Software Protection by Virtualization Obfuscation

Master Thesis

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

Context
Proprietary software is often left unprotected by technical means in the hope that legislation will be enough to deter attackers. However, software can be easily distributed to geographical areas where the laws regarding copyright are different than expected. Therefore, we have to employ technical means of protection to raise the bar against attackers who have physical access to the device on which the software is running.

Software obfuscation is a code transformation technique which aims to make computer programs “unintelligible” while preserving their functionality. Virtualization obfuscation is a particular obfuscation technique that aims to protect the intellectual property of a software vendor by hiding the control-flow of a program $P$. Virtualization obfuscation consists of the following 3 steps:

(i) generation of a random bytecode instruction set architecture (ISA), covering all instructions of $P$,
(ii) translation of $P$’s instructions into bytecode instructions of the previously generated ISA and
(iii) generation of an emulator which is able to interpret this ISA.

The results of steps (ii) and (iii) are put together in the same program $P'$, which has the same input-output behavior as $P$.

Goal
One goal of this thesis is to implement a virtualization obfuscator for C# programs and offer it as an open source alternative to commercial obfuscators such as: Eazfuscator\(^1\) and Agile.net\(^2\), which are the only obfuscators for C# that have virtualization obfuscation as a feature. The implementation should be extensible such that other obfuscation transformations can be easily added in the future.

Another goal is to perform a case-study on a real-world software solution offered by Jungheinrich\(^3\), which has agreed to be the industrial partner for this thesis topic. This case-study will include both a performance evaluation and a security evaluation of the implemented virtualization obfuscation engine and its outputs.

---

\(^1\) http://www.gapotchenko.com/eazfuscator.net
\(^2\) http://secureteam.net/.NET-Obfuscator.aspx
\(^3\) http://www.jungheinrich.com/
Work-plan

1. Develop understanding of virtualization obfuscation and other obfuscation techniques (see references).

2. Implementation of virtualization obfuscator for C# programs
   a. The obfuscator should include several dispatch methods of the generated emulator, i.e. switch-statement, nested if-statements, indirect via table lookups, etc.
   b. The emulator will allow stack-operands and register-operands in its ISA.

3. Case-study on real-world software solution
   a. Performance evaluation of the obfuscation engine itself including memory and runtime overhead
   b. Performance evaluation of the obfuscated software including memory and runtime overhead
   c. Security evaluation of the obfuscated software measured via static analysis and dynamic analysis techniques.

4. The final thesis document must contain:
   a. Description of the problem and motivation
   b. Description of the theoretical background
   c. Implementation description
   d. Performance evaluation of implementation and protected programs
   e. Security evaluation of protected programs
   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 VM should also include 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