2023년 봄학기
캡스톤 축제
[공학 프로젝트 기획]

Bytecode Pre-trained 모델 기반 보안 취약점 자동 수정 기법

지도교수 : 남재창 교수님

팀원 : 배재호 김대석 장주영

I. Background

처음부터 결함 없는 프로그램을 만드는 것은 불가능에 가깝다. 실제로 현업에서는 소프트웨어를 배포하기 전에 여러 단계에서 분석 및 탐색 도구의 도움을 받아 소프트웨어의 결함을 확인한 다음 배포한다.

- **여전히 존재하는 보안 취약점**
  - 이러한 과정을 거친 후 배포된 소프트웨어더라도 무조건적으로 안전하지는 않다. 2021년 대두된 Log4j 보안 문제는 상용화된 역사가 오랜된 프로그램에도 보안 취약점이 존재함을 시사하였다. 그에 따른 보안 취약점 탐색 및 수정에 대한 새로운 방향의 후속 연구의 필요성이 강조되었다. 이를 위해 다양한 접근 방식을 활용한 연구가 요구된다고 판단하였다. 그 중 바이트코드를 기반으로 하는 보안 취약점 검출법 연구에 초점을 뒀다.

- **왜 바이트코드인가?**
  - Java는 바이트코드를 통하여 플랫폼 독립성과 뛰어난 이식성을 보장.
  - 기계어에 가까운 바이트코드의 특성상 바이트코드를 활용한 연구는 많이 이루어지지 않음.
  - 최근 Java 바이트코드에도 naturalness[1]가 있음이 확인됨. 이는 Pre-trained model 제작의 가능성을 시사함.

[1] Choi, Yoonho, and Jaechang Nam. "On the Naturalness of Bytecode Instructions." (2022).

II. Problem Definition

- **문제 제기**
  - Java 계열 프로젝트는 배포 단계 혹은 외부 라이브러리 참조 단계에서 바이트코드만을 활용하는 경우가 많다.
  - 기존의 소스 코드만을 사용한 보안 취약점 검출 기법을 사용할 수 없는 환경이 주어졌을 때, 바이트코드 기반 보안 취약점 검출 기법이 필요하다.
- **당면한 문제**
  - 보안취약점 검출 혹은 모델 학습 기반 연구에서도 Java 바이트코드를 활용한 사례는 드물다.
  - 따라서 Java 바이트코드로 구성된 Test Suites가 존재하지 않는다.
  - 바이트코드에 적합한 학습모델 역시 구현되거나 연구된 바가 없다.
- **목표**
  - 따라서 Java 바이트코드로 만들어진 보안취약점을 판단할 수 있는 Test Suites를 만들어야 한다.
  - Common Weakness Enumeration(CWE)을 기반으로한 dataset 생성에 초점을 둔다.
  - 생성한 Test Suites를 기반으로 pre-trained 가능한 model을 구현한다.
  - 구현한 모델의 성능을 평가하기 위해 비교할 기존의 보안 취약점 검출 도구를 탐색한다.

III. Related Works

1. 바이트코드 기반 보안취약점 탐색 연구

1.1. Locating SQL injection (Jackson et al. SoutheastCon'2018) [2]
1.2. BERT - WebShell Detection (Pu, Ao, et al' 2022)[3]
1.3. Elysium (Torres et al.RAID'2023)[4]

2. Pre-trained Model을 활용한 보안 취약점 검출 기법

2.1. Learning Binary Code with Deep Learning to Detect Software Weakness (Lee et al. ICONI'2017)[5]

3. Dataset 구축

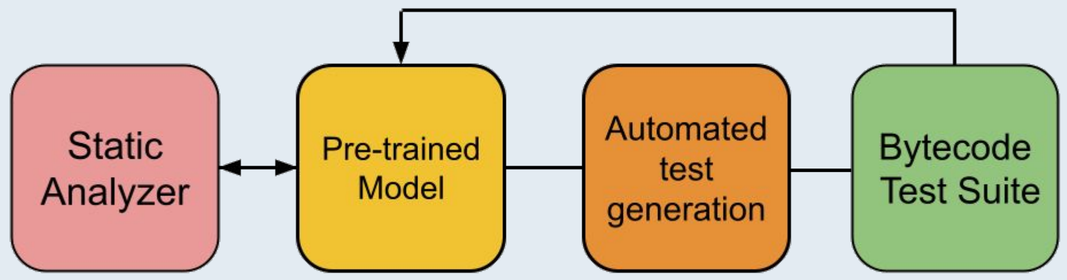
3.1. SATE IV Juliet Test Suite: Java (NSA)
3.2. Draper dataset: C/C++ (Russel et al. ICMLA'2018)[6]

기존 연구의 한계점:

- **광범위한 보안취약점 검출이 아닌 한 분야에만 특화된 모습 (1.1, 1.2)**
- **JVM과 같은 대중적인 플랫폼이 아닌 특정 플랫폼에만 한정 (1.2)**
- **200여개의 자바 관련 CWE중 30% 만을 다룸 (3.1)**

[2] Jackson, Kevin A., and Brian T. Bennett. "Locating SQL injection vulnerabilities in Java byte code using natural language techniques." SoutheastCon 2018. IEEE, 2018.
[3] Pu, Ao, et al. "BERT-Embedding-Based JSP Webshell Detection on Bytecode Level Using XGBoost." Security and Communication Networks 2022
[4] Ferreira Torres, Christof, Hugo Jonker, and Radu State. "Elysium: Context-Aware Bytecode-Level Patching to Automatically Heal Vulnerable Smart Contracts." Proceedings of the 25th International Symposium on Research in Attacks, Intrusions and Defenses. 2022.
[5] Lee, Y. J., Choi, S. H., Kim, C., Lim, S. H., & Park, K. W. (2017, December). Learning binary code with deep learning to detect software weakness. In KSII the 9th international conference on internet (ICONI) 2017 symposium.
[6] Russell, R., Kim, L., Hamilton, L., Lazovich, T., Harer, J., Ozdemir, O., ... & McConley, M. (2018, December). Automated vulnerability detection in source code using deep representation learning. In 2018 17th IEEE international conference on machine learning and applications (ICMLA) (pp. 757-762). IEEE.

IV. Approaches



1. 기존에 존재하는 소스코드 기반 정적분석기와의 성능 상호비교

a. Snyk, SonarQube
접근성과 성능을 고려하여 선택

2. 최근 발표된 pre-trained model 이해 및 구현

a. pre-trained model은 방대한 규모의 데이터로 학습을 먼저 한 후 다른 목적에도 사용되는 machine learning 기법을 의미
b. ChatGPT, BERT, Transformer 활용

3. Automated test case generarion 이해 및 구현

a. PHP automated test case generation research (Stivalet, B외 1명)
b. ChatGPT를 활용하여 보안 취약점을 가지고 있는 Java 코드 생성

4. Bytecode Test Suites 생성

a. CWE : Java에 발생할 수 있는 CWE 종류를 목록화
b. ChatGPT : 목록화된 Weakness들을 포함한 buggy code와 fixed code 생성에 사용
c. Java Juliet 1.3 : 2만여개의 Java 보안 취약점 Dataset으로, 이를 바이트코드로 변환시켜 Bytecode Test Suites로 사용

Thanks to: Jueun20