# Automation of Formal Verification (AUFOVER)

Red Hat
Kamil Dudka
November 11th 2021

## Abstract

Red Hat uses static analyzers to automatically find bugs in the source code of Red Hat Enterprise Linux, consisting of 480 million lines of code and 3700 RPM packages. There are open source tools that can statically analyze this amount of software in a fully automatic way. Could we use formal verification tools to find bugs in (or even prove correctness of) the important pieces of code in our Linux distribution? Thanks to the Automation of Formal Verification (AUFOVER) project, Red Hat was able to integrate formal verifiers Symbiotic and Divine, which are developed by research groups of Masaryk University in Brno. Are these tools ready for industrial software?

# Automation of Formal Verification (AUFOVER)

- Project supported by Technology Agency of the Czech Republic: https://starfos.tacr.cz/en/project/TH04010192
- Driven by Honeywell as the main participant.
- Masaryk University and Brno University of Technology participated.
- Red Hat collaborated primarily with Masaryk University.
- Red Hat integrated formal verification tools into a Linux distribution.

# Static Analysis at Red Hat

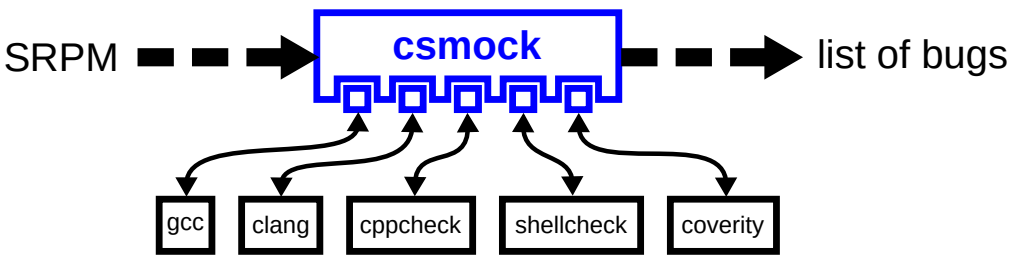
- Automatic detection of programming mistakes in the source code of Red Hat Enterprise Linux (RHEL)
- 480 million lines of code in 3700 RPM packages in RHEL-9 Beta
- Static analysis is fast but its precision is limited.
- False positives and false negatives are expected.

# Formal Verification

- Formal verification can (in theory) prove that programs are correct.
- Formal verification tools are developed at Masaryk University:
  - Divine – explicit-state model checking
  - Symbiotic – instrumentation, slicing and symbolic execution
- Now available in Fedora!
- How can one formally verify an RPM package?

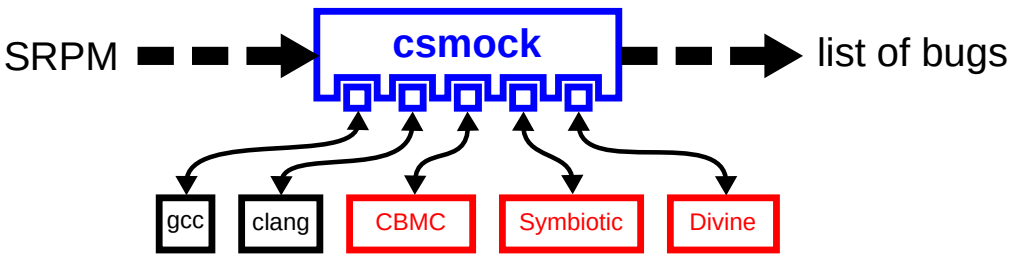
# Static Analysis of RPM Packages

- Command-line tool to run static analyzers on RPM packages.
- One interface, one output format, plug-in API for static analyzers.
- Could we integrate formal verification tools?



# Formal Verification of RPM Packages

```
$ sudo yum install csmock-plugin-symbiotic
$ csmock -r fedora-34-x86_64 -t symbiotic ${pkg}.src.rpm
```



# How does it work?

- Modern RPM packages include regression tests.
- A custom ELF interpreter is set at build-time.
- Dynamic linker wrapper (csexec) is used at run-time.
- Presented at DevConf 2021 (in the context of dynamic analysis):
  - slides: https://kdudka.fedorapeople.org/kdudka-devconf-21.pdf
  - video: https://www.youtube.com/watch?v=FjV84hbD1GY
  - demo: https://github.com/csutils/cswrap/wiki/csexec

# Experiments

- Unable to complete formal verification for most RPM packages.
- Timeouts help to get partial results in a predictable amount of time.
- aufover-benchmark (covered by CI) is now publicly available: https://github.com/aufover/aufover-benchmark
- Our experiments can be easily reproduced on any Fedora system!