DefCon presenters explore programmer de-anonymization, stylistic fingerprints

August 15 2018, by Nancy Owano

Credit: CC0 Public Domain

One of the nicer things about higher education: Gaining awareness of the signature styles of authors, painters, musicians even before we are told their names. Well, signature styles are not just confined to the arts.

Two researchers can show the world their work on stylistic fingerprints and how these can be used to potentially identify programmers from code and binaries.

"Machine Learning Can Uncover Programmers' Identity," was the headline from *Fossbytes*. The article was talking about Rachel Greenstadt and Aylin Caliskan, who presented their work at DefCon. Greenstadt is associate professor, Drexel University; Caliskan is an assistant professor of computer science, George Washington University.

"Stylistic fingerprints"? Meaning? Louise Matsakis in *Wired* looked at something called stylometry—the statistical analysis of linguistic style. She said that "newer research shows that stylometry can also apply to artificial language samples, like code. Software developers, it turns out, leave behind a fingerprint as well."

In this area, anonymous programmers can be identified. *Fossbytes* summed up the research effort: They tested codes submitted by programmers and the system could correctly identify 83 percent of the times the algorithm was run.

They explored "programmer de-anonymization" with machine learning. They arrived at the conference ready to show how abstract syntax trees have "stylistic fingerprints," and sleuths can use these fingerprints potentially to identify programmers, from code and binaries. The question comes up: are these algorithms from heaven or from hell? Two sides of the coin.

The plus factor, obviously, would be in identifying those authors who plant malware. Negative factor: Coders who like to contribute code anonymously may be put off by this, as noted in *Fossbytes*. "There are times when programmers would like to remain unknown for legit reasons and getting identified is not always a good thing."

Matsakis also remarked on privacy implications, "especially for the thousands of developers who contribute open source code to the world."

Wired described their exploration as a binary experiment, where Caliskan and other researchers used code samples from Google's annual Code Jam competition. The machine learning algorithm correctly identified a group of 100 individual programmers 96 percent of the time, using eight code samples from each.

As interesting, even when the sample size was widened to 600 programmers, "the algorithm still made an accurate identification 83 percent of the time."

Cory Doctorow in *Boing Boing*, meanwhile, mentioned additional insights in programming styles. Doctorow reported that, actually, they found that experienced developers appeared easier to identify than novice developers. The more skilled you are, the more unique your work apparently becomes.

How so? Doctorow commented that may be "in part because beginner programmers often copy and paste code solutions from websites like Stack Overflow."

More information: De-anonymizing Programmers from Source Code and Binaries, www.defcon.org/html/defcon-26/ … kers.html#Greenstadt