Terra: A Virtual Machine-Based Platform for Trusted Computing

I found a number of powerful and interesting ideas in this paper. It's certainly true that desktop operating systems offer little or support for trusted computing, and the public Internet is a very scary place indeed, at least in terms of security. Thus, the idea of using a thin software layer, situated between the hardware and the operating system seems like a solid approach to the problem. We have seen several examples in the course of VMM layers that consist of just a few thousand lines of code, as compared to mainstream operating systems that runs to many millions of lines.

The idea of a trusted virtual machine monitor makes a lot of sense to me, as well as the requirement that the attestation of trust start with the hardware itself. We have already discussed “layer below” attacks, and I recently read something on the Web regarding network intrusions during which attackers were apparently able to compromise the firmware of a target machine. Thus, a tamper-proof hardware platform would seem to be a basic requirement for any trusted computing platform.

Naturally, a TVMM would need to incorporate many of the features we have already discussed for VMMs: strong isolation guests, efficiency of operation, and good compatibility with existing (guest) operating systems. In addition to these features, a TVMM would also need to provide a “root secure” environment for guests, support for strong attestation from tamper-proof hardware up to the guests and beyond, and a support for trusted paths for networked applications.

As the article rightfully points out, a TVMM such as that described in this article raises some privacy and consumer protection concerns, some of which are partially addressed in the article. As the article states, “The attestation process completely identifies the machine during the attestation...” While this fact raises some serious questions, it may be that the group signatures approach also discussed in the article would go some distance toward resolving privacy concerns. There is not a full enough discussion of this topic to allow evaluation possible problems or vulnerabilities, however. This is an
Equally as troubling are the potential interoperability and consumer protection problems associated with TVMM. The article seems to indicate that given “today's open distributed systems,” consumers enjoy a lot of freedom. The authors seem to be ignoring the fact that Microsoft controls roughly 90% of the desktop market, which has, of course, had a tremendously negative effect on consumer choice and innovation in the desktop market. A number of features in the new Vista system related to device driver certification and support for DRM clearly indicate Microsoft's preference for guarding the rights of copyright holders over those of consumers. In addition, Mr. Rosenblum's company has recently begun staking out a very Microsoft-like stance in the virtual machine market. I guess consumers ought to be grateful that Microsoft and VMware are shaping up to be competitors, rather than forming some sort of cabal around virtualization and the desktop.

As usual, the attestation/trust devil is in the details. When it comes to storage, encryption of data provides protection for guest operating systems and their applications. In addition, immutable data (OS and application binaries) stored on disk can be attested, but the actual data cannot be. The article discusses a couple of approaches to implementing attestation to minimize the overhead associated with it. These include ahead-of-time and optimistic attestation. Each approach has its pros and cons.

Device driver security is, of course, is large area of concern and has yet to be addressed in the prototype. Some existing approaches are discussed, including that of VMware Workstation's hosted architecture, which provides a simple interface between the host OS's untrusted drivers and the trusting OSs. Whatever assurances might be gained, as the article points out, a trusted path cannot run through a TVMM that relies on untrusted device drivers.

Overall, while I find the ideas introduced in this article to be intriguing, it's quite clear that much work remains to be done before a trusted computing platform can begin to address the many open questions and problems raised in the article. It might be much easier to introduce such a system on the server side, where there is currently much more diversity and interoperability among systems.
and where the device/driver market is somewhat less chaotic than it is in the desktop world. Perhaps some open standards could be further developed that could eventually be leveraged on the desktop side.