Justification for Inclusion

Average Case vs. Worst Case—Margins of Safety in System Design

Category: Research Paper

Andreas Gal Christian W. Probst Michael Franz

Systems we currently use are mostly optimized for the *average case*. This makes them extremely vulnerable to any kind of attack that causes them to exhibit their *worst case* behavior.

Based on this observation (and on several examples we present in the paper), we formulate the new security paradigm

System design must be based on the worst-case behavior of the system being designed.

While there already are areas in which systems are designed for the worst case, we contend that this needs to happen for all systems that in any way are accessible by third parties, that is can be attacked from the outside.

We show that one essential factor in enabling worst-case behavior based attacks is the intimate knowledge of the system being attacked. For many hardware systems, this knowledge can not be easily obtained, since the system usually is embedded in packaging. In the realm of software, however, open source initiatives cause a more difficult situation. While in the general case the open-source statement "attack points are easily spotted if thousands of developers around the globe inspect the code" is certainly true, it does not hold with respect to worst-case behavior, which is due to algorithmic properties. Open source reveals the algorithm, making the design of an attack possible. The main insight is that *bugs* can be fixed easily, but *algorithms* are difficult to replace.