FG Software Engineering
Leibniz Universität Hannover
Welfengarten 1
3. Stock, Flur G

Master-Arbeit

- zu vergeben -

Security Code Clone Detection

Hintergrund

Um das Wissen über existierende Schwachstellen der Allgemeinheit bekannt zu machen, existieren Datenbanken denen dieses Wissen innewohnt. Für einen Großteil dieser Schwachstellen gibt es Quellcode Beispiele welche diese Schwachstellen enthalten. Wird während der Entwicklung ähnlicher Quellcode verwendet, so wird ebenfalls eine Schwachstelle ins Projekt eingefügt. Für einen Menschen ist es nahezu unmöglich alle Schwachstellen inkl. deren Quellcodes zu kennen und während der Entwicklung zu berücksichtigen.
Mit Clone Detection lassen sich ähnliche Programmcodeabschnitte identifizieren. Wird Programcode zu einer bekannten Schwachstelle als Referenz bei der Clone Detection genutzt, sollten potentielle Risiken für eine Softwareanwendung erkannt werden. Es gibt bereits bekannte Clone Detection Ansätze, wie den CCFinder [1]. Der CCFinder ist in C++ implementiert. Es existieren hier Schnittstellen um die CCFinder-Bibliothek in Java zu nutzen. Bei den CCFinder handelt es sich um ein Open Source Projekt.
[1] CCFinder: http://ieeexplore.ieee.org/document/1019480/
https://github.com/gpoo/ccfinderx

Aufgabe

Im Rahmen dieser Arbeit soll ein Eclipe Plugin entwickelt werden, dass diese Clone Detection Ansätze nutzt um Programmcode Clone zu erkennen.
Dem Softwareentwickler soll während der Entwicklung ein Hinweis erscheinen, sobald er Quellcode nutzt, welcher ähnlich zu gefährdeten Programmcode ist.
Dies soll Ihnen, welche nur teilweise Security Kenntnisse besitzen, eine zusätzliche Unterstützung zur Programmcode Schwachstellenanalyse liefern.
Hierfür soll das Wissen von bestehenden (Code-) Datenbanken genutzt werden. Bestehende Clone Detection Ansätze müssen für eine Effizienzsteigerung kombiniert, analysiert und verglichen werden um möglichst viele Arten von Code Clones zu erkennen sowie das Ergebnis der Befunde (Recall und die Precision) zu erhöhen. Weiterhin muss sich ein Konzept der Darstellung überlegt werden, welches diese Befunde dem Entwickler mitteilt.
Die Analyse des Programmcodes soll fortlaufend während der Programmierung durchgeführt werden und darf die Arbeit des Entwicklers nicht negativ beeinflussen (Keine Ladezeiten/Stopper) während der Entwicklungsphase.
Anforderungen:
- Analyse bestehender Clone Detection Ansätze (Ggf. CCFinder)
- Gut strukturierter und kommentierter Programmcode (Modular gekapselt)
- Entwicklung eines Eclipse Plugin
-Analyse des Recall & Precision Verhältnisses
- Evaluation des erstellten Ansatzes (Precision & Recall)
- Hoher Java Programmieraufwand

Organisatorisches

Betreuer: M. Sc. Fabien P. Viertel, fabien.viertel@inf .uni-hannover.de, Raum G307
Prüfer: Prof. Dr. Schneider **Beginn**: Ab sofort möglich