Is the Trusted Computing Base Concept Fundamentally Flawed?

John McLean
Center for High Assurance Computer Systems
Naval Research Laboratory
Washington D.C. 20375

Proposition: The Trusted Computing Base concept as a basis for constructing systems to meet security requirements is fundamentally flawed and should not be used to justify system architectures.

Arguing for: Bob Blakely, IBM
Darrel Kienzle, UVA

Opposed: William R. Schockley
LT James P. Downey, NPS

Discussion

This debate was conceived by Carl Landwehr, who asked me to serve as moderator. The proposition under discussion was first put forward by Bill Wulf in the JavaSoft Forum 1.1 (http://java.sun.com/forum/securityForum.html). Wulf objects to the JavaSoft security model, which aims to protect user files, network connections, and other resources from malicious applets by restricting applets to executing in a "sandbox". Wulf states:

You call the model a sandbox. In the literature it is called the "trusted computing base", or TCB. We might also call it the Electronic Maginot Line for reasons that I will try to explicate. Basically, however, the notion is that it is possible to interpose the TCB between subjects (applets, in this case) and the resources they wish to use. The TCB consults some trusted data base and decides whether the access is to be allowed. Assuming that the TCB is correct (and not penetrated), security is enforced. The sandbox model pervades the thinking of the security community - from the "rainbow books" to the concept of firewalls and beyond. The model is so deeply ingrained that no one questions whether it is valid. In my opinion it is profoundly flawed; so long as it is at the base of our thinking, we have no hope of building secure systems.

The sandbox model has the same problem as the Maginot Line. However formidable it may be to attack, once it is breached the battle is lost completely and irrevocably. Moreover, as the Germans demonstrated, the way to defeat it is to use an attack other than the one(s) anticipated. So long as the sandbox is the dominant model of computer security there will be no security - at least not against a determined attacker.

Most of the interesting penetrations of computer systems have not exploited bugs, but rather used some feature that had been carefully designed into the system - they just used it in a way that the designer didn't anticipate. With 20/20 hindsight, I am dubious that we could ever have guaranteed that there was no feature that some bright, determined attacker couldn't exploit in ways that we hadn't anticipated.

There has some good work over the last 20+ years, but a lot of it needs to be re-examined and disentangled from the gatekeeper assumption, and some approaches need to be scrapped. New approaches and ideas need to be explored. Flawed infrastructure needs to be replaced. An alternative framework needs to be fleshed out and deployed. A path for legacy systems to be absorbed needs to be developed.

Wulf's concern is not simply JavaSoft's use of sandboxing, but the gatekeeper assumption that, in Wulf's view, serves as the foundation for over 20 years of security work. It's unclear whether Wulf believes that we should abandon the TCB entirely or that we need to place it in a larger framework where the TCB is only one of many security mechanisms. He certainly believes that over-dependence on the TCB has stifled new approaches to security and left us with systems that are not as secure as we believe them to be. Has the TCB mentality left us with nonsecure systems? Are there better approaches which are not based on TCB-style separation? This debate is intended to resolve these issues.