Exploring Multiple Execution Paths for Malware Analysis

Andreas Moser, Chri Kruegel, Engin Kirda*Technical University of Vienna*

Rate: ★★★☆☆

September 12, 2013

It explore extra path by dynamically tracking the input data with taint. The system can detect the branch condition with those taint input (such as current time, file, or the result of a check for internet connectivity), whenever those taint input is use, it will copy the memory state and restore these states later. The idea is very similar to the symbolic execution in klee and s2e. The only difference is this paper use the native memory manipulation (data structure and page manipulation) in order to explore the extra path, which is fork in klee and s2e implementation. Specifically, it maintain three different components: (1) sets of taint input source, (2) shadow memory, (3) extension to the machine instructions to propagate the taint label.

Their Design

This paper recorded the system calls invoked by the binary and their arguments, all this is done in QEMU. The system work in the following ways: (1) Taint input and dynamically track the input data. (2) Once the input data is use for control decision, make a snapshot of current memory, (3) Take one of the path to execute until cover full path, (4) Start another path that backuped, (similar to depth first search). Contents of Heading on level 1.

Specific Design Issues Mentioned

For (1), To tracking the taint, the system only support linear propagation, that means when the input related data involving in plus and minus, the label is propagated linearly. This is not true for multiplication. Also, once a memory locations label has been modified, all the other memory location related to this memory should be updated, This is also the main reason why we copy the processs whole memory address in the snapshot. To remember, there is a constraint system that maintain all the relations between labels.

For(2), The saved snapshot include the complete current virtual memory contents of the running process. Another more important point is that we also have to take into account the conditional operation itself, this is help to pick the next condition to run the code in future. In the paper, it take labels into a constraint system, which mean not only the input data has constrain, the label of the input data also has constraint. Otherwise the alternative branch cannot be explored.

For (3) and (4), it is very straightforward.