

## sdmay18-09: Tool Support for Continuous Model-Based Verification of the Linux Kernel

Week 10 Report

November 15 - December 2

### Team Members

Srinivas Dhanwada — *Team Lead*

Collin McIntyre — *Tool Integration Lead, Scribe*

Benjamin Weno — *Automation Lead*

Matthew Wall — *Web Lead*

---

### Summary of Progress this Report

For this period, we found and fixed some minor bugs in the locking instance mapping algorithm and found that our algorithm can accurately map 98% of all locking function calls in the kernel. This greatly exceeds our expectations of roughly 80% coverage. We've begun modifying the algorithm to accept previously generated output as input to the algorithm for the creation of extended maps across several versions of the kernel. We've continued to implement our patch generation algorithm and are currently able to generate the skeletons for the patched header files and parse the existing header files to retain all of the information for locking functions and macros. This information includes function/macro names, return types, number of parameters, and types of parameters. We still need to analyze this information to fill the body of the patched header files, then remove existing locking function declarations and implementations. We've started to research ways to automate L-SAP itself, and have discovered that we don't need a plugin for Atlas and can run L-SAP from a command line. Lastly, we've also begun implementing the back end of our website using Angular and Typescript to help modularize website design and creation.

---

### Pending Issues

Macro declarations simply accept the name of a variable as a parameter, but those variables don't have types. This is creating some difficulty in generating the content of the patch since we aren't necessarily sure which macro parameter should be mapped to each locking function parameter.

---

### Plans for Upcoming Reporting Period

We plan to implement our difference summarizer, which is an extension to the locking instance mapping algorithm that parses the output from the mapper to create a file or set of files that the website can use to display mapping results. We also plan to continue working on patch generation and specifically aim to add the capability to generate the body of the patched header files. Lastly, we plan to continue implementing the back end of our website.

---

### Individual Contributions

Team Member	Contribution	Weekly Hours	Total Hours
Srinivas Dhanwada	Srinu found and fixed some minor bugs in the implementation of the locking instance mapping algorithm and looked into a way to run the mapping algorithm using results from previous runs. He discovered that the	8	81

