Designing and implementing malicious processors

Samuel T. King, Joseph Tucek, Anthony Cozzie, Chris Grier, Weihang Jiang, and Yuanyuan Zhou
University of Illinois at Urbana Champaign, Urbana, IL 61801

1 Introduction

It may be possible for attackers to modify integrated circuits (ICs) to insert covert, malicious circuitry into manufactured components; a recent Department of Defense report[1] identifies several trends that contribute to this threat. First, it is infeasible economically for government-based IC suppliers to produce technology that matches the performance of commercial suppliers. These high-performance ICs provide a tactical advantage making them an indispensable resource. Second, commercial suppliers are moving more design, manufacturing, and testing of ICs to a geographically diverse set of countries in an effort to cut costs, making it infeasible to secure these steps in the IC life cycle. Together, these trends lead to an "enormous and increasing" opportunity for attack.

Motivated attackers will subvert the IC supply chain if doing so provides sufficient value. Since modifying an IC is an expensive attack, it is doubtful that "script kiddies" will turn their adolescent energies to malicious processors, but the same cannot be said for attackers with resources. If malicious processors are capable of running valuable attacks, governments, terrorist organizations, and so on will deploy them despite their cost. Historically, these types of organizations are experienced at covert operations, and have demonstrated considerable ingenuity in pursuing their goals. In contrast, there is little work on malicious processors.

If an attacker were able to include a malicious IC within a computer system, it would give them a fundamentally higher level of control compared to software-based attacks. While the recent SubVirt project shows that attackers can gain control over operating systems by using virtual-machine monitors (VMMs) to control the layer beneath, ICs occupy yet a lower layer. A malicious IC would be below all software, including VMMs, so compromising ICs gives attackers complete control over the entire software stack. This high level of control provides attackers with a fundamental advantage over defenders running above.

In this presentation we will describe the design and implementation of intelligent malicious processors (IMPs) that run malicious services within the processor itself. Clearly simple attacks are possible (e.g., shut off the processor after one billion instructions), but we show that attackers can carry out sophisticated attacks using IMPs. We will discuss four example attacks we implemented and we show that general-purpose attacks implemented using IMPs are possible, practical, and qualitatively harder to detect and defend against than current software-based attacks.

To better illustrate our ideas, this presentation will include a live demo of a hardware-based attack. For our demo we will show one specific attack where we modify the design of a SPARC processor. We run our modified processor on an FPGA development board, and our system includes a full Linux distribution.

For our demo we will show a malicious service that acts as a permanent backdoor into a system. To use the attack, an attacker sends an unsolicited network packet to the target system and the target OS inspects the packet to verify the UDP checksum. The act of inspecting the packet triggers the trojaned hardware, and the malicious service interprets the contents of the packet as new firmware that it loads into the processor invisibly. The target operating system then drops the unsolicited packet and continues operation completely oblivious to the attack.

Our firmware monitors the `login` application looking for an attacker that logs in using the password "letmein". Once this password is detected, the firmware grants access to the attacker. At this point the attacker can then use any traditional methods of manipulating the system to avoid detection and to carry out malicious activities. The underlying mechanism we use to implement this attack increases the logic-gate count by only 0.08%, and gives us unlimited access to the machine without exploiting a software vulnerability.

[1] Defense Science Board Task Force On High Performance Microchip Supply, 2005.