Predictinf Vulnerable Software Components via Text Mining

Luís Felipe Ramos Ferreira

02 de Junho 2023

Conceitos principais

Mineração de dados/texto

- Técnica de aprendizado de máquina
- Extração de informação e características
- Descoberta de padrões e subgrupos
- Modelos de classificação

Vulnerabilidade de componentes de Software

- Fraqueza no sistema
  - Suscetibilidade a ser explorada por ameaçãr ou problemas aleatórios
  - Sistema vulnerável: existe uma oportunidade para uma ameaça quebrar sua segurança
- Causas
  - Erros
  - Falhas no projeto de software
- Exemplos
  - Códigos com vazamento de memória

Projeto de pesquisa realizado

Introdução

- Utilização de técnicas de mineração de textos para classificação de vulnerabilidades em componentes de Software
- Grande relevância no mundo contemporâneo
  - Internet das Coisas
  - Privacidade e segurança
- Hipótese principal: conseguir classificar códigos que contenham vulnerabilidades
- Aplicações e impactos
  - Demonstra força da área de ciência de dados aplicada à cibersegurança
  - Redução de custos, mais segurança, etc

Trabalhos relacionados

- Aprendizado de máquina já foi aplicado na área da cibersegurança
  - Diferentes features de classificação foram utilizadas, em diversas bases de dados
  - Diferentes objetivos: localizar vulnerabilidades, identificar correlações entre variáveis, etc
  - Muitas visavam predizer defeitos e não necessariamente vulnerabilidades
- Análise estática de código
  - *Fortify Source Code Analyzer* (SCA) - ferramenta de anaślise estática de código para identificar potencias vulnerabilidades no código
  - Utilizado para criação das *labels* do conjunto de dados

Metodologia de pesquisa

- Objetivo: construir um classificador binário para predizer se um software é provável de ser vulnerável
- Dados utilizados: arquivos de texto com códigos escritos em Java
- Métrica para construção das *labels*: um código é dito vulnerável se a ferramenta SCA aponta um ou mais avisos em sua análise e não vulnerável caso contrário
- Predições analisadas com base em uma matriz de confusão
  - Verdadeiros positivos (TP)
  - Falsos positivos (FP)
  - Falsos negativos (FN)
  - Verdadeiros negativos (TN)

Indicadores de performance

- Precisão (P) e Revocação (R)

$$P = \frac{TP}{TP + FP} \quad R = \frac{TP}{TP + FN}$$

- F-Score (F)

$$F_\beta = \frac{(1 + \beta^2)PR}{\beta^2 P + R}$$

- Fall-out (O)

$$O = \frac{FP}{FP + TN}$$

- Objetivo: 80% ou mais para ambas precisão e revocação

Conjuntos de dados

- Aplicações *mobile* da plataforma *Android*
- Considerações nas decisões:
  - Linguagem de Programação (Java)
  - Tamanho (1000 linhas de código no mínimo)
  - Número de versões (ao menos 5)
- 10 aplicações escolhidas ao final
- 10 aplicações extras provindas de pré-instalaçoes do *Android OS* foram adicionadas

Conjunto de variáveis relevantes

Variáveis dependentes

- Ferramentas de análise estática de código são custosas
- SCA - Escaneia o código e monta um relatório de possíveis falhas
- Reporta tipo de vulnerabilidade juntamente com sua escala de ameaça (trabalhos futuros)

Variáveis independentes

- Cada arquivos de código é analisado conforme um *pipeline*
- Tokenização, contagem de caracteres, etc

Exemplos

Listing 1. Source code in file HelloWorldApp.java

```
/* The HelloWorldApp class prints "Hello World!" */
class HelloWorldApp {
  public static void main(String[] args) {
    System.out.println("Hello World!");
  }
}
```

Listing 2. Feature vector for file HelloWorldApp.java

```
args: 1, class: 2, Hello: 2, HelloWorldApp: 2,
main: 1, out: 1, println: 1, prints: 1, public: 1,
static: 1, String: 1, System: 1, The: 1, void: 1,
World: 2
```

Técnicas de aprendizado de máquina utilizadas

- *Labels* e *features* já definidas
- *No free lunch theorem*
- Inicialmente, cinco algoritmos famosos foram considerados
- Dois apresentaram melhores resultados inciais: *Naive Bayes* e *Random Forest*
- Ferramenta utilizada: Weka
- Discretização de *features*: limpagem do conjunto de dados

Principais perguntas a serem respondidas

- Um modelo preditivo pode ser criado?
- Predições sobre vulnerabilidades futuras podem ser feitas?
- Predições entre diferentes projetos podem ser feitas?