Open-Source Tools by Yulei Sui

SVF (https://github.com/SVF-tools/SVF) has served as a foundational framework for developing other program analyses and has also been used for online learning of software analysis and verification (https://github.com/SVF-tools/SVF-Teaching. On its GitHub website, SVF has 1400+ stars and 440+ forked repositories with different users and contributors from both academia and industry. SVF has been adopted and extended by over 200 research groups from both industry and academia. These groups hail from leading institutions such as MIT, UIUC, Imperial College London, Cambridge, EPFL, University of Virginia, the University of Toronto, and Purdue University, as well as users from major corporations like Amazon, Google, Apple and Meta. This extensive adoption has led to more than 200 research papers in top-tier venues and the completion of 40+ PhD theses

SVF has been used and cited by researchers from leading program analysis and security groups, e.g. Chopped Symbolic Execution (from Imperial College London@ICSE and @FSE), PinPoint (from HKUST@PLDI), Type-based CFI (from ACSAC@MIT and Northeastern University), Kernel Fuzzing (from Purdue@IEEE S&P), Directed Fuzzer (from NTU@CCS), K-Miner (from TU Darmstadt@NDSS), Permission Check Analysis (from Virginia Tech & Zhejiang University @USENIX Security), probabilistic analysis (from University of Pennsylvania @PLDI), and Hybrid Testing (from Northeastern University @S&P), and system call specialization (from Northeastern University @USENIX Security), and hot patch generation for kernels (from NTU @USENIX Security), and fuzzing for kernel file system (from Georgia Institute of Technology @S&P).

The following lists some comments (extracted from emails and GitHub websites; only their organisations are shown with names removed for anonymity) made on Dr Sui's open-source tools by people from both academia and industry:

- *"Looking for llvm pointer analysis I stumbled over SVF and was quite impressed by it's capabilities - thanks a lot for making it open source!"*

  — Matthias Neugschwandtner $\langle EUG@zurich.ibm.com \rangle$, working in the Cloud and Storage Security Group at IBM Research.

- *"Your tests have been most helpful! ...I'm trying to convince Arthur that SVF can solve some (all?) of the design requirement laid out by Chandler (Google)."*

  — C Bergstrom $\langle cbergstrom@pathscale.com \rangle$ CTO at Pathscale, HPC Compiler Company.

- *"First, I'd like to thank you for your work on SVF and for making it public! The academic LLVM community is in dire need of such a framework and I am very interested in seeing your framework become the foundation for many analyses moving forward. "*

  — Will Dietz $\langle wdietz2@illinois.edu \rangle$, Researcher at UIUC and ALLVM.

- *"I came across SVF recently and am really interested in using it. ... it's a really nice set of tools! Again thanks! "* — Jared Carlson $\langle jared.carlson23@gmail.com \rangle$, Sr. Security Researcher at Veracode, a leading company working in software application security.

- *" I remember watching your presentation on SVF and thought: that's cool, was so excited to be able to reach you. . . . Would be great to be able to chat about how maybe you and our team might be able to work on interesting things together"*

  – Dean Michael Berris $\langle dean.berris@gmail.com \rangle$, Software Engineer at Google.

- *'I hope to use SVF for the DARPA TRACE grant application.'*

  — Stephen N. Lee from University of Wisconsin-Madison, University of Wisconsin-Madison

- *"I'm currently working with your awesome SVF analysis."*

  — Simon Schmitt $\langle symenschmitt@web.de \rangle$ from TU Darmstadt.

- *"I've run into your paper: Region-based Selective Flow-Sensitive Pointer Analysis. It's a great work!"*

  — Marek Chalupa $\langle mchalupa@mail.muni.cz \rangle$ from Masaryk University.

- *"Thank you again and I think that the SVF will be promising for my work."*

  — Muhammad Refaat Soliman $\langle mrefaat@uwaterloo.ca \rangle$ from Waterloo University.

- *" We very much appreciate your work and would like to use the static pointer analysis you developed."*

  — Alexandra Jimborean $\langle alexandra.jimborean@it.uu.se \rangle$, from Uppsala University.

- *" I'm very interested in the SVF analysis that can be performed with your software."*

  — Timo Bressmer $\langle timo.bressmer@uni - ulm.de \rangle$ from Ulm University.

- *"We heard a lot about SABER: Static Memory Leak Detection Using Full-Sparse Value-Flow Analysis. We would like to test our improvement ideas for SABER "*

  — Ahmed Tamrawi $\langle atamrawi@iastate.edu \rangle$ from Iowa State University.

- *"I recently tested SVF (commit 5355fc2). Great piece of work from my point of view! . . . Thank you for providing the code!"*

  — Oliver Braunsdorf $\langle oliver.braunsdorf@tu - ilmenau.de \rangle$ from at TU Ilmenau.

- *" As far as I can tell, the tool fixes all issues I have with musl libc. Thank you."*

  — Anh Quach $\langle aquach1@binghamton.edu \rangle$ from Binghamton University.

- *"Thanks for your great project. "*

  – Qingkai (Thomas) Shi $\langle qingkaishi@gmail.com \rangle$ from Hong Kong University of Science and Technology.

- *" Thanks for the quick answer:) I am looking at SVF implementation, in order to test one of our ideas. "*

  – David Trabish from Tel-Aviv University

- *" Many thanks for this excellent codebase! I'm new to SVF but I have updated it for LLVM 5.0.0 for a project of mine. "*

  – Jack Anthony from Cambridge University, UK

- *"Awesome project! "*

  – Miguel Arroyo from Columbia University

- *"Thank you for sharing and maintaining such a great project! "'*

  – Yian Su from Northwestern University, USA

For teaching and education, SVF has not only been used within UNSW for course 'COMP6131 Software Security Analysis' (`https://webcms3.cse.unsw.edu.au/COMP6131/24T2/`), and at UTS for courses '41128 Software Analysis Studio' (`https://handbook.uts.edu.au/subjects/41128.html`) and '41184 Secure Programming and Penetration Testing' (`https://handbook.uts.edu.au/subjects/41184.html`), but also worldwide. For example, SVF has also been used as a representative static analysis approach in Prof. Qirun Zhang's 'Topics in Program Analysis' class at Georgia Institute of Technology (`https://helloqirun.github.io/course/cs8803/index.html`). SVF has also been recognised as a representative pointer analysis approach in Prof. Phillip B. Gibbons's 'Optimizing Compilers for Modern Architectures' class at Carnegie Mellon University (`https://www.cs.cmu.edu/afs/cs/academic/class/15745-s19/www/lectures/L13-Pointer-Analysis-pre-class.pdf`). Additionally, SVF has been referred to as a standard alias analysis framework in Prof. Simone Campanoni's 'Advanced Topics in Compilers' class at Northwestern University, USA (`https://users.cs.northwestern.edu/~simonec/files/Teaching/ATC/slides/NOELLE_dependences.pdf`).