We are roughly on track with the project so far. Currently, we have gotten the language-c parser library working, and we have developed some initial code to walk the AST and identify call sites of specific functions (currently it's hardcoded to uses of the "pickme" function, but we can easily extend that). Our next major task are to develop a partial dataflow framework so that we can detect indirect uses of specified functions. To evaluate that, we are also working on a couple simple self-timing that can detect things about a system simply by measuring performance – for example, a program which spawns several busy-loop threads can detect when another resource-intensive program is started. Similarly, a program which allocates an L2-sized large amount of memory and repeatedly reads through it can detect when another program has a large working set – with very precise timing and by influencing the branch predictor, this is the basic attack pattern followed by Spectre. These programs have not yet been written, but we have their designs sketched out.

Our project lives at https://github.com/jpdoyle/static-analysis-project