MIT LINCOLN LABORATORY

The Bulletin

News, events, and notices transmitted weekly

March 05, 2021

Bug-Injecting System Helps to Advance the State-of-the-Art in Debugging Software

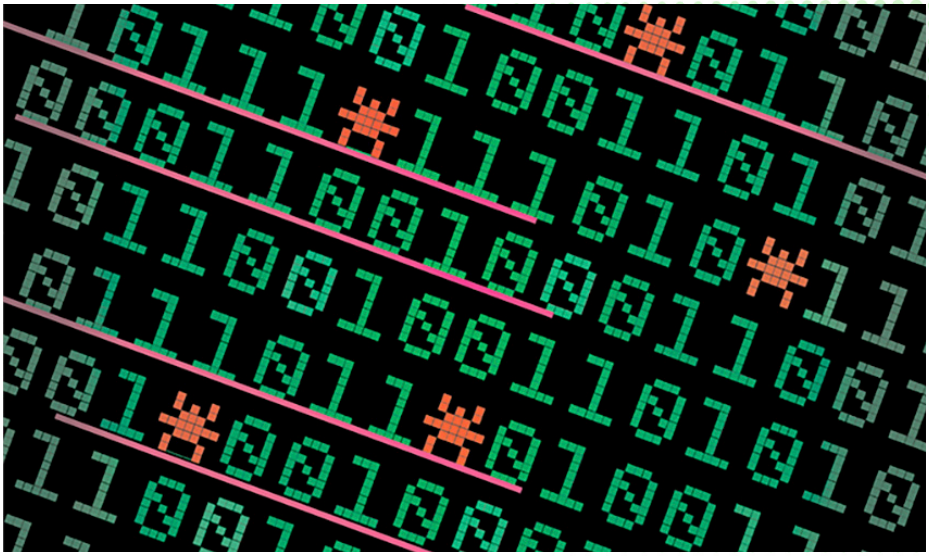
Division 5: Cyber Security and Information Sciences | Lincoln Laboratory

Since about the turn of the century, there has been rapid advancement in technological fields such as self-driving vehicles, image recognition, and human language technology. One of the key factors spurring this technology has been competition, and the Laboratory also is playing a key role in driving innovation in these fields.

One example of the Laboratory spurring progress is a system called Large-scale Automated Vulnerability Addition (LAVA), which enables evaluation of bug-finding systems. Bug finding systems are used after developers have written code to try to identify mistakes they have made. If these systems find a bug, they can be fixed easily before code is deployed. Unfortunately, these systems fail to find many bugs, which is one of the reasons why new vulnerabilities and crashes still exist in computer programs today.

"Finding bugs in software is like trying to find a tiny needle buried deep in a very large haystack," said Timothy Leek, Senior Staff, Cyber System Assessments, Group 59.

For many years, the cyber security community has sought to develop new techniques and tools that can more efficiently find bugs in software. However, in 2014, Leek recognized a flaw in these approaches; the scarce documentation of known bugs and how those bugs manifest in a program was making it impossible to measure the success of bug-finding tools. Computer scientists needed to measure the effectiveness of their bug-detectors to know when they were failing to detect a bug. "If a bug finder reports finding 42 bugs in a program, there is no way to know whether that's 1% or 99% of the total number of bugs actually in the program," said Professor Brendan Dolan-Gavitt, a collaborator on LAVA from New York University.



By automatically injecting thousands of bugs into program code, LAVA aims to improve vulnerability detection.

The LAVA system works by producing thousands of realistic bugs that are automatically injected into pre-existing program code. "It turns out that adding bugs to programs teaches us a lot about how bugs work and thus about how to find them," Leek explained. Once these bugs are injected, various vulnerability discovery techniques and software can be tested to see how many of the bugs they can find and how many they miss.

In a virtual seminar on 22 January, staff presented information about four of the eight Laboratory-developed technologies that were selected to receive a R&D 100 Award in 2020, and LAVA was one of these technologies. Speaking during that seminar, Andrew Fasano, Group 59, summarized the usefulness of having a system like LAVA.

Bug-Injecting System Helps to Advance the State-of-the-Art in Debugging Software (continued)

"LAVA was the first system of its kind," said Fasano. "In the time since [our work was first published], we've seen a significant increase in the number of bug-finding systems being created, and the majority of those new systems have used our corpus in their evaluation."

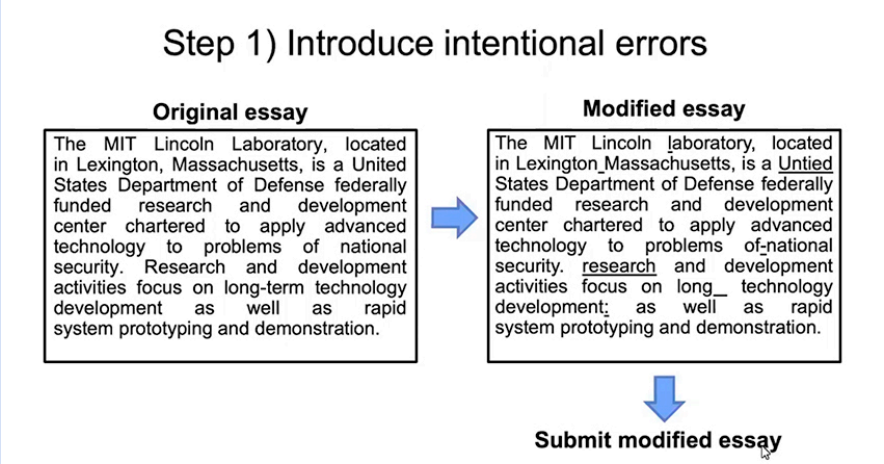
LAVA was developed in 2015–17 with Line funding, and leverages PANDA, another Laboratory-developed dynamic analysis platform and previous R&D 100 winner. In 2016, staff published a paper in IEEE Symposium on Security and Privacy about LAVA, and a few years later, LAVA code was released on GitHub. As a result, over the past five years, LAVA has become the first widely used benchmark for evaluation of bug-finding systems.

After developing and sharing their bug-injecting program, staff needed a way to further test its effectiveness while also disseminating it to the cyber security community. So, they created Rode0day—a bug-finding competition powered by LAVA. Each month, Laboratory researchers released a new, never-before-seen collection of buggy programs and then challenged state-of-the-art bug finding systems to find as many bugs as they can. More than 3,000 LAVA-bugs were injected into more than 60 programs as part of this competition, and the results are being used to develop novel bug-finding systems today. The LAVA team later released their findings from the competition in a paper entitled "The Rode0day to Less-Buggy Programs" that was published to IEEE.

Through the monthly challenges of Rode0day, the LAVA team discovered that it was helpful for the cyber community to have fresh data samples to evaluate and test the effectiveness of their bug-finding tools—however, the competition was insufficient to conduct rigorous evaluations of bug-finding systems, as the Laboratory team didn't know what resources and techniques competitors were using. So, the researchers decided to use the Lincoln Laboratory Supercomputing Center to test leading bug-finding systems.

"Using the Laboratory's supercomputing resources, we were able to run an experiment to test eight of the leading bug-finding systems with a subset of the Rode0day challenges that had nearly 1,000 bugs added," said Fasano, who further explained that this allowed staff to make comparisons using different systems, configurations, and durations. "Using a standard one-CPU system, it would have taken more than 83 years to collect all of the data."

The researchers will be presenting an analysis of their data at the ACM AsiaCCS conference in June, and plan to use this data to continue to advance LAVA's utility and to create more difficult benchmarks for the bug-finding community.



In the illustration above, spelling and grammar errors are introduced to an essay to see if a proofreader can spot those errors. LAVA does this in a similar manner with program code.

MIT LINCOLN LABORATORY
SUPERCOMPUTING CENTER

DISTRIBUTION STATEMENT A. Approved for public release. Distribution is unlimited. This material is based upon work supported by the Under Secretary of Defense for Research and Engineering under Air Force Contract No. FA8702-15-D-0001. Any opinions, findings, conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the Under Secretary of Defense for Research and Engineering.