Pentium III Serial Number Is Just a Tool

But Is It a Can Opener or a Gun?

Intel opened a can of worms recently by announcing that, starting with the forthcoming Pentium III, each processor it produces will include a unique serial number. Intel hopes this number will provide a useful tool for asset tracking and securing Internet transactions. But its value appears limited, and it could potentially be misused.

The processor serial number (PS#) is a 96-bit value that is electronically programmed into the chip (as in a PROM) during the manufacturing process. It is read using an extended version of the current CPU_ID instruction, a "ring 3" instruction that any application can execute. On current Intel processors, this instruction provides 32 bits that identify the type of processor, but on Pentium III, this information will be concatenated with a unique 64-bit number.

Intel touts two main applications for this feature. One is asset tracking. An organization can use the PS# to identify PCs across a network. Information about the PC (location, configuration, applications, etc.) can then be associated with the PS# in a database, allowing simple tracking. This is done today by storing an asset number on the hard drive or using the unique network address. I see little improvement in using the PS# for this purpose.

Improving Internet security is a better use for the new serial number. Web sites such as Amazon could allow you to deny access to your accounts except from a particular PS#. This could help prevent unauthorized access by a third party. You could register all of your PCs if desired and add a new machine when you upgrade. Alternatively, you could create promiscuous accounts if you don't want to or can't use the PS# mechanism.

If hackers obtain your login and password to a particular Web site, the PS# could prevent them from getting into your account. Assuming your account is locked by a PS#, hackers would have to find that number as well. Even if they get the right PS#, they must still spoof the Web site into accepting it. This is a fairly simple hack if the Web site reads your PS# by making a function call from an applet.

To prevent spoofing, Intel recommends that the applet execute the CPU_ID instruction directly. Since the applet exists on your system only long enough to execute, it is more difficult to hack. Because the CPU_ID instruction is not executable by the Java Virtual Machine, it triggers a warning to the user via a Verisign dialog box. This box gives you the option to refuse that applet, thus protecting your PS#. Pure Java applets do not trigger this warning.

Privacy groups are concerned that any Web site you visit can obtain your PS# and collect data on your actions, perhaps even sharing this data with other Web sites to create a detailed personal profile. The Verisign protocol lets you prevent Web sites from reading your PS# by asking a fundamental question: "Do you trust this Web site?" A "yes" answer permits the Web site to download any applet and potentially obtain your PS#; at that point, you must trust that site not to misuse the number.

A "no" answer protects your PS#, but it could block the execution of other interesting applets (for example, ones that use special Pentium III instructions that accelerate multimedia performance). Another way to protect your PS# is to use an Intel utility to disable the PS# register. But to keep software from surreptitiously enabling the PS#, the processor cannot reenable it without rebooting. Thus, once you disable your PS#, you can't use it for secure transactions.

If your PS# is enabled, any application on your system will have access to it. Some software vendors may use this identifier for copy protection, refusing to execute on a processor that is a Pentium III (which is readily apparent using CPU_ID) but won't disclose its PS#. But this approach would probably create enough ill will and customer-support headaches to dissuade software vendors, even large Redmond-based ones, from trying it.

The surprising thing is that Intel has managed to create its own public-relations mess. Privacy advocates, many of whom don't understand the technical details, have condemned Pentium III. An Arizona state senator even proposed banning the processor. Intel should have gotten the browser vendors to publicly back its security plans and prebriefed technical analysts to help communicate the message.

The PS# is merely a tool; Web sites and software makers must choose whether to use it wisely or harmfully. "Trusted" Web sites must not secretly obtain or exchange processor serial numbers, or they will not be trusted for long. Software vendors that use the PS# in poorly implemented and inconvenient copy-protection schemes will also suffer a backlash.

If vendors act appropriately, the PS# won't compromise privacy. Despite Intel's precautions, however, hackers will probably find a way to forge these numbers. As a result, the PS# gives your personal information an extra lock that is only slightly more secure than the current ones. M