Pharos Binary Static Analysis: An Update

featuring Jeffrey Gennari as Interviewed by Suzanne Miller

Suzanne Miller: Welcome to the SEI Podcast Series, a production of Carnegie Mellon University's Software Engineering Institute. The SEI is a federally funded research and development center operated by Carnegie Mellon University and sponsored by the U.S. Department of Defense. Today's podcast will be available at the SEI's website, www.sei.cmu.edu/podcast.

My name is Suzanne Miller. I am a principal researcher here at the SEI. Today I am very pleased to have with me Jeffrey Gennari, who is a senior member of the technical staff in CERT's Threat Analysis Division. So, welcome Jeff.

Jeffrey Gennari: Thank you.

Suzanne: Thank you very much for taking the time out of your schedule to come and talk with us. Before we get started in terms of the aspects that we are going to talk about today, in terms of automated reverse engineering and some of the things that does for us, tell us a little bit about yourself, how you got to be here, and what brought you to the SEI and made you interested in this kind of work.

Jeffrey: Well, I have been at the SEI for over a decade. I started in the vulnerability analysis world in the mid-2000s, studying software vulnerabilities and why they occurred, how to remedy them. That led me down a path to studying software flaws, software security flaws, which got me into secure coding and eventually wound me into reverse engineering and looking at disassembly. A lot of times we did not have source code.

When you are doing reverse engineering, you kind of naturally head into the malware analysis domain. Malware almost never has source code, at least in the artifacts that we find. That is how I started on this path. I did reverse engineering for about 5 or 6 years and got incredibly frustrated with doing it by hand all the time, so I got on board with a team that was working in automated reverse engineering, which leads us to this project.

Suzanne: The SEI has done work in reverse engineering in the past. A lot of it has been focused on source-code analysis, and that leads you into architecture and other things. But you are working with the binaries. You are automating the reverse engineering against the binaries. What is different about today's world than, say, 10 years ago, when you started working with vulnerabilities, that allows us to even think about automating this process?

Jeffrey: Well, a lot of the supporting tools that we use and the supporting infrastructure has matured a bit. We work off a platform known as the ROSE compiler framework out of Lawrence Livermore Labs. We have extended that and developed relationships with those developers. A lot of that supporting technology has matured to the point where we can take advantage of it. We bring our experience as reverse engineers and software developers to that infrastructure and are able to hone it so that the problems that it can solve can be applied to binary code.

Suzanne Miller: What is it that you are looking for when you are doing this reverse engineering? What are you trying to get out of one of these reverse-engineered programs?

Jeffrey: Any number of things. We are all about program understanding. We want to make the job of the analysts a little bit easier. In the traditional, manual, reverse engineering, everything was done by hand. There are tools out there, disassemblers like IDA Pro, that are really good at doing some analysis but are better at incrementally annotating and gaining insights, but [the disassembler] is still driven by a human. We seek to have this all done in the automated way. Save as much time as possible and get as much up front information out of a piece of software to give to an analyst so they can go ahead and get higher-level insights.

Suzanne: And so, the insights in particular that you are trying to enable are malware insights?

Jeff: Yes.

Suzanne: As you mentioned at the beginning, we do not get source code from malware. So, once you have actually been able to do the malware analysis, can you actually go back in and provide remedies for malware with this technique? Or, is this really just the analysis part, and then you are going to need other kinds of techniques to actually be able to remedy the malware?

Jeffrey: Remediation is best done on a case-by-case basis. It really depends on the piece of malware you are looking at. What we want to do is get to that point as fast as possible. So, we get a piece of software. We know nothing about it. We have a battery of tools that we call Pharos that do things like recover object-oriented structures and data structures that are in the code. Those can be really difficult for analysts to find and deal with manually. We generate a lot of statistical information like hashes of functions, information about function call graphs and put that, put this, all together automatically and then give that to the analyst, or at least give them some kind of enhanced representation so they can more easily make insights.

Insights really vary per individual case, but doing things like finding indicators, being able to put together program functionality, being able to find off switches or kill switches for a piece of malware, that is something that we can influence and just generally getting them [the analysts] to the point where they feel comfortable explaining what the software does—which is a big part of reverse engineering—as fast as possible.

Suzanne: So if I was a security analyst, I would want to get my hands on those tools?

Jeffrey: Yes.

Suzanne: Is that something that the SEI is enabling? Is actually getting those out to the analyst community more publicly? Or, is this something they have to come to the SEI to actually get that service?

Jeffrey: We have recently released our tools on GitHub, and our infrastructure on GitHub. With the infrastructure, not only can you use our tools that we are making available, but you can develop your own tools for your own analytical problems. We have also released a number of blog posts and an SEI Cyber Minute explaining how to use our tools, and some of the deeper bits of the technology that go into how they work.

Suzanne: For the sake of our viewers, all of those will be listed in our transcript. We always list any blog posts, any URLs, and things like that for getting to those. So, that is very exciting that this is now becoming available to the public. Have you gotten any feedback so far? Has it been out long enough for people to actually give you feedback?

Jeffrey: It's been out long enough for people to start using the tools. We have had a few bits of interest from various researchers in academia and in industry, but we would always love for people to really get into the tools, try them, see if they work, break them so we can identify shortcomings, make improvements.

Suzanne: Excellent. So, any of our viewers are really asked to do that, to provide feedback back to you on the tools. All the feedback, I know how we are at the SEI, we consider all feedback. It does not mean that anything explicit will be done, but we like to see what is going on with that. So, this is a big milestone, but as we start talking about remedies, that is the other side of this. Have you started doing work that would lead you in that direction?

Jeffrey: We have just started, actually, a new large body of work, a multi-year body of work that is going to be based on our technology in unwanted feature identification and removal. This would be the case where you have a piece of software that does something you do not necessarily like. It calls out to a remote host that you do not necessarily trust or does something, has some bloatware in it that you want to get around.

This new work would help take a binary, find that feature, and go around it so as to make the software a little bit more usable, less risky, or have more desirable functionality.

Suzanne: So, it would not actually remove it. It would just give a path around it?

Jeffrey: Yes, you can imagine this would be useful in cases where there is a backdoor or a component that has a significant vulnerability that can't be fixed.

Suzanne: So, when you get done with that, I know a bunch of people in the legacy part of the world that would love for you to figure out how to actually excise the pieces of dead code that they do not want. That is for the future. We will get the path around first.

This is exciting work. I am really glad that we are able to make this available to the world so that people that are out in the public, in the security field, will be able to access it.

I look forward to some stories, in the future, of things that were avoided because of something like this. I really do think this is the kind of thing that would enable some of our infrastructure, especially, to be able to find things that they could not find before, and knowing what is out there is always the first step to remedying the things that we do not want.

I want to thank you very much for your time.

Jeffrey: Thank you.

Suzanne: …and for telling us about this and for your blog post. That is where people will get a lot more detail about these tools. I am sure there are descriptions of each of the individual tools out there, so all that is going to help people to use this. Thank you very much.

Jeffrey: Thank you.

Suzanne: This podcast is available, as I said, at the SEI's website sei.cmu.edu/podcasts. It is also available on our YouTube channel. And I want to thank you for joining us today. As always, if you have any questions, please do not hesitate to contact us at info@sei.cmu.edu. Thank you.