Supporting Hardware Security Through Software

Device Attacks and Prevention Measures

WNDRVR

INTRODUCTION

When selecting hardware, it often comes down to functionality versus cost. However, the growing importance of security and privacy in a world of intelligent systems strongly impacts hardware; clients now consider hardware security capabilities as well as their constraints. And some hardware (a given medical device, for example) is not connected to the internet, which causes a different problem: Software updates can be difficult to implement. Any vulnerability on such a device will remain there until the firmware is manually updated — and such updates may never be available.

These challenges increase in devices with a long service life, such as a medical pacemaker or the control system in a nuclear power plant.

Hardware can also face challenges from those who refurbish old systems and resell them online. The reseller may not have properly wiped the device of all past data, or the device may have been hacked or could contain vulnerabilities that were never addressed. The use of nonvolatile memory to store program code and configuration data can also lead to security challenges during device recycling, as it can retain sensitive information such as login details. This hardware can easily be reverse-engineered.

This paper provides an overview of the different levels of attack on hardware devices, then reviews the various ways of using software to protect these devices.

TABLE OF CONTENTS

Introduction 2
Types of Device Attacks 3
How to Protect Devices 3
Common Hardware Platform Defenses 4
Conclusion 4

TYPES OF DEVICE ATTACKS

For better defense, it is important to understand how devices can be targeted. As Sun Tzu said, "If you know the enemy and know yourself, you need not fear the result of a hundred battles."

Starting from the lowest level, attackers can gain entry by attacking the boot process of a device. Under normal conditions, a device loads a boot loader whose job is to correctly configure the device as well as launch the main device app. Thus, attackers can sneak in malicious code during this step to run whatever programs they choose.

This brings us into the second level of attack: lack of authentication. A device that does not check the program that it is currently running cannot tell whether the program is authentic or is infected with malware. While authentication can be accomplished using a remote server, a device without an internet connection will not be able to undertake such attestation.

Another means of attack comes through bad implementation of cryptographic functions. Engineers who either design their own cryptographic functions from scratch or poorly implement commonly used ones can set a system up for risk. Generally speaking, only those with deep experience in computer science, mathematics, and security systems can reliably design security measures that are hard to beat.

Some devices may store private data on external flash chips with no encryption whatsoever. This means that an attacker can simply remove the flash chip from the PCB and read a device's entire content of memory. Such data includes Wi-Fi details, login credentials, passwords, private keys, and proprietary code.

Another common attack involves accessing programming ports. Many engineers bring out the programming pins of an IC to a PCB with exposed pads, and, in most cases, these programming ports are not disabled. This allows an attacker to hijack the IC using a programming device and potentially read the contents of its code memory, including configuration data.

HOW TO PROTECT DEVICES

The platform that a given device uses is of critical importance. The three major platforms currently used are Intel® x86, NXP PowerPC, and Xilinx Ultrascale. Each one employs multiple methods for protecting against different attacks.

Secure Boot

Secure boot methods prevent unauthorized code from being launched with the use of a signed boot loader. The process starts with boot loader code that is loaded into memory, and a signature is generated from this code. This signature is then compared with whatever is physically stored inside the CPU. If they match, it is then assured that the boot loader is authentic.

Attestation

Attestation is the step in which a device confirms that the program loaded in the secure boot stage is also authentic — there is no point in using secure boot if the program then loaded is riddled with malware. Because of the deterministic nature of most deployed hardware (the unallocated memory, constant space, and code space are always known on startup), a signature of these areas can be generated. This signature is then compared with a stored signature on the device, generally in a read-only program. If they match, the code is executed.

Cryptographic Processor

Cryptographic processors are dedicated pieces of hardware that perform cryptographic functions including encryption, decryption, key generation, and random-number generation. Such processors can be found either on-chip (internal to a processor) or as a discrete attached device (a separate module). Such modules should be used whenever possible, since software implementations of cryptographic functions can be hacked and poorly written functions can be exploited.

Internal cryptographic processors are generally preferred, as they cannot be accessed and their communication lines cannot be intercepted. External cryptographic processor modules can be accessed and even entirely removed in some cases, which means that external cryptographic modules are far more vulnerable to a hardware attack.

Random-Number Generators

As the name suggests, these devices generate random numbers. They are essential in strong security systems, because true random numbers cannot be guessed. An RNG can be used in key generation as well as nonvolatile initialization and data padding.

Physical Tamper Protection

Physical tamper protection systems are systems that monitor the physical condition of a device. They can be integrated into ICs and products alike. For example, a router could be integrated with a tamper detection system such that if the cover to the router were removed during operation, it would wipe external memory chips and shut down to prevent data theft. IC tamper protection also exists, usually using extra pads that are connected to a known signal. When the IC is lifted and powered externally, the tamper state can be detected and protection systems can be activated.

Hardware Fuses

Hardware fuses are configuration bits that are typically programmed only once during the manufacturing stage. These fuses can be used to store cryptographic keys, prevent external programming ports from being used, and fix boot locations. Such fuses can also be made inaccessible to the outside world so that any stored keys and other sensitive information cannot be read.

COMMON HARDWARE PLATFORM DEFENSES

Wind River® offers drivers and the VxWorks® RTOS for enabling security features on a number of platforms, including x86, NXP PPC, and Xilinx UltraScale.

NXP PowerPC Software Stack

Wind River offers security protection in PPC platforms with the use of multiple in-house drivers running on a Linux implementation. Secure boot is provided by hardware, while all other security features are provided in software. For example, the security engine (SEC) driver provides cryptographic functions, random-number generation, and secure key storage. The Runtime Integrity Checker (RTIC) provides attestation of the device during boot, and the Security Monitor (SecMon) provides physical tamper detection.

Intel x86 Software Stack

Intel x86 integrates a number of hardware security systems that Wind River fully utilizes. For example, secure boot is provided by Boot Guard and UEFI secure boot, while cryptographic functions, RNG, and secure key storage are handled by the Trust Platform Module, which is interfaced using the Wind River TSS2 driver. Attestation of code is done on the application side in software.

Xilinx UltraScale Software

Xilinx UltraScale protection relies on hardware systems and also on the use of drivers written by Wind River. Secure boot is hardware based, as is trusted boot; programming port protection is protected by eFUSE; and keys are stored using physically unclonable functions. The CSU driver provides cryptographic functions, while attestation is done via the application in combination with the CSU. The Versal driver provides random-number generation, while physical tampering is provided by the security monitor IP driver.

CONCLUSION

In terms of selecting devices for an application, Wind River highly recommends consultation before committing to any purchase, in order to access the best security services. Those purchasing devices should also be mindful of who is selling the device, since unauthorized sellers of refurbished equipment might either insert malware or provide inadequate protection against it.

Most devices have multiple points of entry, so always check on the platform used and the security features implemented, and ensure that updates are available. It is also a good idea to check on the number of support years that come with the device, so that future hardware replacements can be planned before updates and support cease to exist.

WNDRVR

Wind River is a global leader of software for the intelligent edge. Its technology has been powering the safest, most secure devices since 1981 and is in billions of products. Wind River is accelerating the digital transformation of mission-critical intelligent systems that demand the highest levels of security, safety, and reliability.