딥러닝 기반의 코드 취약점 탐지 모델의 적대적 공격

정은[1], 김형식[2]
[1]성균관대학교 전자전기컴퓨터공학과 석사과정
[2]성균관대학교 소프트웨어학과 교수

wjddms0926@skku.edu, hyoung@skku.edu

Adversarial Attack against Deep Learning Based Vulnerability Detection

Eun Jung[1], Hyoungshick Kim[2],
[1]Dept. of Electrical and Computer Engineering, Sungkyunkwan University
[2]Dept. of Software, Sungkyunkwan University

요 약

소프트웨어 보안의 근본적인 문제인 보안 취약점을 해결하기 위해 노력한 결과, 딥러닝 기반의 코드 취약점 탐지 모델은 취약점 탐지에서 높은 탐지 정확도를 보여주고 있다. 하지만, 딥러닝 모델은 작은 변형에 민감하므로 적대적 공격에 취약하다. 딥러닝 기반 코드 취약점 탐지 모델에 대한 적대적 공격 방법을 제안한다.

1 서론

최근 ChatGPT 와 같은 대규모 언어 모델(Large Language Model, LLM)의 발전은 AI 기술 기반 코딩 보조 도구를 통한 소프트웨어 개발 자동화를 촉진했고 개발자의 효율성을 크게 향상했다. 하지만 이러한 코드 생성 모델에서 발생하는 코드 취약점은 여전히 심각한 문제로 남아있고, 이는 소스 코드 내의 민감한 정보가 노출되는 상황을 의미한다. 이런 피해를 줄이기 위해 많은 연구자가 노력을 기울인 결과 딥러닝 기반 취약점 탐지 모델은 기존 방식보다 빠르고 효과적으로 취약점을 식별할 수 있다.

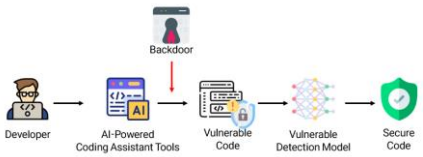
그러나 이러한 딥러닝 모델은 적대적 공격에 취약하다. 적대적 공격은 데이터에 눈에 띄지 않는 노이즈를 추가하여 모델을 속여 잘못된 예측을 하도록 만드는 공격 방법이다. 이러한 공격은 모델의 보안 취약성을 드러내며, 동시에 강건성 높은 모델을 개발하도록 도울 수 있다.

딥러닝 기반의 코드 취약점 탐지 모델의 취약성을 분석하기 위해 블랙박스 환경의 적대적 공격 방법을 제안한다. 취약점 탐지 모델은 코드의 구문적 및 의미적 특성을 분석하여 취약점을 식별하기 때문에, 입력 데이터의 작은 변형에 민감하게 반응할 수 있다. 본 연구를 통해 기존의 보안 시스템을 우회하고 실제 위험을 효과적으로 시뮬레이션할 수 있다. 기여하는 점은 다음과 같다.

- 딥러닝 기반의 코드 취약점 탐지 모델에 대한 새로운 적대적 공격 방법을 제안한다.
- 코드 취약점 탐지 모델의 강건함을 평가한다.

2 공격 시나리오

일반적으로 적대적 공격은 블랙박스 및 화이트 박스에서 이뤄진다. 블랙박스 환경에서의 적대적 공격을 중점으로 다룬다. 이때 공격자는 모델의 내부 구조, 가중치 등에 대해 알 수 없고, 입력과 모델의 예측 출력값만 알 수 있다.



[그림 1] 취약점 탐지 모델의 백도어 공격

많은 개발자는 코딩 보조 도구를 사용한다. 그러나 생성된 코드는 보안 취약점이 존재할 가능성이 있어 취약점 탐지 모델을 사용한다. 공격자는 이런 시스템을 악용하여 탐지 모델이 식별하지 못하도록 적대적 코드를 생성하는 백도어를 사용해 보안 시스템을 교묘하게 우회할 수 있다.

3 적대적 공격 방법

이러한 위협 사나리오에 대응하여, 취약점 탐지 모델을 대상으로 한 적대적 공격 방법을 개발한다. 이 방법은 두 단계로 구성되어 있다. i) 대상 모델에 대한 중요한 토큰 탐색 ii) 해당 토큰을 대체하여 적대적 코드 생성

3.1 중요한 토큰 탐색

원본 코드의 구조와 기능을 최대한 유지하면서 적대적 예제를 생성하기 위해, 입력 코드의 특정 토큰만 변형하고자 한다.

Step 1: 중요한 토큰 탐색 원본 코드를 토큰화하여 각 토큰을 하나씩 순차적으로 [MASK] 처리한다. 이렇게 [MASK] 처리된 토큰을 포함한 코드는 CNN 기반의 취약점 탐지 모델인 VulCNN 에 입력된다. [MASK] 처리된 토큰이 포함된 코드와 원본 코드의 출력 확률값을 비교하여 두 코드 간의 확률 차이를 측정한다. 확률의 차이가 클수록 해당 토큰은 모델 예측에 큰 영향을 미치는 토큰이라 판단되며, 해당 토큰을 '중요한 토큰'이라 정의한다.

Step 2: 중요한 토큰 선택 중요한 토큰 선택을 위해 중요도 점수를 계산한다. 중요도 점수 I_w는 다음과 같이 정의된다. $I_{w_i} = o_y(S) - o_y(S_{\setminus w_i})$ 여기서, S 는 원본 코드를 의미하고,$s_{\setminus w_i}$는 i번째 토큰을 [MASK] 로 대체한 코드이다. $s_{\setminus w_i}$는 다음과 같이 정의한다. $s_{\setminus w_i} = [w_0, \dots, w_{i-1}, [MASK], w_{i+1}, \dots]$ 각 토큰에 대한 중요도 점수를 내림차순으로 정렬한 후, 중요도가 가장 높은 상위 k 개의 토큰을 선택한다.

3.2 CodeBERT 를 사용한 토큰 생성

중요한 토큰을 식별한 후, 해당 위치에 들어갈 새로운 토큰을 생성하기 위해 CodeBERT[1] 알고리즘을 사용한다. CodeBERT 는 자연어 처리와 소스 코드 분석을 모두 지원하는 알고리즘으로, 두 가지 주요한 과정을 통해 학습된다: 코드의 빈칸을 채우는 MLM(Masked Language Modeling)과 코드의 교체 여부를 예측하는 RTD(Replaced Token Detection). 이를 통해 CodeBERT 는 코드의 맥락을 이해하고 해당 문맥에 적합한 새로운 토큰을 예측할 수 있다. 다음과 같이 생성된 토큰은 중요한 토큰의 위치에 삽입되며, 코드의 기본 구조를 유지하며 적대적 변형을 삽입할 수 있다. 이렇게 변형된 코드는 모델이 예측을 잘못하도록 유도하고, 효과적으로 기존의 모델을 우회할 수 있다.

4 실험

4.1 데이터 세트

성능 비교를 위해 VulCNN[2] 에서 제공하는 공개 소스 코드 데이터 세트를 사용하여 실험을 진행했다. 이 데이터 세트는 C/C++ 언어로 작성된 총 33,360 개의 함수를 포함하고 있으며 그 중 12,303 개는 취약한 코드이고, 21,057 개는 취약하지 않은 코드이다.

4.2 타겟 모델

타겟 모델은 CNN 구조를 기반으로 하며, 이는 16 개의 필터와 4 개의 단일 커널을 포함하는 합성곱 계층을 사용한다. Adam 최적화기를 사용하며 학습률은 0.001, 배치 크기는 32 로 설정되어 있고 100 에포크 동안 훈련되었다.

4.3 실험 환경

실험은 NVIDIA v100 GPU 에서 진행했고 하나의 적대적 코드를 생성하는데 30 분 이상의 시간이 필요하다. 이러한 시간적 제약을 고려하여, 전체 데이터 세트에서 무작위로 400 개의 코드 샘플을 추출하고 이를 대상으로 실험을 진행했다.

4.4 실험 결과

취약점 탐지 모델의 성능 지표로 자주 사용되는 정확도를 사용하였고 결과는 다음과 같다. 원본 코드에 대한 VulCNN 의 정확도는 66.7% 이고, 적대적 공격에 대한 정확도는 0% 으로 공격 성공률은 66.7%이다.

5 결론 및 향후 연구

CodeBERT 모델을 활용하여 VulCNN 에서의 적대적 공격 방법을 제안했다. 해당 공격 방법으로 생성된 데이터 세트는 타겟 모델에서 효과적임을 입증했다. 그러나 현재의 적대적 예제 생성 방법을 실제 컴파일 가능 여부나 문법적 정확성에 대해 고려하지 않았다. 향후 연구에서는 이러한 측면을 포함하여 더 현실적이고 실용적인 적대적 예제를 개발하는 것을 목표로 한다. 이를 통해 VulCNN 의 강건성을 더욱 향상하게 할 수 있을 것으로 기대한다.딥러닝 기반 보안 시스템의 안전성과 신뢰성을 강화하는 데 기여할 수 있다.

사사문구

이 논문은 2024 년도 정부(과학기술정보통신부)의 재원으로 정보통신기획평가원의 지원 (RS-2023-00229400, 안전한 메타버스 환경을 위한 사용자 인증 및 프라이버시 보호 기술 개발)과 2024 년 정부(과학기술정보통신부)의 재원으로 정보통신기획평가원의 지원(No. 2022-0-00995, 자연어로 기술된 요구사항에서 전문 개발자 수준의 고품질 코드를 자동 생성하는 기술)과 2024 년도 정부(개인정보보호위원회)의 재원으로 한국인터넷진흥원의 지원을 받아 수행된 연구(No. 2024-0960, 브라우저상 수집되는 정보주체의 온라인 행태정보 탐지 및 자기 통제기술 개발)와 2024 년도 정부(과학기술정보통신부)의 재원으로 정보통신기획평가원의 지원을 받아 수행된 연구임 (No. 2022-0-01199, 융합보안핵심인재양성)

참고문헌

[1] Feng, Zhangyin, et al. "Codebert: A pre-trained model for programming and natural language" arXiv preprint arXiv: 2022.08155(2020)

[2] Wu, Yueming, et al. "Vulcnn: An image-inspired scalable vulnerability detection system." Proceedings of the 44th International Conference on Software Engineering. 2022.