식과 동일한 동작을 수행한다.

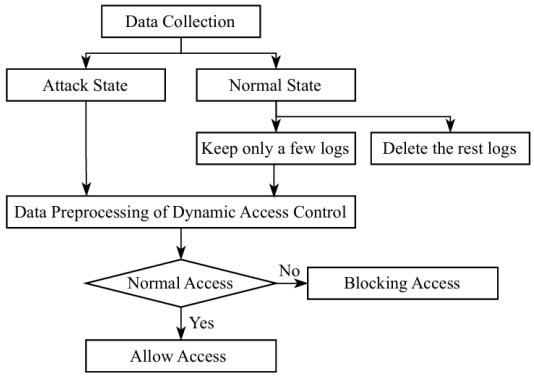

Fig. 3. Flowchart of Adaptive Zero Trust Method

IV. 성능 평가

4.1절에서는 네트워크 트래픽 기반 동적 접근 제어 방식 및 적응형 제로 트러스트 검증 방식의 성능을 입증하기 위해 진행한 실험 환경과 조건을 설명한다. 4.2절에서는 실험 결과를 분석한다.

4.1 성능 평가 환경 및 조건

4.1.1 데이터 셋

성능 평가를 위한 데이터 셋으로 KDD-NSL[16], CICIDS 2017[17]를 이용했다.

KDD-NSL 데이터 셋은 2009년 New School of Information Security & Technology(NSL)에서 제작되었으며 기존에 네트워크 침입 탐지 분야에서 활발히 이용되던 KDD'99 데이터 셋의 문제점을 해결했다. KDD-NSL은 대규모의 데이터 셋으로 다양한 유형의 네트워크 공격을 포함하며 실제 네트워크 트래픽을 기반으로 제작되어 현재까지도 네트워크 관련 연구에서 활발히 이용되고 있다.

KDD-NSL은 총 148,517개의 데이터로 구성되어 있으며 그중 77,054개가 정상(Normal) 데이터이고 71,463개가 비정상(Anomaly) 데이터이다. 공격 데이터는 DoS, Probe, U2R, R2L 등 총 4가지 종류의 네트워크 공격으로 이루어져 있다. 총 43개의 특성으로 구성되어 있으며, 이 중 42번 특성은 연결 수행 결과를 나타내어, 'normal' 또는 4가지 종류의 네트워크 공격에 포함된 항목의 값을 가진다. 'normal'을 제외한 모든 공격에 대한 값은 'attack'으로 대체했다. 43번 특성은 공격에 대한 난이도를 나타내는데, 본 시뮬레이션에서는 세부 공격에 대해 고려하지 않고 공격 여부만 고려하기 때문에 43번 특성은 삭제했다. 따라서 총 42개의 특성만을 이용한다.

CICIDS 2017 데이터셋은 2017년 Canadian Institute for Cybersecurity (CIC)에서 제작된 네트워크 침입 탐지 데이터 셋 중 하나로, 5일간 수집된 실제 네트워크 트래픽을 기반으로 다양한 유형의 네트워크 공격을 포함하고 있다. 이 데이터 셋은 현실적인 네트워크 환경에서 발생할 수 있는 여러 가지 침입 시나리오를 재현하여 네트워크 보안 연구와 머신러닝 기반의 침입 탐지 시스템 개발에 유용하게 사용되고 있다.

5일 간 수집된 데이터 중 수요일(Wednesday)에 해당되는 데이터는 정상 및 무작위 대입 공격(Brute Force Attack), DoS(Denial of Service) 공격, Heartbleed 공격 등의 트래픽이 혼합되어 있다. 총 692,703개의 데이터로 구성되어 있으며 그중 440,031개가 정상(Normal) 데이터이고 252,672개가 공격(Attack) 데이터이다. 총 79개의 특성으로 구성되어 있으며, 이 중 'Label' 특성에는 'Normal' 또는 각 공격의 세부 사항에 대한 값을 포함하고 있다. CICIDS 데이터 셋 역시 공격 세부 사항을 고려하지 않고 공격 여부만 고려했기 때문에, 'Normal'을 제외한 모든 공격 세부 사항은 'Attack'으로 대체했다.

4.1.2 데이터 전처리

데이터 전처리 과정의 첫 번째 과정은 특징 중요도 분석을 위해 MDI 방식을 이용한 특성 선택이다. 랜덤 포레스트 알고리즘을 이용하여 MDI 특성 선택을 수행했으며, 중요도 점수가 0.02보다 낮은 특성은 삭제했다.

KDD-NSL 데이터 셋의 경우, 총 42개의 특성 중 25개의 특성을 제외한 17개의 특성만 이용하여 데이터 셋을 구성했다. CICIDS 2017 데이터 셋의 경우, 총 62개의 특성을 제외한 17개의 특성만 이용하여 데이터 셋을 구성했다.

데이터 전처리의 두 번째 과정에서 데이터 인코딩을 위해 원 핫 인코딩 방식을 이용했다. 원 핫 인코딩 방식은 범주형 데이터를 수치형 데이터로 변환하는 기법으로, 특히 범주형 데이터 간의 순차적 특징이 없는 경우에 적합하다. Table 1은 특성 선택 과정을 통해 구성된 데이터 셋에 포함된 범주형 데이터이며, 해당 특성에 대한 데이터를 수치형으로 변환하기 위해 원 핫 인코딩을 수행했다. 이 중 KDD-NSL의 'outcome'과 CICIDS 2017의 'Label' 은 정상 및 공격 상태를 나타내는 특성으로 정상의 경우 0, 공격의 경우 1로 인코딩되었으며 해당 특성의 이름을 'label'로 수정했다. 또한 CICIDS 2017 데이터 셋의 경우, 무한대 값(infinity)과 특정 임계값을 초과하는 값을 처리하고, 결측값을 처리하는 과정을 수행했다.

본 연구에서 이용하는 데이터 셋에는 정수형 데이터가 포함되어 있으므로 데이터 정규화를 수행했고, 최소-최대 정규화(Min-max Normalization)를 이용했다.

Table 1. Categorical Feature List by Dataset

Dataset	Categorical feature list
KDD-NSL	protocol_type, service, flag, outcome
CICIDS 2027	Label

4.1.3 학습 모델과 파라미터

본 연구에서는 Random Forest, Decision Tree, Extreme Gradient Boosting(XGBoost)

알고리즘을 이용하여 로그 검증을 수행했다.

Random Forest 알고리즘은 여러 개의 결정 트리를 생성하고 각 트리의 예측 결과를 종합하여 최종 예측을 수행하는 앙상블 학습 기법이다. 각 트리는 랜덤하게 선택된 데이터 샘플과 특성으로 학습된다. 또한, 각 노드에서 최적의 분할을 찾는 대신 무작위로 일부 특성을 선택하여 분할 후보를 고려하여 트리들 사이의 상관 관계를 줄이고 모델의 일반화 성능을 향상시킨다.

Decision Tree 알고리즘은 데이터를 기반으로 하여 의사 결정 규칙을 학습하고 표현하는 모델이다, 트리 구조를 사용하여 각 노드는 특정 기준에 따라 데이터를 분할하며, 분할 기준은 주어진 데이터의 속성값에 따라 선택된다. 각 분할은 정보 이득이나 지니 불순도 등의 지표를 최적화한다.

XGBoost 알고리즘은 결정 트리를 기반으로 하는 부스팅 알고리즘이다. 부스팅 알고리즘은 모델이 이전에 학습한 결과를 기반으로 약한 학습자(weak learner)에 가중치를 부여하여 다음 단계의 학습 성능을 향상시킬 수 있도록 동작한다. 부스팅 알고리즘은 이전의 학습이 다음 단계의 학습에 영향을 미친다는 점에서 의존적으로 동작하기 때문에 모델 학습 시 속도가 느린 문제가 있다. XGBoost는 병렬 처리를 지원하여 학습 속도를 개선했으며, 대규모 데이터 셋에서도 높은 효율성을 보장한다. 또한 조기 종료를 수행하여 모델의 성능 향상이 발생하지 않으면 학습을 중단하고 과적합을 방지하여 최적의 반복 횟수를 통해 학습 및 검증이 수행되도록 한다[14]. XGBoost는 병렬 처리 방법을 활용하여 분류 속도가 빠르고 회귀 성능이 우수하며, 사용자 행동 분석과 이상탐지에 활용되고 있다[15].

데이터 전처리 과정을 수행한 후 머신러닝을 수행한다. 이 때 전체 데이터 셋의 80%를 학습 데이터 셋으로 이용했으며, 나머지 20%를 테스트 데이터 셋으로 이용하여 머신러닝을 수행했다.

Random Forest, Decision Tree, XGBoost의 분류 모델을 생성하고, 해당 모델을 학습시킨 후 예측을 수행했으며, 각 모델의 파라미터 설정은 Table 2, Table 3, Table 4와 같다.

Table 2. Setting Random Forest Parameters

Parameters	Value	Parameter description
n_estimators	100	Number of trees
max_features	auto	Max features for split
min_samples_split	2	Min samples to split node
max_depth	None	Max tree depth

Table 3. Setting Decision Tree Parameters

Parameters	Value	Parameter description
criterion	gini	Split quality measure
max_depth	None	Max tree depth
min_samples_leaf	1	Min samples at leaf node
max_features	None	Max features for split

Table 4. Setting XGBoost Parameters

Parameters	Value	Parameter description
booster	gbtree	Boosting: Tree
learning_rate	0.3	Learn Rate
max_depth	6	Max Depth
n_estimators	100	Boosting Rounds
objective	binary: logistic	Objective

4.1.4 적응형 제로 트러스트 실험 환경

적응형 제로 트러스트를 검증하기 위해 테스트 데이터 셋을 다음과 같이 재구성했다. KDD-NSL 데이터 셋 및 CICIDS 2017 데이터 셋은 효과적인 모델 학습 및 검증을 위해 정상 데이터와 공격 데이터의 비율이 유사하게 구성되어 있다. 실제로 검증을 수행하는 환경에서는 정당한 검증 수행을 요청하는 정상 데이터의 경우가 대부분이며 공격자에 의한 악의적 공격 데이터는 극히 일부분에 해당한다. 적응형 제로 트러스트 검증 방식의 시뮬레이션을 위해서 실제 환경을 고려한 테스트 데이터 셋을 구성했다.

테스트 데이터 셋은 기존 시뮬레이션과 동일한 크기를 유지하되, 정상 로그 비율을 달리하며 검증을 수행했다. 설정한 '고정 정상 로그 비율' 만큼 정상 데이터를 전반에 구성했으며, 나머지 데이터는 정상 및 공격 데이터에 상관없이 랜덤하게 추출하여 구성했다. 고정 정상 로그 비율은 55%, 75%, 95%로 설정했다.

제안한 방법은 데이터를 학습할 때, 공격이 탐지되지 않은 정상 상황에서는 로그 데이터를 10개당 1개만 검증하여 10%의 비율로 검증을 수행했다. 공격 상황은 오탐율을 줄이기 위해 공격이 5회 탐지되는 경우로

설정했다. 따라서 공격이 5회 탐지되면, 공격 상황으로 변경하고 모든 로그를 검증했다.

4.2 성능 평가 결과 및 분석

시뮬레이션을 통해 제안한 네트워크 트래픽 기반 저지연 동적 접근 제어 방식 및 적응형 제로 트러스트 검증 방식의 성능을 평가했다. 성능 평가에는 데이터 셋의 크기, 공격 탐지 성능, 처리 소요 시간 등의 평가 지표를 이용했다. 제안한 전처리 방식을 통해 데이터 셋의 크기를 줄여서 검증 소요 시간을 단축했으며, 동시에 공격 탐지 성능은 일정하게 유지할 수 있었다. 데이터 셋의 크기 및 검증 소요 시간은 저지연 달성 여부를 평가하기 위해 이용되었다. 공격 탐지 성능은 비정상적인 접근 시도를 정확하게 탐지하는지 평가하여 제안 방식의 신뢰성을 입증하기 위해 이용되었다.

4.2.1 데이터 셋 크기 비교

기존 데이터 셋과 제안한 데이터 전처리 과정을 거쳐 구성한 데이터 셋의 크기는 Table 5(KDD-NSL), Table 6(CICIDS 2017)와 같다.

KDD-NSL 데이터 셋의 경우, 기존 데이터 셋은 총 43개의 특성을 이용하고, 제안한 데이터 전처리 과정을 거쳐 구성한 데이터 셋은 총 17개의 특성을 이용한다. 제안 방법의 데이터 셋은 기존 데이터 셋에 비해 크기가 총 62.47% 감소했다.

CICIDS 2017 데이터 셋의 경우, 기존 데이터 셋은 총 79개의 특성을 이용하고, 제안한 데이터 전처리 과정을 통해 구성된 데이터 셋은 총 17개의 특성을 이용한다. 제안 방법의 데이터 셋은 기존 데이터 셋 대비 크기가 78.48% 감소했다.

제안한 특성 검증 방식을 통해 검증 수행 시 이용하는 데이터를 효과적으로 줄였다.

Table 5. (KDD-NSL) Compare dataset size of conventional and proposed methods (unit: KB)

Category	Conventional Dataset	Dataset of the proposed method
Train set	39858.77	14270.50
Test set	7133.19	3365.03
Total	46991.96	17635.53

Table 6. (CICIDS 2017) Compare dataset size of conventional and proposed methods (unit: KB)

Category	Conventional Dataset	Dataset of the proposed method
Train set	342021.85	73599.64
Test set	85505.77	18399.97
Total	427527.63	91999.61

4.2.2 탐지 성능 비교

제안한 데이터 전처리 과정을 통해 축소한 데이터 셋과 머신러닝 알고리즘의 탐지 성능을 입증하기 위한 성능을 평가했다. Random Forest, Decision Tree, XGBoost 등 3개의 단일 모델과 Gradient Boosting(GB)와 랜덤 포레스트(RF), Gradient Boosting과 XGBoost(XGB)를 결합한 2개의 하이브리드 모델을 이용하여 성능을 비교했다. 정확도(Accuracy), 정밀도 (Precision), 재현율(Recall), F1-점수 (F1-score) 등의 평가 지표를 이용했다. Fig.4.는 총 43개의 특성으로 구성된 KDD-NSL의 기존 데이터 셋을 이용한 결과이고, Fig.5.는 총 17개의 특성으로 구성된 제안 방식의 데이터 셋을 이용하여 검증한 결과이다.

기존 데이터 셋을 이용하는 경우에는 모든 모델에서 전반적으로 좋은 결과를 보였다. 제안한 전처리 과정을

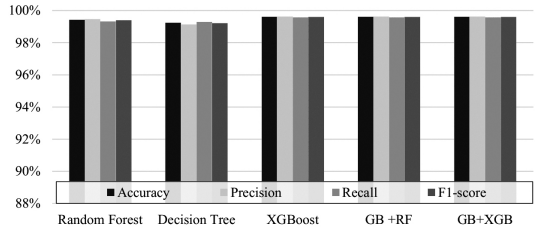

Fig. 4. (KDD-NSL) Comparison of detection performance using conventional dataset

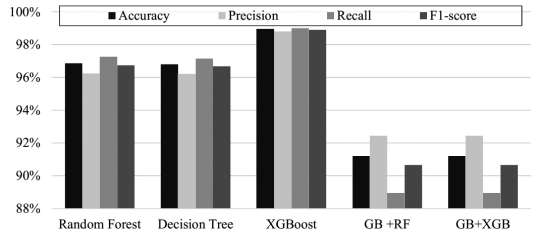

Fig. 5. (KDD-NSL) Comparison of detection performance using proposed dataset

통해 축소한 데이터 셋을 이용하는 경우에는 XGBoost를 제외한 모든 모델에서 성능 열화가 발생했다. 특히 하이브리드 모델은 기존 방식을 이용했을 때 약 99%의 정확도를 보였지만, 제안 방식을 이용한 경우에는 성능이 열화되어 약 91%의 정확도를 보였다. XGBoost의 경우, 모든 평가 지표에서 가장 높은 결과를 보였다.

Fig.6.은 총 79개의 특성으로 구성된 CICIDS 2017의 기존 데이터 셋을 이용한 결과이고 Fig.7.은 총 17개의 특성으로 구성된 제안 방식의 데이터 셋을 이용하여 검증한 결과이다.

CICIDS 2017 데이터 셋의 경우, 제안 방식의 데이터 셋을 이용했을 때, 정확도 측면에서 평균 0.13%의 미미한 탐지 성능 열화가 발생했다. 하지만 기존 방식과 제안 방식 모두 전체적으로 약 99% 이상의 높은 탐지 성능을 보였다.

Fig.8.은 데이터 학습 및 검증에 소요되는 실행 시간을 비교한 결과를 보여준다. KDD-NSL은 제안 방식의 데이터 셋을 이용한 경우, 기존 데이터 셋을 이용한 경우와 비교했을 때, 실행 속도가 평균 58% 감소했다. 특히 XGBoost는 기존 데이터 셋 이용 시 0.74초를 소요하여 가장 빠른 검증을 수행했고, 제안 방식을 적용한 경우에도 0.33초를 소요하여 좋은 성능을 보였다.

CICIDS 2017은 제안 방식의 데이터 셋을 이용한 경우, 실행 속도 측면에서 기존 데이터 셋 대비 평균

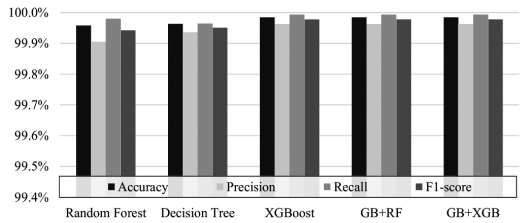

Fig. 6. (CICIDS 2017) Comparison of detection performance using conventional dataset

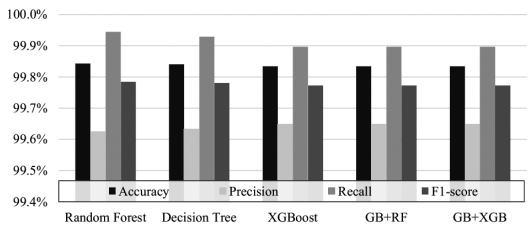

Fig. 7. (CICIDS 2017) Comparison of detection performance using proposed dataset

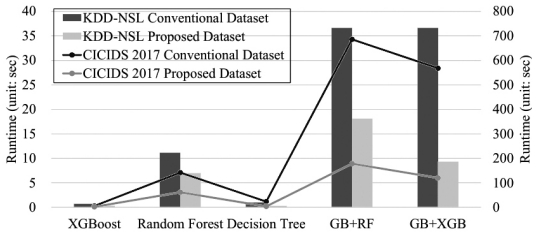

Fig. 8. Comparison of Runtime for data training and validation

71% 감소했다. 제안 방식의 데이터 셋을 이용했을 때, XGBoost의 경우 2.16초, Decision Tree의 경우 3.56초를 소요하여 가장 좋은 성능을 보였다.

본 논문에서 제안한 데이터 전처리 과정을 적용한 데이터 셋을 이용하여 데이터 셋의 크기와 검증 수행 속도를 효과적으로 줄일 수 있었다. KDD-NSL의 경우, 제안한 데이터 셋을 이용했을 때, 기존 데이터 셋과 비교하여 약간의 탐지 성능 저하가 발생했는데, XGBoost 알고리즘의 경우 약 0.6%의 성능 저하만 관찰되었다. 대용량 데이터인 CICIDS 2017의 경우, 총 62개의 데이터 특성을 제거하여 데이터 셋의 크기를 크게 줄였음에도 불구하고 기존 데이터 셋과 제안 데이터 셋의 탐지 성능 차이거 거의 발생하지 않았다.

본 연구의 실험 결과에 따르면, 제안된 전처리 과정을 활용할 경우에 이용하는 데이터의 특성 수를 줄여 검증 수행 속도를 단축하면서도 높은 탐지 성능을 유지할 수 있음을 입증했다. 특히 XGBoost 알고리즘을 이용하면 원본 데이터 셋의 크기와 상관없이 유사한 탐지 성능을 유지할 수 있음을 보였다.

4.2.3 적응형 제로 트러스트 성능 평가 결과 분석

본 연구에서 제안한 적응형 제로 트러스트 방식의 효과를 입증하기 위해 성능을 비교 평가하기 위해 고정 정상 로그 비율을 55%, 75%, 95%로 조정하고 XGBoost 알고리즘을 이용했다.

Fig.9.은 정상 로그 비율에 따른 검증을 수행하는 데이터 셋의 크기와 실행 시간을 비교한 그래프이다. 테스트 데이터 셋은 네트워크 트래픽 기반 저지연 동적 접근 제어 방식과 동일하게 이용하지만, 정상 상황에서는 10%의 로그만 검증을 수행하기 때문에 실제로 검증하는 데이터의 크기에 차이가 있다. 고정 정상 로그 비율이 0%인 경우는 네트워크 트래픽 기반 저지연 동적 접근 제어 방식의 전처리 과정을 적용한 데이

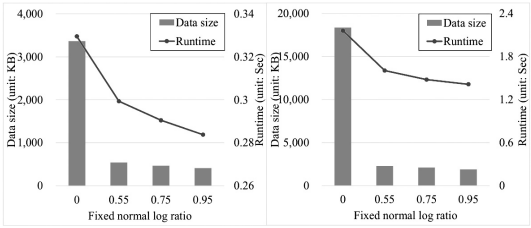

Fig. 9. Comparison of data size and Runtime according to fixed normal log ratio

터 셋을 의미한다. 실제 검증에 사용된 데이터 셋의 크기를 비교했을 때, 적응형 제로 트러스트 방식을 적용한 경우 데이터 셋 크기가 KDD-NSL은 평균 86%, CICIDS 2017은 평균 89% 감소한 것을 확인할 수 있다. 또한 실행시간 측면에서도 KDD-NSL은 평균 12%, CICIDS 2017은 평균 31%의 감소를 보였다.

Fig.10.과 Fig.11.의 그래프에서는 KDD-NSL 데이터 셋 및 CICIDS 2017 데이터 셋의 고정 정상 로그 비율에 따른 탐지 성능을 비교했다.

KDD-NSL 데이터 셋은 고정 정상 로그 비율을 95%로 설정했을 때 탐지 성능이 평균 7.8% 열화되었다. 고정 정상 로그 비율이 55%, 75%인 경우는 탐지 성능이 평균 0.6%, 1.3% 열화되었지만, 모든 평가 지표에서 95% 이상의 높은 탐지 성능이 관찰되었다.

CICIDS 2017 데이터 셋은 탐지 성능 측면에서 비교적 높은 탐지 성능을 보였다. 고정 정상 로그 비율을 55%, 75%로 설정한 경우, 탐지 성능이 평균 0.5% 미만의 열화가 발생했지만, 모든 평가 지표에서 98% 이상의 높은 성능을 보였다. 고정 정상 로그 비율이 95%인 경우는 정밀도 측면에서 약 7.4%, F1-Score 측면에서 약 4%의 성능 열화를 보이며 오탐율이 증가했다.

본 연구에서는 적응형 제로 트러스트 검증 방식의 우수한 성능을 입증했다. 적응형 제로 트러스트 검증 방식은 네트워크 트래픽 기반 동적 접근 제어 방식을

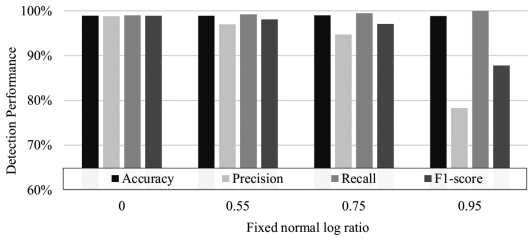

Fig. 10. (KDD-NSL) Comparison of detection performance with fixed normal log ratio

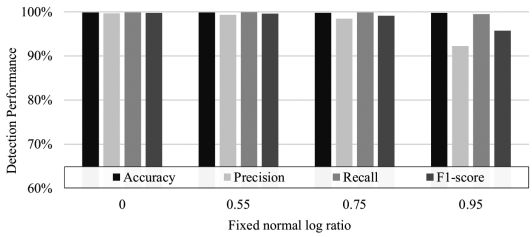

Fig. 11. (CICIDS 2017) Comparison of detection performance with fixed normal log ratio

적용한 경우와 비교하여 데이터 셋의 크기 및 검증 수행 속도를 더 효과적으로 줄일 수 있었다. 특히, 고정 정상 로그 비율을 55%로 설정한 경우에는 실행시간이 각각 9.1%, 25.8% 감소했으며, 탐지 성능은 1% 미만의 열화만을 보이며 큰 성능 저하 없이 검증을 수행할 수 있었다. 더불어, 고정 정상 로그 비율을 55%로 설정한 적응형 제로 트러스트 검증 방식과 네트워크 트래픽 기반 동적 접근 제어 방식을 결합하여 기존 데이터 셋 크기 대비 약 94% 감소한 데이터 셋을 이용하여 높은 탐지율을 달성했다.

V. 결 론

본 논문에서는 제로 트러스트 보안 모델을 효율적으로 활용하기 위해 SDP를 활용한 실시간 이벤트 로그 기반 동적 접근 제어 방식을 제안했다. 특성 선택을 포함한 데이터 전처리 과정과 머신러닝 알고리즘을 통해 데이터 셋의 크기를 줄이는 동시에 높은 탐지 성능을 유지한다. 실험 결과에 따르면 데이터 셋의 크기를 효과적으로 축소시키면서 지연을 최소화하는 동시에 탐지 성능을 비슷하게 유지할 수 있었다.

또한 제로 트러스트 모델 도입의 어려움을 해결하기 위해 적응형 제로 트러스트 검증 방식을 추가로 제안한다. 네트워크 트래픽 기반 동적 접근 제어 방식과 함께 적용하면 데이터 셋의 크기 및 검증 수행 시간 측면에서 효과적인 개선을 이루어 내며, 기존 방식과 유사한 탐지 성능을 보인다.

제로 트러스트 모델은 높은 보안 수준을 달성할 수 있지만 성능 요구사항을 만족시키며 운영하기 어렵다. 본 논문은 제로 트러스트 모델을 효율적으로 활용할 수 있는 소프트웨어 정의 경계를 이용한 네트워크 트래픽 데이터 기반 동적 접근 제어 방식을 제안했다.

본 연구에서는 특성 선택을 통해 데이터 셋의 크기를 줄임으로써 저지연을 달성하였다. 이는 특정 환경에

서 성능을 극대화하기 위한 방법으로 유용하게 작용하였으나, 제로 트러스트가 이용되는 환경은 매우 다양하며 각 상황에 따라 요구되는 최적화 방식이 다를 수 있다. 본 연구는 특정한 환경에서 성능 평가했지만 실제 네트워크 환경은 매우 다변적이며, 각 환경마다 요구되는 보안 수준과 성능 요구사항이 상이할 수 있다.

후속 연구에서는 다양한 네트워크 환경과 상황을 시뮬레이션하여 본 연구에서 제안한 방법의 적용 가능성과 성능을 검증할 계획이다. 후속 연구를 통해 제로 트러스트 환경의 다양한 현실적 요구사항을 충족시킬 수 있는 보편적이고 유연한 보안 솔루션을 연구개발할 계획이다.

후속 연구에서는 다양한 네트워크 환경과 상황을 시뮬레이션하여 본 연구에서 제안한 방법의 적용 가능성과 성능을 검증할 계획이다. 이를 통해 다양한 환경에서도 안정적이고 효율적으로 동작할 수 있는 제로 트러스트 보안 솔루션을 개발하고, 본 연구의 한계를 보완하여 보다 넓은 범위에서의 적용 가능성을 확인할 것이다. 후속 연구를 통해 제로 트러스트 환경의 다양한 요구사항을 충족시킬 수 있는 보편적이고 유연한 보안 솔루션을 제안하는 것을 목표로 하고 있다.

References

〔1〕 Syed, N. F., Shah, S. W., Shaghaghi, A., Anwar, A., Baig, Z., & Doss, R., "Zero trust architecture (zta): A comprehensive survey," IEEE Access, Volume: 10, pp.57143-57179, 12 May 2022.

〔2〕 Teerakanok, S., Uehara, T., Inomata, A., "Migrating to zero trust architecture: Reviews and challenges," Security and Communication Networks, pp. 1-10, 25 May 2021

〔3〕 TN, N., Pramod, D., Singh, R., "Zero trust security model: Defining new boundaries to organizational network," In Proceedings of the 2023 Fifteenth International Conference on Contemporary Computing, pp. 603-609, August. 2023

〔4〕 So-Hui Kim, Il-Gu Lee, Ye-Eun Lee, "Practical Zero Trust Technology and Policy for 5G Network Convergence Industry," Korean Journal of Industrial Security, Vol. 14, pp.203-222, Jan. 2024

〔5〕 Fernandez, Eduardo B., and Andrei Brazhuk, "A critical analysis of Zero Trust Architecture (ZTA)," Computer Standards & Interfaces, volume: 89, 103832, April. 2024

〔6〕 Zolotukhin, M., Hämäläinen, T., & Kotilainen, P., "Intelligent solutions for attack mitigation in zero-trust environments. In Cyber Security: Critical Infrastructure Protection", Cham: Springer International Publishing, COMPUTMETHODS, volume 56, pp. 403-417, 2022

〔7〕 S. Rose, O. Borchert, S. Mitchell, S. Connelly, "Zero Trust Architecture," NIST Special Publication 800-207, National Institute of Standards and Technology, U.S. Department of Commerce, August 2020.

〔8〕 Park, C. S., "Zero Trust Architecture." Review of Korea Institute of Information Security and Cryptology (KIISC), volume. 33, No. 4, pp.131-141, August 2023

〔9〕 Lee. J. Y, Choi. B. H, Koh. N, Chun. S, "A Study on the Establishment of Zero Trust Security Model," Korea Information Processing Society/Computer and Communication System, volume: 12, No. 6, pp. 189-196, June 2023

〔10〕 Devarakonda, A., Sharma, N., Saha, P., & Ramya, S., "Network intrusion detection: A comparative study of four classifiers using the NSL-KDD and KDD'99 datasets," In Journal of Physics: Conference Series, IOP Publishing, Vol. 2161, No. 1, p. 012043, 2022

〔11〕 Yao, Q., Wang, Q., Zhang, X., Fei, J.,

"Dynamic access control and authorization system based on zero-trust architecture," In Proceedings of the 2020 1st international conference on control, robotics and intelligent system, pp. 123-127, October. 2020
〔12〕 Jin, Q., Wang, L., "Zero-trust based distributed collaborative dynamic access control scheme with deep multi-agent reinforcement learning," EAI Endorsed Transactions on Security and Safety, 27 August 2022
〔13〕 Jeon, So-Eun, et al. "Suboptimal Feature Selection Techniques for Effective Malicious Traffic Detection on Lightweight Devices." CMES-Computer Modeling in Engineering & Sciences, 140(2), May. 2024
〔14〕 Gouveia, A., Correia, M., "Network intrusion detection with XGBoost," In Recent Advances in Security, Privacy, and Trust for Internet of Things (IoT) and Cyber-Physical Systems (CPS), Chapman and Hall/CRC, pp. 137-166, 2020
〔15〕 Deng, Y., Lumley, T, "Multiple imputation through xgboost", Journal of Computational and Graphical Statistics, volume: 33, No: 2 pp.1-19, October. 2023
〔16〕 M. Tavallaee, E. Bagheri, W. Lu, and A. Ghorbani, "A Detailed Analysis of the KDD CUP 99 Data Set," 2009 IEEE symposium on computational intelligence for security and defense applications (CISDA), p. 1-6, July. 2009
〔17〕 Iman Sharafaldin, Arash Habibi Lashkari, and Ali A. Ghorbani, "Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization", 4th International Conference on Information Systems Security and Privacy (ICISSP), Portugal, January 2018

〈저자소개〉

김 서 이 (Seo-Yi Kim) 학생회원
2023년 8월: 성신여자대학교 융합보안공학과 졸업
2023년 9월~현재: 성신여자대학교 미래융합기술공학과 석사과정
〈관심분야〉 이상탐지, 융합보안

이 일 구 (Il-Gu Lee) 종신회원
2003년 2월: 서강대학교 전자공학 학사
2005년 2월: KAIST 정보통신 석사
2016년 2월: KAIST 전산학부 박사
2005년 2월~2017년 2월: 한국전자통신연구원 선임연구원
2017년 3월~현재: 성신여자대학교 융합보안공학과/미래융합기술공학과 부교수
2024년 2월~현재: 성신여자대학교 융합보안공학연구소 소장
〈관심분야〉 융합보안, 미래융합기술, 정보보호, 정보통신