Journal of KIIT. Vol. 24, No. 2, pp. 43-53, Feb. 28, 2026. pISSN 1598-8619, eISSN 2093-7571
http://dx.doi.org/10.14801/jkiit.2026.24.2.43

심층 강화학습 기반 사이버 전장 시뮬레이션 공격 기술 연구

김정현*, 김민서**, 김민석***

Research on Attack Techniques for Cyber-Range Simulations based on Deep Reinforcement Learning

Jung-Hyun Kim*, Min-Seo Kim**, and Min-Suk Kim***

This research was supported by a 2024 Research Grant from Sangmyung University

요 약

고도화된 정보통신 환경을 기반으로 운영되는 시스템은 사이버 공격 및 고도화로 인해 지속적으로 보안 취약점이 발생하고 있다. 이러한 위협에 대응하기 위해 규칙 기반의 기존 방식이 아닌 인공지능 기법인 강화학습을 이용한 능동적 대응 체계로의 전환이 주목받고 있다. 본 논문에서는 실제 공격 절차를 모사한 사이버 전장 시뮬레이션 환경을 구축하고 DQN, PPO, SAC를 적용하여 공격 에이전트의 학습 성능을 비교 분석하였으며, 도출된 공격패턴을 기반으로 대응 기법의 유효성을 검증하였다. 최적 공격 시퀀스 목표가 9개인 조건에서 DQN과 SAC는 목표와 동일한 9개 시퀀스를 도출했으나, PPO는 평균 30개 이상을 생성해 목표 대비 약 3.3배 낮은 최적화 효율을 보였다. 실험 결과는 향후 공격 패턴 분석 및 방어 전략 검증에서 강화학습의 기법별 성능을 정량적으로 비교하고 평가하는 근거를 제공한다.

Abstract

Systems operating based on advanced information and communication environments continue to generate security vulnerabilities, and cyber-attacks are also on the rise. In order to cope with these threats, the transition to an active response system using reinforcement learning, an artificial intelligence technique, is drawing attention. In this paper, we propose a cyber battlefield simulation that mimics real attack procedures and evaluates DQN, PPO, and SAC for training attack agents and validating defense measures. With a target of nine optimal attack sequences, DQN and SAC met the target, while PPO averaged over 30 sequences. These results support quantitative benchmarking of RL methods for cyber-attack pattern analysis and defense strategy validation.

Keywords

reinforcement learning, network security, cyber-range, simulation, unreal engine

* 상명대학원 전자정보시스템공학과 박사과정
- ORCID: https://orcid.org/0009-0003-0940-5979
** 상명대학교 휴먼지능로봇공학과 학사과정
- ORCID: https://orcid.org/0009-0007-9160-2638
*** 상명대학교 휴먼지능로봇공학과 교수(교신저자)
- ORCID: https://orcid.org/0000-0003-4519-1683

· Received: Jan. 06, 2026, Revised: Feb. 12, 2026, Accepted: Feb. 15, 2026
· Corresponding Author: Min-Suk Kim
Dept. of Human Intelligence and Robot Engineering, Sangmyung University, Cheonan, 31066, Korea
Tel.: +82-41-550-5113, Email: minsuk.kim@smu.ack.kr

Ⅰ. 서 론

현대에 이르러 국가와 기업은 고도화된 정보통신 환경을 기반으로 운영되고 있다. 이에 지속해서 새로운 취약점이 발생하고, 사이버 위협 역시 다양해지고 있다. 최근 발생한 여러 정보 유출 사고는 통신사들이 네트워크 위협 탐지와 대응, 보안 패치 등에 미흡했던 점이 주요 원인으로 지적되었으며[1], 취약점을 악용하는 공급망 공격이 새로운 주요 공격 기술로 나타나고 있다[2]. 이러한 사이버 위협에 대응하는 전통적인 보안 기술은 규칙 기반 기술이며 미리 정의된 공격패턴에 대응하는 것에는 뛰어난 성능을 보인다. 하지만 변칙적인 공격 방식에 대해서 즉각적인 대응이 어려운 구조적 한계를 지닌다[3][4]. 최근 공격자들은 생성형 AI 기반 공격 기법을 통해 악성코드와 회피 전략 생성 시간을 단축하고, 사회공학적 공격을 포함한 자동화 기법도 정교해지고 있다. 이에 따라 탐지·대응 분야에서도 AI 기반 기법이 활발히 연구되고 있다. 하지만 기존 지도 및 비지도학습 기반 AI 모델은 학습 데이터에 포함되지 않고 시간에 따라 분포가 바뀌는 새로운 공격 기법에 대해서는 대응이 어렵다. 따라서 실시간 환경에서 지속적 업데이트가 수반되는 보완 기법이 필요하며[5], 사전에 수집한 데이터에 의존하여 학습하는 구조적 한계를 해결하고 빠르게 변화하는 환경에 즉각 반응하는 보안 시스템의 필요성이 강조되고 있다. 이러한 문제를 해소하기 위해 환경과 직접 상호작용하고 실시간으로 문제를 피드백 받을 수 있는 강화학습 기법이 필요하며, 이는 마르코프 결정 과정(MDP, Markov Decision Process)을 기반으로 보안 환경을 학습하여 모델링을 진행할 수 있다. MDP는 일반적으로 환경 상태(S)에서 에이전트가 행동(A)을 실행했을 때 보상(R)을 얻으며 행동으로 인해 상태변화 확률(P)을 기본적인 구조로 가지고 이를 기반으로 에이전트가 환경과 상호작용을 통해 시행착오를 겪으며 누적 보상을 최대화하는 정책을 최적화한다. 강화학습은 MDP를 기반으로 환경 변화에 적응하며 정책을 지속적으로 업데이트할 수 있어 새로운 공격 및 방어 정책 생성에 유용하다[6]. 학습을 위한 강화학습의 환경 모사 방식은 다양하다. 특히 물리적인 공간 모사를 필요로 하는 ROS2[7], Unreal Engine[8][9]을 사용한다면, 보다 현실에 가까운 환경을 모사할 수 있다.

본 논문은 강화학습을 이용한 학습 모델링 및 성능 검증, 고도화를 위해 Unreal 엔진 기반 사이버 공방 시뮬레이션을 설계 및 제작하였다. 또한, 강화학습 에이전트를 활용한 자율적 공격 정책 생성은 매우 중요하지만, 검증되지 않은 사이버 환경에서 생성된 공격 에이전트는 그 타당성과 신뢰성이 낮을 수 있으므로, 이를 해결하기 위해 네트워크 보안 시나리오를 재설계하여 실제 해커가 작성한 Ground Truth 시퀀스를 기준으로 방어자가 부재한 상황에서 에이전트가 해당 시퀀스를 유사하게 재현하는지에 대한 검증을 통해 공격 모델의 정당성을 확보하였다.

본 논문의 구성은 다음과 같다. 2장에서는 심층 강화학습 기반 사이버 공격 기술과 사이버 보안 대응 전략 시뮬레이션에 대한 설명, 3장에서는 Unreal Engine을 기반으로 제작한 시뮬레이션의 공격 행동과 보상 설계에 관한 내용을 작성하였다. 4장에서는 제작된 환경에서 강화학습 공격자의 공격 결과와 성능에 대해 분석하였고, 마지막으로 5장에서 결론과 향후 작업을 설명하였다.

Ⅱ. 관련 연구

강화학습 기반 사이버 보안 분야의 공격과 방어 전략 수립 관련 연구는 사이버 전장 시뮬레이션은 공격 절차를 파악하고 이에 대응하는 방어 요소를 분석하며, 향후 대응 전략을 도출해 실제 상황에 대비하는 데 매우 중요한 역할을 한다.

2.1 심층 강화학습 기반 사이버 공격 기술

사이버 공격의 보안 기술의 자동화를 위해 환경 적응형 AI 기반 연구가 다양하게 진행되고 있다[10]. 특히, 강화학습은 공격 에이전트가 환경과 상호작용을 통해 학습을 진행하여 환경 적응력이 뛰어난 학습 모델링 방법이다[11]. 일반적으로 강화학습은 에이전트가 사용하는 행동 공간에 따라 가치 기반과 정책 기반 강화학습 방법으로 구분된다. 가치 기반은 이산적인 행동 공간 중심의 학습 학습하며 이산적인 상태 공간에서 뛰어난 성능을 보인다. 정책 기

반 방법은 연속적인 행동 공간에서 학습을 진행한다. 현재 사이버 공격 기술의 대부분은 이산적인 행동 공간과 상태 값을 가지고 있기 때문에 공격자는 정보를 취득하며 노드 점거나 탈취 같은 최종 목표를 달성하고자 할 때 MDP를 이용하여 모든 상태변화 확률의 정답 값을 인식하고 문제 풀이를 진행한다[12][13]. 또한, 사이버 전장에서 공격자가 정보를 취득해야 할 경우에는 POMDP(Partially Observable Markov Decision Process)의 구조를 채택해야 할 필요성이 있다. POMDP는 부분 관측 공간을 사용하여 학습하는 구조로 에이전트가 학습을 통해 관측 상태 정보를 취득하며 학습하는 구조를 가진다[14]. 추가로 강화학습에서 보상 함수의 설계는 에이전트의 정책 생성과 성능에 있어 큰 영향을 끼치며, 이를 위해 공격자가 학습 과정에서 탐색 실패 문제를 완화하는 보상 재설계, 난이도 조절 시나리오 랜덤화 등 다양한 방법들이 제안되고 있다[15].

2.2 사이버 보안 대응 전략 시뮬레이션

일반적으로 강화학습 기반 모델링 기법에서는 시뮬레이션, 에뮬레이션 같은 환경이 함께 고려되어야 한다. 마이크로소프트의 CBS(CyberBattleSim)은 여러 종류의 노드를 정의하여 네트워크 환경을 구현하여 공격자가 로컬 혹은 원격으로 공격을 진행하고 노드를 연결하는 방식이다[16]. 이는 시각적으로 공격자가 어떤 노드를 공격하고 있는지에 대한 노드의 상태를 쉽게 확인할 수 있는 장점이 있으며 공격 기술은 크게 3가지로 구분하여 로컬 공격, 원격 공격, 노드 접속으로 정의되어 있다. 또한, 관측 공간 역시 노드를 관측하거나 점거했는지를 표현하고, 자격 증명과 권한 단계, 공격 가능 유무만을 사용하기 때문에 다소 단순한 상태변화를 확인할 수 있다[17]. 네트워크 공격 시뮬레이션인 NASim(Network Attack Simulation)은 이전 CBS와 비교하면 좀 더 세부적인 공격자의 행동과 관측 공간을 포함하고 있다. 먼저 Subnet과 topology를 통해 네트워크의 연결 상태와 하위 노드의 연결 개수를 설정할 수 있고, 노드별로 OS, Service, Processes의 정보를 세부적으로 구성할 수 있다. 이는 네트워크 상태를 좀 더 현실적으로 제작할 수 있고 행동 역시 큰 카테고리로 Exploits와 Privilege Escalation, Scan으로 분리되어 있어 공격자가 특정 네트워크에서 목적 노드를 위해 실행되는 공격 순서에 대한 파악이 쉽다. 또한, NASim은 공격 시나리오를 쉽게 변경할 수 있다. Tiny, Small 등 기존의 다양한 시뮬레이션 시나리오 환경을 새롭게 정의하여 Tiny-Alpha, Small-Alpha와 같은 시나리오를 통해 강화학습 모델 성능의 비교 검증을 할 수 있다[18]. 하지만, NASim은 공격 행위를 중심으로 시뮬레이션 환경으로 국한되어 있어 다양한 사이버 공격 시나리오를 통해 환경 구조 및 상태 정보를 발전시킬 필요성이 있다[19].

III. Unreal Engine 기반 시뮬레이션 기술

기존 CBS나 NASim[20]은 제공되는 한정된 시뮬레이션 환경 위주로 공격 기술이 정의되고 있다. 이는 공격과정을 단순화하여 환경 정보를 쉽게 파악할 수 있다는 장점이 존재하지만, 대략적인 상태 정보만을 이용하므로 구체적으로 어떤 공격과정을 통해 학습을 진행하는지 파악하는 데 한계가 있다. 따라서 본 논문에서는 확장성이 뛰어나고 환경과 에이전트가 통신하는 방법을 적용하여 실제 환경과 유사한 구조로 모사가 가능한 Unreal Engine을 채택하여 기존 단점을 보완하고 공격 시나리오와 시뮬레이션 환경을 새롭게 설계하여 검증을 진행하였다.

3.1 공격 시뮬레이션 환경 설계

사이버 전장을 설계할 때 주요하게 고려되어야 할 사항은 환경을 실제 공간보다 얼마나 간략화하여 표현할 것인지 판단하는 것이다.

공격 위주의 사이버 전장 시뮬레이션인 NASim 경우 네트워크 구조를 비교적 상세하게 설계할 수 있지만, 공격 기술은 크게 간략화되어 있다. 또한, 학습에 있어 낮은 수준을 요구하지만, 간략화된 공격 기술은 단순 공격의 순서만을 파악하는 것을 중점으로 진행되기 때문에 상세한 공격 순서를 확인하는 것에는 한계가 존재한다. 그림 1은 NASim의 Tiny 시나리오를 간단하게 표현한 것이다. Tiny 시나리오에서 공격 기술은 Scan, Exploits, Privilege escalation으로 되어 있다. 이때 시나리오에서 정의

된 운영체제와 서비스, 프로세스 정보를 기반으로 Scan 또는 Exploits 행동이 작동하게 되어 있다. 즉, 정확히 특정 공격 기술을 사용하는 것이 아닌, 에이전트가 전체적으로 통합적인 기술을 사용했다고 가정하고 작동하는 것이며, 이는 네트워크에 대한 공격의 순서를 파악할 수 있지만, 실제 환경과는 다소 차이가 있다.

본 논문에서는 위와 같이 간략화된 공격 행동들을 세부적으로 구분하여 현실적인 공격 순서를 생성할 수 있는 고도화된 공격 메커니즘 기술을 구현한다. 더하여 구현된 공격 도구가 실제 환경에서 실행되는데 필요한 조건들을 확인하여 성공 여부를 판단할 수 있는 구조를 포함하는 사이버 전장 시뮬레이션 환경을 구축하였다. 특히 해당 시스템의 확장성을 확보하기 위해 Unreal Engine을 기반으로 시뮬레이션 환경을 개발하였고 환경·서버·클라이언트 간의 유기적인 통신 아키텍처를 통해 에이전트의 행동 결과가 실시간으로 반영되는 상호작용 구조를 설계하였다.

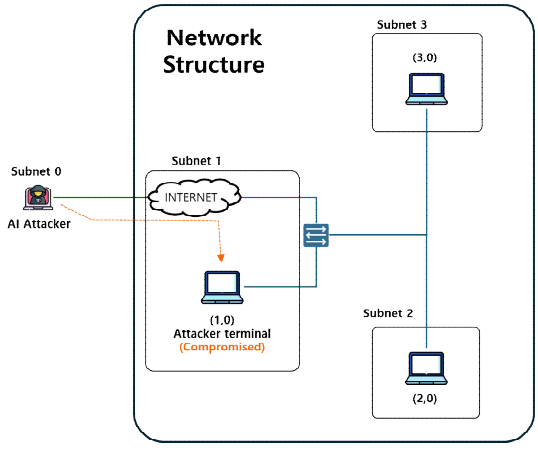

그림 1. NASim의 Tiny 시나리오 아키텍처
Fig. 1. NASim Tiny scenario architecture

그림 2는 전체적인 통신 구조를 나타내고 있다. 해당 서버는 전체적인 네트워크 결과 반영과 통신을 관리하는 역할을 담당한다. 그리고 시뮬레이션 환경을 담당하는 클라이언트와 환경 내에서 활동하는 에이전트를 담당하는 클라이언트로 구분하고 있으며 에이전트가 사용하는 행동 공간(Agent space)은 최초 한 번만 실행하여 행동을 수행할 수 있게 설계하였다.

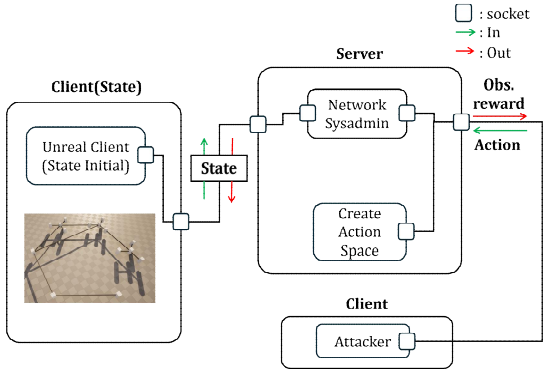

그림 2. Unreal 엔지 기반 네트워크 파이프라인
Fig. 2. Unreal engine based network pipeline

공격 기술 역시 세부화하기 위해 실제 MITER ATT&CK 기술[21]을 참고하여 사전 조건과 결과를 반영하는 PPS(Pre-Post State)를 설계하였다. 이때 공격자 행동은 총 40종의 공격 기술을 구현하였으며 공격 카테고리는 11개의 카테고리로 구분하였고 분류된 카테고리는 다음과 같다.

1. 정찰(Reconnaissance): 공격자가 공격을 시도하기 위한 정보를 수집하는 행위로 T1595.001@1, T1595.001@2, T1505.002@3 등 6가지 행동이 있음
2. 초기 접근(Initial access): 공격자가 네트워크에 접속을 시도하는 행위로 T1190@16, T1190@3, T1190@5 등의 3가지 행동이 정의됨
3. 실행(Execution): 공격자가 네트워크 또는 개인 컴퓨터에 심어둔 악성코드를 실행하는 행위로 T1059.003@3, T1059.007@1 등 2가지 행동이 있음
4. 유지(Persistence): 공격자가 이미 행한 행동에 관한 결과를 유지하려는 행동으로 T1053.005@1으로 정의됨
5. 권한 상승(Privilege Escalation): 공격자가 시스템 내에서 높은 권한을 탈취하려는 행위로 T1068@1의 한 가지 행동으로 정의됨
6. 오탐지(Defense evasion): 공격자가 방어자가 탐지하려는 행위를 방해하는 행동으로 T1036.004@1, T1036.005@2, T1134.001@1 등 5가지 행동이 있음
7. 자격 증명 획득(Credential access): 공격자가 계정과 비밀번호를 훔치는 행위로 T1003.002@1으로 정의됨

8. 탐색(Discovery): 공격자가 사용자의 환경을 파악하는 행위로 T1016.001@1, T1033@2, T1082@2 등 5가지 행동이 있음
9. 데이터 획득(Collection): 공격 목적에 도움이 되는 데이터수집을 위해 T1005@1, T1074.001@3, T1074.001@4 등의 5가지 행동으로 정의됨
10. 지휘통제(Command and control): C&C 공격으로도 정의하며 악성코드에 감염된 시스템과 통신하며 제어하는 행위로 T1105@23, T1105@24의 2가지 행동이 있음
11. 정보 유출(Exfiltration): 컴퓨터 시스템이나 네트워크에서 정보를 탈취하기 위한 행위로 T1048.003@1, T1048.003@2, T1048.003@6 등 9가지 행동으로 정의됨

본 논문에서 사용하는 행동 공간은 크게 그림 3과 같이 2가지로 구분하였고, 공격 기술 40종과 에이전트 공격하는 노드를 선택하는 액션으로 분리하여 학습 효율을 증가시켰다. 정의된 공격 기술들은 카테고리에 맞는 결과를 도출하며 사전 조건(Pre-state)의 여부에 따른 성공에 따라 보상이 주어진다. 이때 보상 함수는 식 (1)과 같다.

Num	Action	Num	Action	Num	Action	Num	Action
1	T1595.001@2	11	T1059.007@1	21	T1033@4	31	T1041@2
3	T1595.002@5	13	T1134.001@1	23	T1082@2	33	T1041@4
5	T1595.002@8	15	T1036.004@1	25	T1074.001@3	35	T1048.003@2
7	T1190@3	17	T1140@1	27	T1560.003@1	37	T1048.003@7
9	T1053.005@1	19	T1016.001@1	29	T1105@23	39	T1048.003@9

Num	Node
1	(2,0)
3	(3,1)
5	(3,3)
7	(4,1)
9	(4,3)
11	(5,1)
13	(5,3)

그림 3. 공격자 에이전트 행동공간
Fig. 3. Attacker agent action space

$$R(s_t,a_t)=\begin{cases}+1 & \text{if } S_{action}\\ +1\times N & \text{if } S_{scan}\\ +100 & \text{if } S_{compromised}\\ -1 & else\end{cases} \quad (1)$$

$$\text{cost}=2$$
$$R'(s_t,a_t)=R(s_t,a_t)-\text{cost}$$

기본적으로 행동에 비용(cost)이 존재하며 얻을 수 있는 보상(R)에서 행동 비용이 차감된다. 이는 강화학습에서 음의 보상이 없는 경우 성능적으로 감소할 수 있기 때문이다. 또한, 행동 성공(S_{action})에서 성공했을 때 보상 값 +1을 주어진다. 이때 세부적으로 Scan 행동의 성공(S_{scan})을 판단했을 때 자신이 찾은 노드의 수(N)에 비례하여 보상 값을 획득한다. 그리고 목적으로 다가가기 위한 필수 노드를 침해(Compromised)하는 행동을 성공($S_{compromised}$) 한다면 보상 값 100을 획득하도록 구성되어 있다.

3.2 시뮬레이션 시나리오 설계

Unreal Engine 기반 사이버 전장의 구조를 완성하기 위해서는 시뮬레이션 시나리오를 제작하여 환경을 검증할 필요가 있다. 그림 4는 본 논문에서 실험을 진행한 네트워크 공격 시나리오이다. 네트워크의 전체적인 구조는 5개의 Subnet을 가지고 노드들의 역할은 PC와 서버에 따라 각각 OS, Process, Service 등 여러 조건을 달리하여 설계하였다.

설정된 공격 시나리오는 최적의 11개의 공격 시퀀스를 생성하는 것을 목표로 한다. 전체 시나리오의 시작은 공격자가 공격하는 네트워크의 내부에 존재하는 단말을 이미 점거한 상태에서 공격을 시작한다.

Subnet1에 존재하는 점거된 공격자 단말에서 Subnet2에 존재하는 노드(2,0) 서버에 접근하고 이후 Subnet4에 존재하는 노드(4,0) PC를 공격하여 Subnet5에 있는 노드(5,3) 서버의 접근 권한을 얻는다. 이후 서버에 접근하여 권한을 탈취하는 것으로 공격 시나리오는 종료된다.

본 논문에서 현재 시나리오는 노드(5,3)에서 관리자 권한을 취득하는 것을 목적으로 한다. 표 1은 시나리오에서 공격자의 Ground Truth 시퀀스를 표현한 값이며, 앞서 정리한 행동 공간에서의 번호와 그 번호에 해당하는 카테고리로 정리하여 공격이 어떤 과정으로 최종 목적에 도달하는 것인지 분석하게 되어 있다. 또한, 실험 검증을 위해 공격 시퀀스를 두 가지 영역으로 구분하였으며, 강화학습 모델링을 통해 학습된 에이전트가 Ground Truth가 어느 정도 수준으로 재현하는지에 대한 검증을 수행하였다.

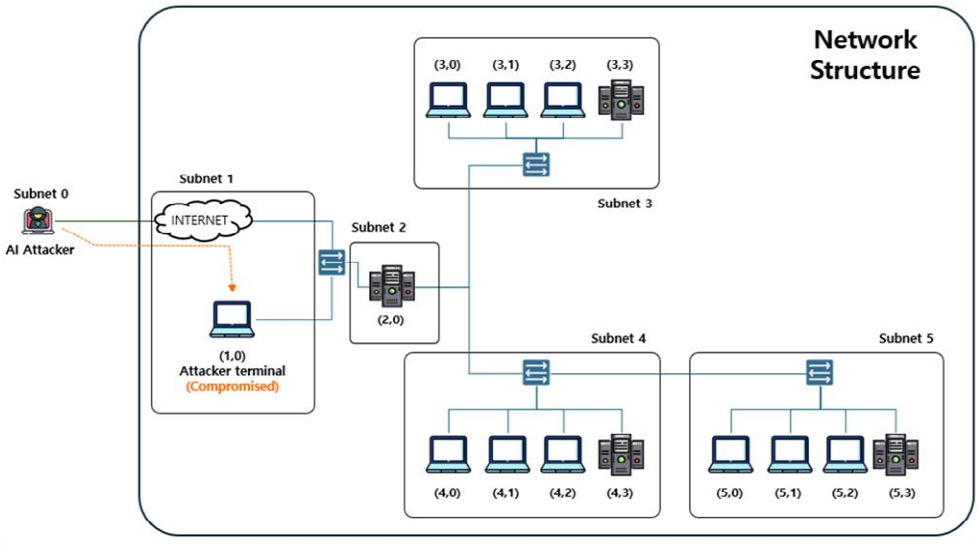

그림 4. 전장 시나리오 네트워크 구조
Fig. 4. Unreal based cyber-range simulation structure

표 1. 시나리오의 Ground Truth 설정
Table 1. Scenario's Ground Truth sets

Action ID	Target address	Action category
1	0	Reconnaissance
2	0	Reconnaissance
8	0	Initial access
0	1	Reconnaissance
3	1	Reconnaissance
7	1	Initial access
0	6	Reconnaissance
4	6	Reconnaissance
6	6	Initial access
5	13	Reconnaissance
12	13	Privilege escalation

IV. 실험 및 분석

본 논문에서 제작한 환경과 시나리오를 통해 학습된 강화학습 에이전트가 의도한 Ground Truth 값 생성을 확인하고 성능을 평가하여 환경에 대한 타당성을 검증한다.

4.1 실험 환경

설계된 사이버 전장 시뮬레이션에서 강화학습 모델을 학습시킬 때, POMDP 구조를 기반으로 학습을 진행하였다.

표 2. 실험 시나리오 환경 설정
Table 2. Experimental scenario environmnet setting

Node	OS	Service
(1,0)	Linux	Web
(2,0)	Windows	Web, Log4j
(3,0)	Windows	SMB
(3,1)	Windows	RDP
(3,2)	Windows	RDP
(3,3)	Linux	VNC
(4,0)	Windows	SMB
(4,1)	Windows	RDP
(4,2)	Windows	RDP
(4,3)	Linux	VNC
(5,0)	Windows	SMB
(5,1)	Windows	RDP
(5,2)	Windows	RDP
(5,3)	Linux	VNC

POMDP는 학습 에이전트가 모든 State 값을 Observation으로 사용하는 것이 아닌, 에이전트가 Observation 할 수 있는 정보만을 사용한다. 또한,

에이전트는 학습 과정에서 필요한 정보를 획득하고 획득한 정보를 기반으로 공격을 진행한다. 실험에서 사용한 시나리오 아키텍처의 세부 설정은 표 2와 같다. 표 2에서 나타난 환경의 State 정보는 일부의 정보 값을 나타내고 있다. 공격자는 사전에 설계된 공격 기술을 사용하여 각 노드의 OS 혹은 Service와 관련된 취약점을 찾기 위한 행동을 진행하고, 이후 Initial access나 exploit, Privilege Escalation과 같은 행동을 선택하여 목적을 달성하기 위한 정책을 생성해야 한다.

4.2 성능 검증

본 연구에서 제안하는 시뮬레이션 환경은 공격 도구와 대상 노드를 선택하는 이산적 행동 공간(Discrete action space)으로 구성되어 있다. 일반적으로 이산 행동 공간을 가지는 환경에서 행동의 가치를 기반으로 학습하는 DQN(Deep Q-Network)[22]를 먼저 실험에 적용하였다. 또한, 강화학습에서 학습 안정성이 뛰어나 다양한 분야에서 활용되는 PPO(Proximal Policy Optimization)[23], 연속적 또는 이산적인 환경 모두에게 준수한 성능을 가지는 SAC(Soft Actor-Critic)[24]을 이용하여 성능을 검증하고 결과를 분석하였다. 우선 간단한 학습으로 검증을 진행하기 위해 공격 에이전트의 초기 목적인 3개의 노드를 침해하는 것을 학습 목표로 설정하여 실험을 진행하였다. 이때 표 2의 14개의 노드 중 10개의 노드만 유효한 노드로써 사용된다. 학습 목표에 따른 에피소드-보상 결과 그래프는 그림 5와 같이 나타났다.

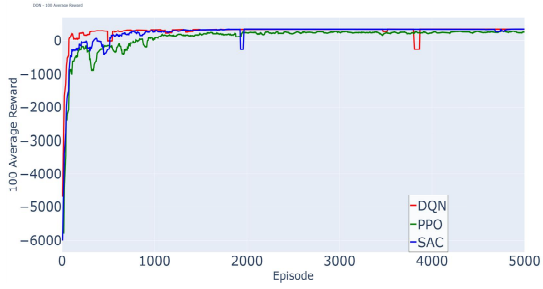

그림 5. 공격자 학습 그래프
Fig. 5. Attacker training graph

현재 보상 시스템과 최적의 시퀀스 9개를 기반으로 얻을 수 있는 최대의 보상 값은 350이다. SAC와 DQN은 성공적으로 최적의 시퀀스 9개를 생성함과 동시에 최대 보상 값 350을 달성하였지만, PPO의 경우 최종 평균 37개의 시퀀스로 최적의 시퀀스 생성에 도달하지 못하였다.

표 3은 DQN, SAC가 생성한 공격 시퀀스를 정리하고 있다. 사전에 정의한 시나리오는 Ground Truth의 값과 동일한 순서로 시퀀스를 생성하고 있다. 현재 상태 공간과 행동 공간은 이산적으로 제작되어 있으므로 DQN은 안정적으로 시퀀스를 생성하는 것을 확인할 수 있다. SAC의 경우 내부적으로 entropy라는 변수를 사용하여 행동의 선택 확률을 넓은 범위에서 탐색할 수 있으므로 성공적인 최적의 공격 시퀀스를 생성하고 있는 것을 확인할 수 있다.

표 3. DQN, SAC 공격 시퀀스 요약
Table 3. Summary of DQN and SAC attack sequence

Step	Action ID	Target_ addr	Reward
1	1	0	−1
2	2	0	−1
3	8	0	+98
4	0	1	+6
5	3	1	−1
6	7	1	+98
7	0	6	+2
8	4	6	−1
9	6	6	+150

반면 PPO가 생성한 공격 시퀀스는 표 4와 같다. PPO는 On-Policy 기반으로 현재 정책에서 수집된 정보를 토대로 학습을 진행한다. 이는 현재 정책을 업데이트할 때 현재 정책에서 얻은 보상과 행동, 그리고 관측 공간을 기반으로 학습한다. 따라서 초기 탐색 과정에서 음의 보상이 많이 축적하면 정책이 음의 방향으로 편향될 수 있으며, 그 결과 최적의 정책 생성에 도달할 수 없다. 더욱이 현재 제작한 시나리오 환경은 Action에 성공하더라도 Scan이나 침해 행동이 성공하지 못한다면 기본적으로 음의 보상을 받아 학습을 진행하게 되어 있다. 따라서 이러한 현상은 PPO의 정책 생성에 있어 악영향을 미치는 것으로 확인하였다.

표 4. PPO 공격 시퀀스 요약
Table 4. Summary of PPO attack sequence

Step	Action ID	Target_ addr	Reward
1	2	0	-3
2	2	0	-3
3	1	0	-1
4	8	0	-3
5	7	1	-3
6	8	0	-3
7	2	0	-1
8	8	0	+98
9	8	0	-3
10	2	0	-3
...	...	...	...
37	6	6	+150

위 실험을 통해 DQN과 SAC가 성공적으로 시나리오에서 의도한 Ground Truth와 동일한 결과값을 도출하는 것을 확인하였다. 이를 통해 공격 도구 및 시뮬레이션 환경 설정의 유효성을 검증하였으며, 이를 바탕으로 본 시나리오의 최종 목표인 권한 상승 단계를 추가하여 실험을 확장하였다. 공격 프로세스의 단계가 세분됨에 따라 에이전트가 탐색해야 할 상태-행동 공간(State-action space)이 확장되었고 에피소드 완수를 위한 조건이 까다로워짐에 따라 이전 단계 대비 학습 복잡도가 유의미하게 증가하였다. 이때 학습 복잡도는 탐색 과정에서 증가한 가짓수를 Log 함수를 사용하여 계산하였다. 수식 2에서 A는 Action의 수, N_{node}는 시나리오에서 사용할 수 있는 node의 수를 나타낸다. 이때의 조합은 계산 복잡도 C로 계산을 진행하였다.

$$\log(N_{complex}) = \log(|A| \times |N_{node}|)$$
$$C = \log(N_{complex}) \quad (2)$$

처음 실험을 진행한 노드를 침해하는 것을 목적으로 하는 공격자는 총 10가지의 노드를 Action으로 선택할 수 있으므로 복잡도를 계산한다면 3.60의 계산 복잡도를 가진다. 반면 권한 탈취의 목적이 추가된다면 3.89의 계산 복잡도를 가지게 된다. 계산 복잡도가 약 8.06%가 상승했으며, 이는 목적 달성이 더 어려워지는 것을 나타내고 있다. 학습 목표에 따른 에피소드-보상 결과 그래프는 그림 6과 같다. 최적의 공격 시퀀스의 수인 11을 생성하는 것으로 얻을 수 있는 최대 보상은 358이다. DQN과 SAC는 최종적으로 최적의 시퀀스와 함께 최대 보상을 얻는 것을 확인하였으나, PPO는 평균 256의 보상으로 최적의 시퀀스를 최종적으로 생성하지 못하였다.

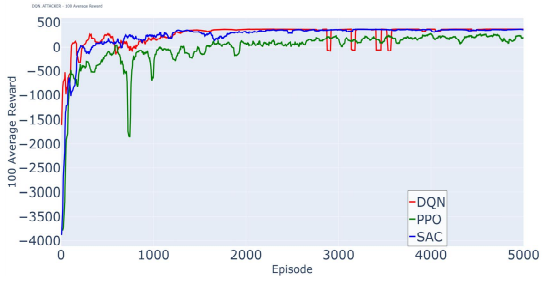

그림 6. 공격자 학습 그래프
Fig. 6. Attacker training graph

위 결과를 통해 DQN과 SAC는 성공적으로 최적 시퀀스에 도달하는 것을 확인하였으나, PPO는 성능이 크게 떨어졌음을 확인하였다.

V. 결론 및 향후 과제

본 논문에서 구축한 사이버 전장 시뮬레이션 환경은 정교한 네트워크 공격 시나리오 설계를 통해 실제 환경과 유사한 공격 프로세스를 재현하고 있다. 기존 연구들이 간소화된 공격 기법 및 상태 정보의 한계가 있어 이를 극복하기 위해 보다 상세한 파라미터 정의를 통해 실험 및 검증을 수행하였고 그 결과, 도출된 공격 시퀀스가 실제 네트워크 아키텍처 상에서의 공격 순서와 매우 유사함을 확인할 수 있었다. 또한 강화학습 알고리즘별 학습 특성에 따라 공격 성능과 수렴 양상이 달라짐을 확인하였다. DQN은 이산 행동 선택에 최적화되어 최적 공격 경로를 가장 효과적으로 형성하였지만, PPO는 학습 안정성은 높았으나 보상 체계의 페널티 영향에 따라 정책 수렴 성능이 상대적으로 낮았다. 추가로 상태 공간의 차원이 확장될수록 DQN의 학습 성능이 다소 감소하는 것을 확인하였으며, 이는 고차원 상태 정보에 따른 가치 함수 근사의 어려움에 따른 결과로 분석된다.

향후 본 연구의 확장으로, 공격 에이전트와 방어 에이전트가 상호 경쟁하는 멀티 에이전트 시스템을 설계할 계획이다. 이는 공격 에이전트가 방어 기제를 무력화하기 위한 새로운 공격 루트를 지속적으로 발굴하고, 방어 에이전트는 이에 적응하여 동적 위협을 차단하는 사이버 공방 최적화 시나리오 실증을 목표로 연구를 진행할 것이다.

References

[1] K. Ramezanpour, J. Jagannath, and A. Jagannath, "Security and Privacy vulnerabilities of 5G/6G and WiFi 6: Survey and Research Directions from a Coexistence Perspective", arXiv preprint arXiv:2206.14997, Feb. 2022. https://doi.org/10.48550/arXiv.2206.14997.

[2] C. Okafor, T. R. Schorlemmer, S. Torres-Arias, and J. C. Davis, "SoK: Analysis of Software Supply Chain Security by Establishing Secure Design Properties", arXiv preprint arXiv:2406.10109, Jun. 2024. https://doi.org/10.48550/arXiv.2406.10109.

[3] A. Quffa and S. S. Abu-Naser, "A Rule-Based Expert System for Cybersecurity Threat Detection: Evolution, Applications, and the Hybrid AI Paradigm", International Journal of Academic Engineering Research (IJAER), Vol. 9, No. 8, pp. 44-62, Aug. 2025. https://doi.org/10.13140/RG.2.2.21026.49606.

[4] M. W. A. Ashraf, A. R. Singh, A. Pandian, R. S. Rathore, M. Bajaj, and I. Zaitsev, "A hybrid approach using support vector machine rule-based system: detecting cyber threats in internet of things", Scientific Reports, Vol. 14, No. 27058, Nov. 2024. https://doi.org/10.1038/s41598-024-78976-1.

[5] F. Ceschin, M. Botacin, A. Bifet, B. Pfahringer, L. S. Oliveira, H. M. Gomes, and A. Gregio, "Machine Learning (In) Security: A Stream of Problems", ACM Digital Threats: Research and Practice, Vol. 5, No. 9, pp. 1-32, Mar. 2024. https://doi.org/10.1145/3617897.

[6] M. R. Naeem, R. Amin, M. Farhan, F. S. Alsubaei, E. Alsoami, and M. D. Zakaria, "Cyber security Enhancements with reinforcement learning: A zero-day vulnerability identification perspective", PLoS One, Vol. 20, No. 5, May 2025. https://doi.org/10.1371/journal.pone.0324595.

[7] S. Choi, S. Choi, C. Kim, H. Kwon, and S. Lee, "Development of a ROS 2-based Digital Twin Environment for Mobile Robots", The Journal of Korean Institute of Information Technology, Vol. 23, No. 11, pp. 101-108, 2025. https://doi.org/10.14801/jkiit.2025.23.11.101

[8] H, Choi, D. Woo, S. Yu, and J. Kim, "A Connected Framework for a Satellite/OSM Building Pipeline for Digital Twins in Unity", The Journal of Korean Institute of Information Technology, Vol. 23, No. 12, pp. 177-187, 2025. https://doi.org/10.14801/jkiit.2025.23.12.177

[9] T. Lee and W. Huh, "Planning and Implementation of a Healing Adventure Game ‘Memory Studio’ based on Unreal Engine 5", The Journal of The Institute of Internet, Broadcasting and Communication, Vol. 24, No. 6, pp. 245-251, 2024. https://doi.org/10.7236/JIIBC.2024.24.6.245

[10] D. Lopes-Montero, J. L. Alvarez-Aldana, A. Morales-Martines, M. Gil-Lopes, and J. M. A. Garcia, "Reinforcement Learning For Automated Cybersecurity Penetration Testing", arXiv preprint arXiv:2507.02969, Jun. 2025. https://doi.org/10.48550/arXiv.2507.02969.

[11] U. G. de Sousa, "LogGuardQ: A Cognitive-Enhanced Reinforcement Learning Framework for Cybersecurity Anomaly Detection in Security Logs", arXiv preprint arXiv:2509.10511, Sep. 2025. https://doi.org/10.48550/arXiv.2509.10511.

[12] Z. Hu, R. Beuran, and Y. Tan, "Automated Penetration Testing Using Deep Reinforcement Learning", IEEE European Symposium on Security

and Privacy Workshops (EuroS&PW), Genoa, Italy, pp. 2-10, Oct. 2020. https://doi.org/10.1109/EuroSPW51379.2020.00010.

[13] F. M. Zennaro and L. Erdodi, "Modelling penetration testing with reinforcement learning using capture-the-flag challenges: Trade-offs between model-free learning and a priori knowledge", IET Information Security, Vol. 17, No. 3, pp. 441-447, Jan. 2023. https://doi.org/10.1049/ise2.12107.

[14] H. Emerson, L. Bates, C. Hicks, and V. Mavroudis, "CybORG++: An Enhanced Gym for the Development of Autonomous Cyber Agents", arXiv preprint arXiv:2410.16324, Oct. 2024. https://doi.org/10.48550/arXiv.2410.16324.

[15] S. Zhou, J. Liu, Y. Lu, J. Yang, Y. Zhang, and J. Chen, "Mind the Gap: Towards Generalizable Autonomous Penetration Testing via Domain Randomization and Meta-Reinforcement Learning", Frontiers of Information Technology & Electronic Engineering, Vol. 26, pp. 2511-2528, Dec. 2025. https://doi.org/10.48550/arXiv.2412.04078.

[16] H. Suk and M. S. Kim, "Enhancing Reinforcement Learning-Based Agent Learning Process and Restructuring Reward Using Cyber Battle Simulator", The Korea Society of Computer and Information, Vol. 32, No. 2, pp. 41-44, Jul. 2024.

[17] T. Kunz, C. Fisher, J. L. Novara-Gsell, C. Nguyen, and L. Li, "A Multiagent CyberBattleSim for RL Cyber Operation Agents", arXiv preprint arXiv:2304.11052, Apr. 2023. https://doi.org/10.48550/arXiv.2304.11052.

[18] B. S. Kim, J. H. Kim, and M. S. Kim, "A Study of Reinforcement Learning-based Cyber Attack Prediction using Network Attack Simulator (NASim)", Journal of the Semiconductor & Display Tech., Vol. 22, No. 3, pp. 112-118, Sep. 2023.

[19] J. Janisch, T. Pevny, and V. Lisy, "NASimEmu: Network Attack Simulator & Emulator for Training Agents Generalizing to Novel Scenarios", European Symposium on Research in Computer Security, Vol. 14399, pp. 589-608, Mar. 2024. https://doi.org/10.1007/978-3-031-54129-2_35.

[20] Network Attack Simulation, https://github.com/Jjschwartz/NetworkAttackSimulator. [accessed: May 15, 2021]

[21] MITRE ATT&CK, https://attack.mitre.org/. [accessed: Jun. 25, 2023]

[22] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, and D. Hassabis, "Human-level control through deep reinforcement learning", Nature, Vol. 518, No. 7540, pp. 529-533, Feb. 2015. https://doi.org/10.1038/nature14236.

[23] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal Policy Optimization Algorithms", arXiv preprint arXiv:1707.06347, Jul. 2017. https://doi.org/10.48550/arXiv.1707.06347.

[24] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor", Proc. 35th Int. Conf. on Mach. Learn. (ICML), Stockholm, Sweden, Proc. Mach. Learn. Res. (PMLR), Vol. 80, pp. 1861-1870, Jul. 2018.

저자소개

김 정 현 (Jung-Hyun Kim)

2023년 2월 : 상명대학교 휴먼지능로봇공학과(공학사)
2025년 2월 : 상명대학원 전자정보시스템공학과(공학석사)
2025년 3월 ~ 현재 : 상명대학원 전자정보시스템공학과 박사과정
관심분야 : 사이버 보안, 강화학습, 시뮬레이션 개발

김 민 서 (Min-Seo Kim)

2022년 3월 ~ 현재 : 상명대학교 휴먼지능로봇공학과 학사과정
관심분야 : 기계학습, LLM, 강화학습

김 민 석 (Min-Suk Kim)

2010년 5월 : University of Pittsburgh 정보통신(석사)
2016년 9월 : University of Massachusetts Lowell 컴퓨터공학(박사)
2016년 11월 ~ 2020년 3월 : 한국전자통신연구원 선임연구원
2020년 3월 ~현재 : 상명대학교 휴먼지능로봇공학과 조교수
관심분야 : 기계학습, 강화학습, 딥러닝, 엣지컴퓨팅, 사이버 보안