CS 8803: Exploring Multiple Execution Paths for Malware Analysis Response

Yacin Nadji

August 22, 2016

Malware is often analyzed by tracing its execution in a virtualized environment. This allows malware researchers to figure out the behavior of malware that may be difficult to statically analyze, for example a packed binary, without allowing compromise of a physical machine. Unfortunately, this only traces a single execution of the malware, which may not include the malicious behavior researchers are trying to analyze. This paper aims to solve that problem. Input data (data pulled from syscalls for example) is tainted by the virtualized environment. If this input is part of a control flow decision, the environment takes a snapshot of the process and continues execution. When the program would normally terminate, any outstanding snapshots are then handled. The program is reset to the previous snapshot, the memory is altered to force the other branch, and tracing the execution resumes. This is done for every snapshot. The evaluation mentions several nifty findings, for example, some malicious programs will look for other instances of itself running on the infected machine. Using multipath exploration, researchers would be able to uncover this behavior without having to blindly guess sample inputs.

It's a cute paper that tries to address a serious shortcoming of dynamic executable analysis, execution coverage. The approach is pretty straightforward and seems to generate reasonable results based on the paper's evaluation. However, it isn't entirely clear how long the approach will be useful. It seems as malware becomes more complex, the number of execution paths will blow up, making this kind of analysis difficult to do. Furthermore, if malware authors are aware of this technique, it seems trivial to artificially add execution paths to force this combinatorial explosion.