The Trusted PC: Skin-Deep Security

Andrew "bunnie" Huang, Xenatera Partners

The rapid growth of the World Wide Web and advances in networking technology have made it more important than ever to secure personal computers and operating systems. Individual users as well as enterprises need to know that the systems they are using will not divulge personal or copyrighted information to hackers or accept viruses, worms, Trojan horses, or unsolicited e-mail that can slow or damage systems.

To address this problem, a consortium led by Intel formed the Trusted Computing Platform Alliance (http://www.trustedcomputing.org/). Cofounder Microsoft recently announced Palladium, a parallel implementation of the trusted PC that may be available next year. Unfortunately, the TCPA approaches generally, and Microsoft's in particular, offer a robust solution against software but not hardware attacks.

This weakness is a design tradeoff based on the assumption that mounting a hardware attack is too costly for individuals. However, a study of Microsoft's Xbox gaming console (http://web.mit.edu/bunnie/www/proj/anatak/AIM-2002-008.pdf), which implements security techniques parallel to those proposed for Palladium, indicates just how easy it can be for an end user to penetrate PC hardware.

INSIDE THE XBOX

The Xbox is essentially a PC with small hardware enhancements that nominally make it impossible to access and modify the console's kernel via a software-only attack. However, once the cover is off the console, extracting the key and algorithm to decrypt the kernel is a fairly straightforward task.

Current proposals to secure the personal computer will leave users with two undesirable options.

A read-only memory chip in the southbridge application-specific integrated circuit stores the core crypto routines that protect the Xbox. The electrical and protocol details of the high-speed internal bus that transmits these routines to the CPU are easily inferred by comparing the console's hardware to well-documented PC hardware and the HyperTransport bus standard (http://www.hypertransport.org).

Observing traffic on this bus provides the information necessary to decrypt and encrypt kernel images. I custom-built the equipment required in less than three weeks for about US$50; you could also rent a piece of stock test equipment capable of extracting data from the bus for less than US$500 per month.

The hardware's power-on initialization procedures contain other back doors, which let users indirectly obtain the kernel plaintext as well as gain control of the console's program counter.

These weaknesses make it possible to leverage the Xbox's test points and diagnostic ports to implement cheap hardware attacks. They also underline the challenge of securing a platform designed to be open and user-serviceable, with little concern for hardware security.

SEALED STORAGE

Palladium and other TCPA-compliant platforms employ a technique that can potentially defeat the simple kernel patch or substitution attacks that are effective on the Xbox. *Sealed storage* guarantees that the relevant system state—including the operating system and trusted helper programs—is identical at data storage and data access times (http://vitanuova.loyalty.org/2002-07-05.html). If the current machine state's hash does not match a copy of the expected system state sealed inside encrypted data, the tamper-resistant module managing the machine verification process prevents further decryption.

One application of sealed storage is in digital rights management (DRM). Here is a simplified example of how it might work, based on the "'Web Administration" section in the TCPA's *Credible Interoperability* white paper (http://www.trustedcomputing.org/docs/Credible_Interoperability_020702.pdf).

A content provider, Bob, requires that Alice, a consumer, use a PC that implements a trusted architecture. This PC runs Bob's content viewer, which will enforce a pay-per-view management

policy. When Alice requests content from Bob, her trusted PC's secure cryptomodule sends digitally signed credentials to Trent, a trusted authentication authority who keeps records of every PC's cryptomodule signature.

Trent verifies Alice's signature and machine state hashes. If they check out, Trent can assert that Alice's PC is running an unpatched operating system and has Bob's unpatched content-viewing software. Once Bob is satisfied that he can trust Alice's PC, he sends her the requested content. The receiving application immediately encrypts and seals the data on Alice's hard drive. The operating system on Alice's PC is now responsible for managing Bob's content-viewing software in a manner consistent with Bob's intended policies.

The operating system and trusted PC-aware applications running under it can use sealed storage to enforce any content-access policy, including policies that limit user choices. Alice can never freely access her copy of Bob's content with a third-party browser because her computer's secure cryptomodule encrypts the session key. She cannot patch the operating system or Bob's browser either because the cryptomodule checks the machine's state before decrypting the data.

Further, Alice cannot send Bob's content to a friend or use a second PC to restore backed-up hard-drive data because each trusted PC has a unique secure cryptomodule. The cryptomodule would also detect any intruders during hash calculation, preventing a hacker from stealing Alice's plaintext copy of Bob's content.

HARDWARE ATTACKS

Although sealed storage is effective against software attacks, the unsealing protocol is weak in the face of some simple hardware attacks.

SPAM

One such attack involves the use of *schizophrenic access memory*. SPAM fits in a standard dual inline memory module (DIMM) socket that contains a reasonable amount of memory and a field-programmable gate array. The FPGA sits between the DIMM interface and the memory chips and can present the system with slightly different memory images.

During the secure cryptomodule's machine-state inspection, SPAM presents a correct, unmodified memory image; at all other times, it presents a patched memory image that lets users violate their system's security policies. SPAM would be fairly inexpensive to manufacture in quantity; the FPGA would add perhaps US$50 over the base cost of a DIMM—not a bad investment for a lifetime of discount content.

Sealed storage is effective against software attacks, but it is weak in the face of some simple hardware attacks.

SPIOS

A slightly less expensive but also less effective attack uses a schizophrenic basic input/output system (SPIOS). A user can boot up using a secure version of the BIOS, load content from the hard drive into memory, and then perform a soft-reset of the system while switching the BIOS image. The soft-reset will leave most of the system RAM intact, allowing recovery of plaintext content for later use. Because SPAM and SPIOS are both legitimate debugging tools, categorizing them as circumvention devices, especially if SPAM is shipped with an unconfigured FPGA, may be difficult.

Problems

Most defensive measures against such attacks would negate the PC's greatest asset—inexpensive, convenient, high-performance hardware—and thus be impractical. For example, placing a cipher between a processor's L2 cache and main memory would significantly inhibit performance. Also, integrating an entire PC, including main memory, onto a single piece of silicon is unlikely to be economically feasible any time soon.

Tamper-resistant cases and potting would be unpopular with users, many of whom like to occasionally open up their computer, and could create thermal problems. A suitable membrane that can detect intrusion and modify PC behavior, even when it is unplugged, is likewise prohibitively expensive. Even if an affordable and effective solution is developed, users could still point a video camera at the monitor; hold a microphone up to the speakers; or use a printer and scanner to make free digital copies of videos, music, and text.

IT CUTS BOTH WAYS

In light of these weaknesses, the wisdom of the trusted PC initiative is open to question. Is it worth the time, effort, and money to develop a system that can be cracked for a song?

The trusted PC is a double-edged sword that can be used against as well as on behalf of the consumer. Ciphers protect secrets, but only intelligent, well-informed policy choices can protect consumers' rights.

Interoperability

In the long run, abuse of the authentication infrastructure would likely prevent interoperability with newer, less expensive applications or content. Although the trusted PC architecture is open, with anybody free to create and sign software, secure data sharing between applications requires establishing a trust relationship.

The process of creating and managing these trust relationships and the cryptographic rights to share data with established trusted-PC-aware applications is likely to be expensive and politically charged, with application vendors obligated to inspect all interoperability candidates for Trojan horses and back doors prior to trusting them.

Personal freedom

Beneath the promise of a more reliable, secure computing experience lies the imminent threat of individuals losing their fair use rights. It is easy to imagine a greedy digital content provider leveraging the trusted PC architecture to enforce draconian DRM policies or revoke users' ability to access content at an arbitrary date.

For example, suppose Bob distributes to trusted PC users a data-compression utility that gains wide acceptance because it is free. Bob is a clever guy—he wrote the utility so that it requires an authorization command obtained from his servers. Once his free utility has put all of the competing pay utilities out of business, Bob changes his authorization policy to require users to pay a dime every time they invoke the utility.

With a healthy revenue stream from users who have no choice but to pay for the software because they want to recover their compressed data, Bob hires programmers to improve his utility while at the same time he is buying out any newcomers to the data-compression utility space, giving him a total monopoly on such utilities.

POSSIBLE SOLUTIONS

To employ today's high-level protocols to perform secure transactions, it is first necessary to ensure that attackers cannot hack machines. Fortunately, two simple tried-and-true architectural techniques are available that provide security in computer systems without using cryptography. You can use these techniques in conjunction with many of the secure online transaction features promised by Palladium and other TCPA-compliant platforms to provide a nominally trustable system in the face of malicious third-party attacks.

Guarded pointers

Hardware-enforced capabilities, or *guarded pointers,* as used in the M-Machine (http://cva.stanford.edu/m-machine/cva_m_machine.html), compartmentalize and secure computers against rogue programs by tagging pointers with a few extra bits of bookkeeping data.

You can efficiently implement guarded pointers on 64-bit architectures, as the hardware rarely uses all 64 bits of address space to keep track of memory. For example, the 64-bit Alpha 21164 implements a 43-bit virtual address and a 40-bit physical address, while AMD's 64-bit Sledgehammer implements a 48-bit virtual address and a 40-bit physical address.

Combining guarded pointers with data tags makes an operating system more robust.

As part of Project Aries at MIT's Artificial Intelligence Laboratory, my colleagues and I defined a memory-efficient (less than 6 percent internal fragmentation) guarded-pointer scheme that provides exact object bounds and subobject security with only 16 bits of pointer overhead (http://www.ai.mit.edu/projects/aries/documents/memos/aries-05.pdf).

Data tags

For even more protection, the hardware can keep track of the data pedigree using *data tags* that carry ownership and security information. Every operation on a piece of data will leave a unique hardware-enforced mark related to the operator's security class on the data's pedigree tag, so that rogue programs cannot read or modify data unnoticed (http://www.ai.mit.edu/projects/aries/Documents/Memos/ARIES-15.pdf).

Data tagging is economically viable today thanks to Moore's law. For example, error-correcting code (ECC) memory includes 8 extra bits of data per 64 bits, and the price difference between ECC and non-ECC 128-Mbyte double-data rate SDRAM (synchronous dynamic random-access memory) DIMMs is marginal.

Combining guarded pointers with data tags makes an operating system more robust against viruses and hackers as well as bad programming.

Current efforts to secure the PC's traditionally open architecture will give consumers two unattractive choices: They will either have to pay a huge premium for an unwieldy system that employs impenetrable membranes, encrypted buses, and tamper-resistant memory, or they will have to settle for an inferior solution that fails to thwart dishonest users and limits the ability to backup data and interoperate with third-party software.

Investing in proven architectural improvements such as guarded pointers and data tags is a more cost-effective and long-overdue alternative. ■

Andrew "bunnie" Huang is cofounder of Xenatera Partners, a technology development and consulting company based in San Diego, Calif. Contact him at bunnie@xenatera.com.