강화 학습을 이용한 웹 애플리케이션 취약점 탐지 현황*

이소영O 손수엘
한국과학기술원
soyoungleell@kaist.ac.kr, sl.son@kaist.ac.kr

A Survey on Web Application Vulnerability Detection with Reinforcement Learning

Soyoung LeeO Sooel Son
KAIST

요 약

현재 웹 애플리케이션 취약점 탐지 도구들은 대부분 사전 공격을 사용하며 이는 복잡한 공격 구문을 요구하는 취약점의 경우 탐지하기까지 오랜 시간이 걸리거나 탐지하지 못하는 경우가 발생한다. 이 문제를 해결하기 위해 공격 구문 생성에 마르코프 결정 과정 및 강화 학습을 적용한 연구들이 존재한다. 본 논문에서는 기존 연구 결과를 통해 한계점을 분석하며 그에 대한 해결 방안을 고찰해보고자 한다.

1. 서 론

현재 웹 애플리케이션의 취약점을 탐지하기 위해 웹 애플리케이션에 공격 구문을 삽입하는 침투(Penetration) 테스트가 행해지고 있다. 대부분의 오픈 소스 및 상용 웹 애플리케이션 취약점 탐지 도구들은 이미 정해진 공격 구문을 이용하는 사전 공격(Dictionary Attack)을 이용한다. 사전 공격은 간단한 공격 구문으로 탐지되는 취약점의 경우 단시간 안에 찾을 수 있다는 장점이 있다. 하지만 복잡한 공격 구문을 요구하는 취약점의 경우 해당 구문까지 도달하기 위해 모든 공격 구문을 시도해야 하므로 오랜 시간이 걸릴 가능성이 있다. 이런 문제를 해결하기 위해 강화 학습(Reinforcement Learning)을 적용하는 방안이 고안되었다. 웹 애플리케이션 취약점 탐지의 여러 부분에 강화 학습이 적용될 수 있다. 본 논문에서는 효과적인 공격 구문 생성을 위해 강화학습을 적용한 연구들의 방법과 그들의 한계점 및 해결방안에 대하여 알아보고자 한다.

2. 강화 학습

강화학습이란 마르코프 결정 과정(Markov Decision Process) 문제를 풀기 위한 기계 학습 기법 중 하나이다. 마르코프 결정 과정은 환경(environment)을 나타내는 상태(state)들의 집합, 각 상태 간의 천이 확률 (transition probability), 상태에 따라 에이전트가 수행하는 행동(action), 그리고 환경이 선택된 행동에 대해 에이전트에게 주어지는 보상 (reward function)으로 이루어진다[1].

MDP에서 에이전트는 정책(policy)에 따라 행동을 결정하게 되며, 강화학습 알고리즘은 주어지는 보상이 최대가 되는 방향으로 정책을 학습시킨다. 기본적인 강화학습 알고리즘에는 Q-Learning이 있다. 각 상태-행동 쌍에 대해 주어진 보상과 기대 보상 값(Q-value)을 저장하여 누적된 값으로 주어진 상태에 대해 알맞은 행동을 제시한다. Q-value를 저장하는 방식에 따라 간단한 환경의 경우 표를 사용하는 tabular Q-learning을 이용한다. 표를 인공신경망으로 대체한 Deep Q-Network(DQN)[2]을 DeepMind가 발표하였으며 현재는 Actor-Critic 방식을 적용한 Asynchronous Advanced Actor Critic(A3C)[3]과 Proximal Policy Optimization (PPO)[4] 알고리즘 등 발전된 알고리즘이 발표되고 있다.

2.1. 강화 학습의 웹 취약점 탐지 적용

웹 애플리케이션에 존재하는 취약점을 탐지할 수 있는 공격 구문을 생성하는 것은 기존의 지도학습, 비지도 학습으로는 해결하기 어려운 특성을 가진다. 이는 취약점에 대한 웹 페이지 정보와 그에 알맞은 공격 구문을 라벨링 하는 것이 어렵다는 점과 많은 양의 데이터 세트을 구축하는 것이 어렵다는 단점이 있기 때문이다. 이와 달리 강화학습은 정해진 데이터 세트가 필요하지 않기 때문에 웹 취약점 탐지에 적용하기에 알맞은 특성을 가진다.

*이 논문은 2021년도 정부(과학기술정보통신부)의 재원으로 정보통신기획평가원의 지원을 받아 수행된 연구임
(No. 2019-0-01343, 융합보안핵심인재양성)

3. 강화 학습 적용 사례

강화 학습은 웹 해킹 및 방어의 여러 부분에 적용될 수 있다. 본 장에서는 침투 테스트의 성능을 높이기 위한 강화 학습 적용의 접근 방식을 알아보고자 한다.

3.1. 강화 학습 문제로서의 웹 해킹

2019년 Anders 등은 침투 테스트의 성능을 높이기 위해 침투 테스트 환경을 마르코프 결정 과정으로 나타내는 것을 제안하였다[5]. 이는 실제 공격을 수행할 공격 구문을 생성하는 것은 아니며 침투 테스트 도구 사용자가 취약점을 찾을 수 있도록 취약점이 존재할 것 같은 지점으로 유도하고, 비슷한 취약점을 다른 해커들이 어떤 방식으로 탐지했는지를 알려주는 역할을 한다. 환경은 HackerOne에서 제공하는 취약점 리포트들이며 상태는 종류별 취약점의 존재 여부, 행동은 취약점이 존재하는지 질문하는 것으로 정의하였다. 현재 구현된 도구는 실제로 취약점을 자동으로 탐지할 수 있는 것은 아니며 주어진 리포트를 통해서 사용자에게 취약점을 탐지하는 데 도움을 주는 정도의 기능을 가진다.

2020년 László 등은 침투 테스트 문제를 강화학습 문제로 해결하기 위하여 Capture the Flag(CTF) 문제로 정의하는 방법을 제안하였다[6]. CTF의 경우 상용 웹 애플리케이션보다 명확한 목적과 정확한 종료 조건을 가지고 있다. 취약점을 가진 것이 당연하며 취약점을 찾거나 시간제한이 있어 종료조건이 존재한다. CTF 문제를 게임으로 치환하면 이는 기존에 강화학습으로 해결하고자 했던 게임 문제와 같은 특성을 가지게 된다. 이를 이용하면 마르코프 결정 과정 문제로 정의하고 강화학습 알고리즘을 적용하는 것을 효율적으로 할 수 있다는 장점이 있다.

3.2. 공격 구문 생성을 위한 강화 학습의 적용

2020년 Xianbo 등은 웹 방화벽 우회를 위한 공격 구문 생성에 강화 학습을 적용하였다[7]. 웹 방화벽은 완벽하게 모든 악성 코드를 막아낼 수는 없으며 이를 우회하기 위한 공격 구문을 생성하는 것이 가능하다. 웹 취약점 중 SQL 삽입 취약점에 중점을 두었으며 환경은 오픈 소스 웹 방화벽인 ModSecurity와 WAF-Brain을 선정하였다. 상태는 SQL 삽입 공격에 사용되는 공격 구문의 각 요소로 나타내었으며 행동은 각 요소를 바꾸는 변화 규칙으로 정의하였다. 보상은 방화벽에서 돌려주는 공격 구문이 통과하였는지의 여부 및 신뢰 구간 값을 사용하였다. 또한, 공격이 짧은 시간 안에 성공하게 하도록 시간이 지남에 따라 적은 보상을 주도록 하였다. 실험 결과, 학습된 에이전트는 WAF-Brain에서 최대 성공률 35%, ModSecurity에서 최대 성공률 8.7%를 보였으며 이는 무작위로 공격 구문을 선택하는 에이전트의 성공률 각 최대 10%, 0.2%보다 좋은 결과를 보였다.

2021년 László 등은 Q-Learning을 이용한 SQL 삽입 취약점 탐지 방법을 고안하였다[8]. 먼저 SQL 삽입 공격 타겟을 단순화하기 위해 이를 CTF 문제로 타겟을 정의하였다. 환경은 CTF 문제 형식의 웹 애플리케이션이다. 행동은 기본 SQL 삽입 공격 구문에서 각 요소를 바꾸는 것으로 정의하였다. 상태는 각 행동의 사용 여부로 나타내었다. 보상은 CTF 문제에서 Flag를 찾으면 10, 수행한 행동이 성공의 결과를 나타내지 못한 경우에는 -1의 보상을 주었다. 학습 결과 에이전트는 평균 5.197의 보상을 얻을 수 있도록 발전하였다.

2021년 Caturano 등은 강화 학습을 이용한 Reflected Cross Site Scripting 취약점 탐지 도구인 Suggester를 고안하였다[9]. 환경은 Reflected Cross Site Scripting 취약점을 가진 벤치마크로 정의하였다. 상태는 공격 구문의 기본 구조를 정의한 후 각 요소가 타겟에서 나타났는지, HTML 문법에 맞도록 형성되었는지의 정보를 사용하였다. 행동은 공격 구문의 각 요소를 변화시키는 것으로 정의하였으며 한 행동은 하나의 요소만을 변화시킬 수 있다. 보상은 타겟에 공격 구문이 삽입되었으면 양수의 보상을, 행동에 의해 공격 구문이 변화하지 않았거나 취약점 탐지에 있어 좋은 방향으로 변화하지 않았으면 음수의 보상을 주는 것으로 정의하였다.

Suggester는 침투 테스트 도구가 아닌 사용자에게 어떤 행동을 할 것인지 제안해 주는 도구이며 모델이 행동을 선택하여 공격 구문을 생성하면 사용자가 공격구문을 HTTP 요청으로 변환하여 타겟으로 전송하고 사용자가 타겟에서 나타나는 공격 구문으로 상태 정보를 수집한다. 학습은 Web Application

표 1 관련 논문 비교

	공격 대상	목적	타겟 취약점	알고리즘	자동화	학습 데이터	테스트 데이터
Anders, 2019[5]	웹 애플리케이션	탐지 방법 제안	전체	Q-Learning	Fully-automated	HackerOne 취약점 리포트	Case study
László , 2020[6]	CTF	웹 해킹 모델링	전체	MDP	X	X	X
Xianbo, 2020[7]	웹 방화벽	공격 구문 생성	SQL Injection	PPO	Fully-automated	ModSecurity, WAF-Brain	ModSecurity, WAF-Brain
László , 2021[8]	CTF	공격 구문 생성	SQL Injection	Q-Learning	Fully-automated	CTF	X
Caturano, 2021[9]	웹 애플리케이션	공격 구문 생성	Reflected XSS	Q-Learning	Human in-the-loop	WAVSEP	Yahoo webseclab

Vulnerability Scanner Evaluation Project (WAVSEP)를 타겟으로 진행하였다. 모델의 성능 검증은 Yahoo의 webseclab과 OWASP의 웹 벤치마크를 이용하여 진행하였다. 결과는 Yahoo webseclab에 대해 true positive rate 0.70, false positive rate 0.00으로 비교군 6개의 오픈 소스 침투 테스트 도구 중 4개보다 높은 true positive rate를 보였으며 5개보다 낮은 false positive rate를 보였다.

4. 한계점 및 제안

표 1은 본 논문에서 소개한 각 연구를 비교한 결과를 보여준다. Anders, 2019와 László , 2020는 침투 테스트 환경을 마르코프 결정 과정으로 나타내어 모델링 하는 방법을 제안하였고 표 하단의 연구들은 강화 학습 기법을 이용하여 문제를 해결하는 형식을 취하며 직접 공격 구문을 만들어 내는 방법을 고안하였다. 이때 마르코프 결정 과정의 각 요소를 결정하는 것은 절대적인 것이 아니며 각 취약점 및 타겟으로 하는 환경에 따라 달라진다.

현재 소개된 연구들에서는 강화 학습을 적용한 자동화된 침투 테스트를 하는 것이 불가능하다. 강화 학습 알고리즘을 적용하기 위해 환경을 구축하고 각 요소를 정의하였지만 이를 충족시키기 위해 인간이 직접 정보를 수집해야 하는 경우가 있었다.

또한, 강화 학습 적용의 편의성을 위해 취약점이 존재하는 환경을 단순화하는 경우가 많았으며 이는 실제 상용 애플리케이션 등에 학습된 모델을 사용하여 취약점을 탐지하는 것은 불가능하다는 한계점이 있다. 강화 학습의 환경을 실제 공격 시나리오에 맞추어 여러 가지 경우의 수를 고려할 수 있도록 해야 한다. 이에 더하여 실제 웹 애플리케이션에 적용할 수 있는지에 대한 여부는 실험하지 않은 경우가 많았고 이는 많은 침투테스트 도구 사용자들의 사용성을 충족시키는 어려울 것으로 보인다.

마지막으로 공격 구문 생성연구의 경우 각 모델이 학습하는 것은 한가지의 취약점에 대한 공격 구문 생성을 위한 것이며 한 애플리케이션에 존재하는 여러 종류의 취약점을 탐지하는 것은 불가능하다. 취약점을 찾고자 하는 파라미터에서 어떤 웹 공격이 가능할지 판단하는 요소가 있다면 발전된 웹 취약점 탐지가 가능할 것이다.

이러한 문제점을 해결하기 위해 웹 애플리케이션 환경 분석을 통한 상태 정보의 세분화 및 상태 요소를 추가하는 방법을 취할 수 있다. 에이전트에게 공격을 실행할 부분에 대한 정보를 제공해주면 행동을 선택할 때의 부정확성이 줄어들 것이다. 예를 들어, 현재 환경은 공격 구문의 정보 및 타겟에 삽입 성공 여부 등을 고려했다면, 발전된 상태 정보에서는 공격 구문이 삽입된 곳이 HTML 문서의 어떤 요소인지 그리고 실제로 삽입된 공격 구문이 실행되었는지의 여부 등을 자동으로 분석하여 에이전트가 알 수 있도록 하는 것이 도움이 될 것이다.

5. 결론

효과적인 웹 애플리케이션 취약점 탐지를 위해 해결해야 하는 문제를 마르코프 결정 과정으로 변환, 강화 학습 알고리즘을 적용하여 학습시켜 기존의 취약점 탐지 도구보다 좋은 성과를 나타낸 연구들이 존재한다. 하지만 강화 학습 적용을 위해 한정된 환경 안에서 수행한 경우가 많았으며 상용 애플리케이션을 타겟으로 취약점 탐지를 하기 위해서는 발전된 공격 구문 생성 및 강화 학습 타겟 환경의 일반화가 필요할 것으로 보인다.

[참고문헌]

[1] M. L. Puterman, “Chapter 8 Markov decision processes,” in *Stochastic Models*, vol. 2, Elsevier, pp. 331–434, 1990.

[2] V. Mnih *et al.*, “Playing atari with deep reinforcement learning,” *arXiv preprint arXiv:1312.5602*, 2013.

[3] V. Mnih *et al.*, “Asynchronous methods for deep reinforcement learning,” *International conference on machine learning*, pp. 1928–1937, 2016.

[4] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, “Proximal Policy Optimization Algorithms,” *CoRR*, vol. abs/1707.06347, 2017,

[5] A. Pettersson and O. Fjordefalk, “Using Markov Decision Processes and Reinforcement Learning to Guide Penetration Testers in the Search for Web Vulnerabilities,” 2019.

[6] L. Erdodi and F. M. Zennaro, “The Agent Web Model - Modelling web hacking for reinforcement learning,” 2020.

[7] X. Wang and H. Hu, “Evading Web Application Firewalls with Reinforcement Learning,” 2020.

[8] L. Erdodi, Å. Å. Sommervoll, and F. M. Zennaro, “Simulating SQL Injection Vulnerability Exploitation Using Q-Learning Reinforcement Learning Agents,” 2021.

[9] F. Caturano, G. Perrone, and S. P. Romano, “Discovering reflected cross-site scripting vulnerabilities using a multiobjective reinforcement learning environment,” *Computers & Security*, vol. 103, p. 102204, 2021.