Project Initiation Document

Evaluation of Methods for Effective Control Flow Recovery

Robin Eklind, 870915-0216

June 8, 2017

1 Basic details

Student name:	Robin Eklind
Draft project title:	Evaluation of Methods for Effective Control Flow Recovery
Course:	*To be determined*
Project supervisor:	Christian Schulte

2 Outline of the project environment and problem to be solved

Control flow recovery is the process of identifying the structures of high-level control flow primitives, such as 2-way conditionals (i.e. *if-else* statements) and pre-test loops (i.e. *for-loop* statements), in unstructured control flow graphs (CFGs).

Information of high-level control flow primitives in the original source code is lost during compilation, and the recovery of such information presents interesting challenges since compiler optimizations (such as jump threading and code relocation) may produce irreducible graphs [1]. Solutions to this problem domain therefore include trade-offs between preservation of the original CFG and introduction of additional nodes (e.g. auxiliary conditional nodes [2] or code duplication through node splitting [3]) to recovery high-level control flow primitives from otherwise irreducible CFGs.

The applications of control flow recovery are versatile. It may be used to:

- facilitate malware analysis
- provide information for branch prediction
- discover and mitigate bugs and security vulnerabilities
- recover source code with high-level control flow primitives
- facilitate verification of compiler output (e.g. Reflections on Trusting Trust [4])
- analyze proprietary algorithms

3 Project aim and objectives

The aim of this project is to evaluate a set of control flow recovery methods, and assess their effectiveness at identifying high-level control flow primitives in control flow graphs.

In order to achieve this aim, the author will:

1. Define a metric for measuring the effectiveness of control flow recovery.
2. Review a set of control flow recovery methods presented in research.
3. Determine if there exist public implementations for the reviewed control flow recovery methods; and if not develop implementations.
4. Evaluate the control flow recovery methods against the defined effectiveness metric.

4 Project deliverables

Documents to be produced:

- Project report.

System artefacts to be developed:

- Library for control flow analysis with support for a set of control flow recovery methods; refer to objective 3.
- Tool for performing control flow recovery on a given CFG; refer to objective 4.

5 Project approach

Use the Clang compiler to produce test cases, as it is capable of emitting LLVM IR from C source code. The goal will be to reconstruct the high-level control flow primitives (such as *for-loops*, *if-else* statements) of the original C code from the LLVM IR.

A preliminary metric for the effectiveness of control flow recovery is how many of the original high-level control flow primitives that were recovered and how many were omitted. The metric for effectiveness should also includes a notion of false positives for high-level primitives that were recovered but were not part of the original source code. The metric for effective control flow recovery will be refined as more intuition is gained in the problem domain, and the final metric for effective control flow recovery will be clearly defined as part of the project.

All work will be made available on GitHub from day one, and the source code will be released into the public domain to encourage open source adaptation. The project plan will be organized using the GitHub issue tracker, where each issue corresponds to a task. Milestones, containing a set of issues, will track the progress of the project and enforce deadlines.

6 Starting point for research

- C. Cifuentes, Reverse Compilation Techniques [1]
- K. Yakdan, S. Dechand, E. Gerhards-Padilla and M. Smith, Helping Johnny to Analyze Malware: A Usability-Optimized Decompiler and Malware Analysis User Study [5]
- E. Schwartz, J. Lee, M. Woo and D. Brumley, Native x86 Decompilation using Semantics-Preserving Structural Analysis and Iterative Control-Flow Structuring [6]
- K. Yakdan, S. Eschweiler, E. Gerhards-Padilla and M. Smith, No More Gotos: Decompilation Using Pattern-Independent Control-Flow Structuring and Semantics-Preserving Transformations [2]
- T. Conradi, Matching of Control- and Data-Flow Constructs in Disassembled Code [7]
- S. Moll, Decompilation of LLVM IR [3]

References

[1] C. Cifuentes, *Reverse Compilation Techniques*. PhD thesis, Queensland University of Technology, 1994.

[2] K. Yakdan, S. Eschweiler, E. Gerhards-Padilla, and M. Smith, "No more gotos: Decompilation using pattern-independent control-flow structuring and semantics-preserving transformations," in *NDSS*, 2015. [Online] Available: `https://net.cs.uni-bonn.de/fileadmin/ag/martini/Staff/yakdan/dream_ndss2015.pdf`. [Accessed: 08 Jun 2017].

[3] S. Moll, "Decompilation of LLVM IR," BSc thesis, Saarland University, 2011. [Online] Available: `http://www.cdl.uni-saarland.de/publications/theses/moll_bsc.pdf`. [Accessed: 08 Jun 2017].

[4] K. Thompson, "Reflections on trusting trust," *Communications of the ACM*, vol. 27, no. 8, pp. 761–763, 1984.

[5] K. Yakdan, S. Dechand, E. Gerhards-Padilla, and M. Smith, "Helping johnny to analyze malware: A usability-optimized decompiler and malware analysis user study," in *Security and Privacy (SP), 2016 IEEE Symposium on*, pp. 158–177, IEEE, 2016. [Online] Available: `https://net.cs.uni-bonn.de/fileadmin/ag/martini/Staff/yakdan/dream_oakland2016.pdf`. [Accessed: 08 Jun 2017].

[6] E. J. Schwartz, J. Lee, M. Woo, and D. Brumley, "Native x86 decompilation using semantics-preserving structural analysis and iterative control-flow structuring," in *Proceedings of the USENIX Security Symposium*, vol. 16, 2013. [Online] Available: `https://www.usenix.org/system/files/conference/usenixsecurity13/sec13-paper_schwartz.pdf`. [Accessed: 08 Jun 2017].

[7] T. Conradi, "Matching of Control- and Data-Flow Constructs in Disassembled Code," BSc thesis, Hamburg University of Technology, 2015. [Online] Available: `https://www.sts.tuhh.de/pw-and-m-theses/2015/conradi15.pdf`. [Accessed: 08 Jun 2017].