Reverse Engineering Object-Oriented Code with Ghidra and New Pharos Tools

Featuring Jeff Gennari as Interviewed by Cory Cohen

Welcome to the SEI Podcast Series, a production of the Carnegie Mellon University Software Engineering Institute. The SEI is a federally funded research and development center sponsored by the U.S. Department of Defense. A transcript of today's podcast is posted on the SEI website at sei.cmu.edu/podcasts.

Cory Cohen: Hello, my name is Cory Cohen. I am a researcher at the Software Engineering Institute, working on binary program analysis. I am here today with my colleague Jeff Gennari to talk about some of our work in the Pharos Binary Static Analysis Project. So, Jeff, in the years that we have worked together, I have done vulnerability analysis, program analysis, incident response, some malware analysis. Tell the audience a little bit about what kinds of work you have done here.

Jeff Gennari: I am a senior researcher also in the program-analysis/binary-analysis space. I too did vulnerability work before that, malware analysis, reverse engineering. I teach a few classes at CMU [Carnegie Mellon University] in software reverse engineering and software verification, and I am a developer in Pharos.

Cory: We are here today primarily to talk about our recent updates to Pharos on the Pharos website, in particular our updates to the object analyzer [OOAnalyzer] program. But I wanted to start by asking a little bit about, why is object-oriented reverse engineering a serious challenge problem for the Department of Defense?

Jeff: Object-oriented [OO] code includes many high-level abstractions that are difficult to reverse engineer. The binary representations of objects include a lot of state that is not captured well by existing tools. The Pharos work, and the object analyzer in particular, has been focused on recovering those abstractions and applying them in a format that reverse engineers can easily reason about, so they can apply those to their tools.

Cory: So, Jeff, tell me a little bit more about what Pharos is and how it works, and what kinds of things you can do with it?

Jeff: We talk about Pharos as a platform because it really is a collection of services to support reverse engineering and program analysis. It does everything from disassembly to partitioning of functions to identification of higher-level data structures, of which OOAnalyzer is a part. Pharos also provides a robust analytics framework, where the semantics of different instructions—what they actually do and their impact on the system—can be analyzed and used to produce deeper analyses.

Cory: I can add to that that we have a number of really great collaborators working with us in this Pharos work. The most obvious is our partners at Lawrence Livermore National Labs, the ROSE developers that the Pharos platform is built on top of. We work very closely with them. We have also had good interactions with Prolog developers at XSB and SWI-Prolog who have helped us develop and expand the capabilities of our OO analysis tool. And we have a partnership with Arie Gurfinkel at the University of Waterloo, who is helping us integrate the Z3 SMT solver into Pharos, so that we can answer reachability questions and other program-analysis problems.

Jeff: I think that is one of the big design goals that we had. Pharos is not just a self-contained, reverse-engineering platform. We're always looking for new ways to bring in formal computer science program-analysis strategies to bear on this problem. So the inclusion of SMT solvers, getting into symbolic-execution type problems, and model checkers really allows us to explore these problems in ways that other entities do not.

Cory: What else is new in this release that people on the Internet might be interested in?

Jeff: Aside from some of the extensions we will talk about, this release includes a number of bug fixes and performance enhancements. Notably there are improvements to the macOS port, so Mac users can now use our tools natively. I guess the next thing to talk about is the extensions themselves.

Cory: Yes, so we have obviously released a Ghidra plug-in for the OOAnalyzer tool. Why is a Ghidra plug-in important, and why have we chosen at this time to do some work in Ghidra?

Jeff: Ghidra was recently released by the NSA [National Security Agency]. It is a new reverse-engineering platform that is very interesting. It provides a lot of new interesting features like a robust decompilation. We thought it was very important to get our tools into that space to help DoD [Department of Defense] reverse engineers.

Cory: What is next for the Pharos platform in general?

Jeff: Well, we are going to continue to look at different reverse-engineering platforms like Ghidra. We still produce and maintain tools for IDA Pro. We continue to work on applying

rigorous reasoning frameworks, like SMT solvers, into this space, and we continue to include new features in object-oriented analysis into OOAnalyzer.

Cory: If people want to be more involved in the Pharos project, how would you recommend that they get started?

Jeff: I recommend that people go to github.com/cmu-sei/pharos and just get a sense of what tickets are available or open, what comments are out there, what updates are going on, so they can see what the recent activity was.

Cory: I can talk a little bit more about the capabilities that are on the GitHub site. We do have some documentation there. There is a Docker image, so that people can download the Docker image and get started without actually having to build the software. You can build from source from the GitHub site, if people are interested in doing that. After experimenting with the tools for a little while, if they have any questions, I would encourage them to just open an issues ticket on GitHub for us. We try to stay on top of those and respond and provide assistance as we are best able.

Thank you so much for joining us today. As we have mentioned in this podcast, we have released a new version of the Pharos tools. They are available at github.com/cmu-sei/pharos. You can also keep up to date with what we are doing on our team by following the SEI blog at insights.sei.emu.edu. Thank you.

Thanks for joining us. This episode is available where you download podcasts, including SoundCloud, Stitcher, TuneIn Radio, Google Podcasts, and Apple Podcasts. It is also available on the SEI website at sei.cmu.edu/podcasts and the SEI's YouTube channel. This copyrighted work is made available through the Software Engineering Institute, a federally funded research and development center sponsored by the U.S. Department of Defense. For more information about the SEI and this work, please visit www.sei.cmu.edu. As always, if you have any questions, please don't hesitate to email us at info@sei.cmu.edu. Thank you.