Intra-process data-flow tracking

Master Thesis

Supervisors: Prof. Dr. Alexander Pretschner, Enrico Lovat
Email: enrico.lovat @ in.tum.de
Phone: +49 89 289 – 17885
Starting date: immediately
Prerequisites: Basics of Operating Systems, Basics of low-level programming (Assembly)

Context and Goal
Usage Control is an extension of access control that concerns what must and must not happen to data after access to them has been granted. In [1] we describe a generic architecture for Usage Control and some exemplary instantiations of it, among others at the level of operating system. At this level, events are system calls, and this allows for a conservative estimation of the sensitive content read/written by a process, according to its interaction with the system resources (e.g. I/O devices, files, etc.). See [2] for further details. Unfortunately, this approach does not allow the monitor to observe how data flow within the process. This is reflected in a sensible over-estimation of the data dissemination. For instance, if a process reads content from a sensitive file, everything that is subsequently written by the process may possibly be sensitive as well, and so is how it should be treated. To overcome this limitation, a monitor that observes how (sensitive) data flow inside processes is needed: every time a process outputs some content (e.g. writes to a file), such monitor shall communicate to the operating system which (sensitive) data is actually being output.

Description
The student is supposed to conduct a survey of existing technologies for runtime monitoring of binary applications, like [3]–[7], and to build a monitor for runtime data-flow tracking based on one of them. Such monitor should be combined to our previous solution for usage control based on system-call interposition, with the goal of improving its precision as described above.

Submission of the following is mandatory at the end of the assignment:

(i) Dissertation, which includes:
   a. Comparison of different monitoring technologies and discussion of the chosen solution
   b. Analysis of the implementation in terms of precision and performance
(ii) Raw (commented) code of the implemented solution
(iii) Virtual machine(s) for demo of the work
(iv) Documentation explaining the set up and usage of the code and the virtual machine(s)

Workplan
1. Read and understand the documentation about Usage Control at the Operating System level
2. Familiarize with the existing implementation.
3. Compare existing solutions for runtime data-flow tracking in literature
4. Implement one of them and combine it with the existing usage control framework
5. Evaluate it in terms of precision and performance
6. Write a dissertation
7. Submit the work as explained above.
References


