Towards Testing Malware Detection Systems using Behavioral Obfuscation

Bachelor thesis

Supervisors: Prof. Dr. Alexander Pretschner, Tobias Wüchner, Sebastian Banescu
Email: {alexander.pretschner, tobias.wuechner, sebastian.banescu} @ in.tum.de
Phone: +49 89 289 – 17, 314
Starting date: immediately

Context
Obfuscation consists of software transformations at the source code, intermediate representation and/or binary level, which aim to hide sensitive information available in the software application from attackers. Such sensitive information includes but is not limited to: proprietary algorithms and confidential data. Software diversity aims to decrease the applicability of attack instances (i.e. exploits) by creating syntactically diverse, but functionally equivalent instances of one software program.

However, software obfuscation and diversity are employed by malware developers, in order to avoid detection by anti-virus software. It is known that diversity in malware renders signature-based detection approaches useless. However, we are not aware of any study which performs a thorough investigation about how syntactic and semantic diversity influences other types of malware detection approaches.

Goal
In this thesis we also plan to investigate the effectiveness of software diversity and obfuscation transformations at binary level against different behavioral malware detection approaches. While some existing publications in the field of malware detection explicitly state that they are resilient against malware variants which employ syntactic code transformations [3, 4, 5], it is not clear whether they are also resilient against semantic code transformations, which change program/malware behavior, however, they preserve its original functionality as well. Therefore, we will focus on semantic obfuscation in this thesis.

Work-plan
1. Develop knowledge of code transformation and malware detection techniques:
   a. Read references and find related work on this topic [1, 2, 3, 4, 5]
   b. Write a state-of-the-art survey on related work, which presents and compares the investigated techniques.
2. Design and implement an obfuscation engine for malware:
   a. Describe in writing each code transformation, offering a rationale regarding which kind of automated detection techniques they can evade.
   b. Implement the chosen transformations as a separate tool which must be able to take a malware binary and outputs several diverse obfuscated versions of the input.
3. Evaluation of the implementation (case-study):
   a. Select a representative set of malware binary samples.
   b. Set-up a system to record the detection rates of different malware detection techniques against the sample set.
   c. Apply the obfuscation and diversification engine to each element of the set
   d. Measure the detection rates of the same malware detection techniques
against the diversely obfuscated set of malware.
e. Analyze and discuss the results and any performance impact of the implemented solution.

4. The final thesis document must contain:
a. Description of the problem and motivation for the chosen approach
b. State of the art survey, including analysis and motivation
c. Rationale for choosing certain technique(s) for implementation
d. Implementation description
e. Evaluation of implementation
f. Conclusions and future work.

Deliverables
- Virtual machine able to run a demo of the implementation, including instructions on how to run the demo.
- The source code of the implementation.
- Technical report with comprehensive documentation of the implementation, i.e. design decision, architecture description, API description and usage instructions.
- Final thesis report written in conformance with TUM guidelines.

References