Thesis Proposal: Generating Safety Proofs for eBPF in LLVM

Martin Fink

November 2025

Details

Type	MA/BA/GR/IDP
Timeframe	WS24/25, SS25
Keywords	eBPF, PCC, Z3, static analysis

1 Introduction

eBPF [2] programs run inside the kernel without user–kernel boundaries, which means any mistake can corrupt kernel memory or break isolation. Current verifiers run entirely on generated eBPF code and are limited [2, 3]: They reject many safe programs and still fail to catch certain bugs. To address this, we propose a two stage verification approach, based on Proof-Carrying Code (PCC) [4]. The eBPF compiler generates safety proofs during compilation and the kernel only checks these proofs after JIT compilation, which provides end to end safety while reducing false rejections.

2 Objective of the Thesis

In this thesis you will implement the LLVM side of the system. You will integrate Z3 [1] into LLVM, derive safety properties during IR lowering, and emit compact proof artifacts that the kernel level verifier can check.

3 Required Skills

- C++ knowledge
- Experience with LLVM
- Basic understanding of compilers and static analysis

References

[1] Leonardo de Moura and Nikolaj Bjørner. "Z3: An Efficient SMT Solver". In: *Tools and Algorithms for the Construction and Analysis of Systems*. Ed. by C. R. Ramakrishnan and Jakob Rehof. Berlin, Heidelberg: Springer, 2008, pp. 337–340. ISBN: 978-3-540-78800-3. DOI: 10.1007/978-3-540-78800-3_24.

[2] Linux Kernel Developers. *eBPF Verifier — The Linux Kernel Documentation*. https://docs.kernel.org/bpf/verifier.html. (Visited on 11/12/2025).

[3] Elazar Gershuni et al. "Simple and Precise Static Analysis of Untrusted Linux Kernel Extensions". In: *Proceedings of the 40th ACM SIGPLAN Conference on Programming Language Design and Implementation*. PLDI 2019. New York, NY, USA: Association for Computing Machinery, June 2019, pp. 1069–1084. ISBN: 978-1-4503-6712-7. DOI: 10.1145/3314221.3314590. (Visited on 11/07/2025).

[4] George C. Necula. "Proof-Carrying Code". In: *Proceedings of the 24th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*. POPL '97. New York, NY, USA: Association for Computing Machinery, Jan. 1997, pp. 106–119. ISBN: 978-0-89791-853-4. DOI: 10.1145/263699.263712. (Visited on 06/21/2024).