Name: Christopher Theisen

Hard Problem Area: Security Metrics – Attack Surface

Overall Research Goal(s):
*What do you ultimately hope to show with your research? This section can involve some jargon, but try to relate these goals to your broad impact section. Start with the larger goal(s) and narrow the scope towards your specific study (and specific goals, below).*

The goal of this research is to aid software engineers in approximating the attack surface of software systems by developing metrics based on crash dump stack traces.

Broad Impact:
*Why is your research important? This section should avoid any technical jargon and should be meaningful to the general public. Try to keep this down to five sentences. This should be hierarchical: the broad impact decomposed into more specific impacts that connect your overall research goals to your more specific goals.*

Deciding what code to review is an important issue for security professionals, as resource limitations prevent them from reviewing every change developers make. Prioritization techniques come in many different flavors, including identifying the attack surface of the system, or the code that is reachable by outside entities. In this work, we use crash dump stack traces to develop a series of metrics to assist in the prioritization of security efforts.

Specific Research Goals:
*Lay out the steps you are going to take to achieve your overall research goal. You can get technical here.*

We developed three classes of metrics in this work. First, change metrics identify what code has been added to the attack surface, removed from the attack surface, and has remained on the attack surface between versions and over time. Second, complexity metrics identify the number of “fan-in” and “fan-out” calls from a specific code artifact, which we then combine with vulnerability rates. Finally, we can identify the boundary, or edge, of the system by looking for the point in the stack trace where it transitions from third party code to system code. We then determine of vulnerabilities are more likely to be on the boundary.
**Proposed Data Collection (if applicable):**

What data will you collect to answer your research goals? How will you collect it? Will it be an observational study, randomized comparative experiment, or simulation study? Include potential biases and be prepared to explain how the data will achieve your specific research goals.

We collected 24.5 million crash dump stack traces from Windows 8, 8.1, and 10, from 2014 and 2015. These crash dump stack traces were collected from the CODEMINE system at Microsoft during an internship, and made available after the internship via a data sharing agreement. This is an observational study. This data will be used as an input to generate the metrics described above.

**Success Criteria:**

How will you determine whether you satisfied your specific and overall research goals?

After generating the metrics, each will be evaluated to see if they would provide substantive feedback for security professionals. For change metrics, this would be defined as identifying a substantial change in code covered by crash dump stack traces across versions and/or time. For complexity metrics, this would be defined as a correlation between the complexity measures and vulnerability density. For boundary, this would be defined as a higher density of vulnerabilities on the boundary versus the rest of the codebase.

**Anticipated Difficulties, Limitations, and Criticisms:**

What will make the above specific research goals difficult to achieve? How do you plan on dealing with these difficulties if they arise?

Anonymization of data is an issue, and we will need to work with Microsoft on the final format of the data reporting. Additionally, each of our metrics may or may not prove to be useful in practice, and properly expressing the negative result will be a challenge.