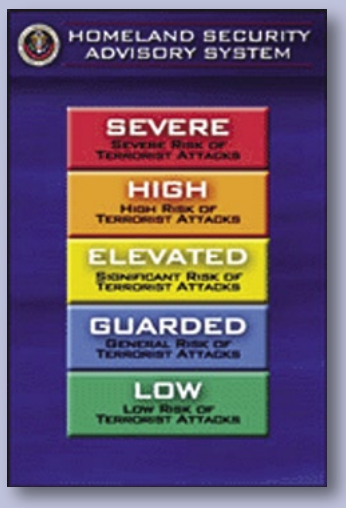
Homeland security and embedded devices



By Arun Subbarao

The events of September 11, 2001 and its aftermath have created a heightened awareness of security loopholes in every aspect of homeland security. One area that is highly susceptible to terrorist attacks is cyberspace. The ubiquitous nature and interconnectivity of the Internet make it an easy target for a malicious user. Furthermore, the proliferation of Internet-enabled embedded devices has created more opportunities for malicious users to exploit security weaknesses to gain access to sensitive information. It has become clear that there is an immediate need for more cyber-security within embedded devices.

One of the key components within an embedded device is the Operating System (OS). The security of an embedded device depends heavily on the ability of the OS to provide a secure environment for its applications. However, the design of most commercially available embedded OSs does not provide high levels of security. This article discusses the technical issues involved in designing and developing a "Secure Embedded OS."

A secure embedded OS

Most commercially available embedded OSs contain areas of vulnerability that contribute to the weakening of cyberspace security. For example, an embedded OS does not associate a *privilege* with the requester of information to ensure that the requester is entitled to the accessible information. This lack of access control capability is a significant area of vulnerability in today's embedded OSs. An embedded OS that addresses this and other vulnerabilities, thereby increasing the security of a system, is a *secure embedded* OS. The most fundamental feature of a secure embedded OS is its ability to control information flow using capabilities such as the following.

Access control

The access control feature prevents unauthorized users from accessing a system and from unauthorized execution of applications on the OS.

Information encryption

This feature stores and disseminates information, either locally or over the network, in an encrypted manner so only authorized applications can access it. An unauthorized application cannot corrupt or manipulate the information that is stored in or disseminated from the system.

Information isolation

The OS can enforce a policy where the same information has different levels of access for different applications. An application with higher privileges might have the ability to read and modify the information, whereas an application with lower privileges can only read the information.

These capabilities can substantially increase the security of embedded devices. However, several technology considerations in the design of a secure embedded OS also affect security.

Technology considerations

Some of the technology considerations for a secure embedded OS are:

- Feature trade-offs
- Security evaluation
- Standards conformance
- Hardware support

The implementation of such security capabilities as access controls, information encryption, and information isolation might affect key features of an embedded OS, like real-time response, throughput, and size. Designers should assess and minimize the impact of these trade-offs in the secure embedded OS.

There is a United States Government standard for information-technology security evaluation called Common Criteria (http://www.commoncriteria.org). The Common Criteria standard defines *Evaluation Assurance Levels* (EAL 1-7) that indicate the process rigor associated with the development of an information technology product. The desired EAL for a secure embedded OS should be assessed against the Common Criteria standard.

The ability of a secure embedded OS to conform to established standards is a critical element in enabling a critical

mass of security-aware middleware and applications executing in its environment. A secure embedded OS should conform to key standards like IEEE POSIX, ARINC, and RTCA/DO178B.

Embedded device hardware can range from large chassis to smaller form factor PC/104 boards as well as small custom boards. A significant challenge in the design of a secure embedded OS is ensuring support for the entire range of microprocessors and hardware used by embedded devices.

Companies such as LynuxWorks that already support the critical features of POSIX conformance, ARINC653 scheduling, and DO-178B certification are forging ahead to develop the industry's next-generation secure embedded OS.

Application considerations

There are many application areas for a secure embedded OS, including aviation, surface transportation, industrial control, financial markets, telecommunications, and government systems, to name a few. These crucial infrastructure areas communicate information that is critical for the safety and security of the nation and, as

such, have become increasingly Internet-enabled. For example, a truck transporting hazardous material may involve an embedded device that is communicating information about its cargo to a central database through the Internet. Only security-aware software can achieve the ability to prevent access to such sensitive information.

> "Companies ... are forging ahead to develop ... secure embedded OS."

Looking ahead

As embedded devices start to proliferate, the interconnectivity of these devices through the Internet creates enormous challenges for cyber-security, and perhaps even creates vulnerabilities that we have yet to recognize. The *secure embedded* OS is one important piece of the puzzle in our quest to make embedded devices more secure.

Arun Subbarao *is the director of software engineering for LynuxWorks, responsible for the LynxOS technology and product development. Arun has more than 11 years of software experience in the embedded industry working on UNIX, Linux and RTOS kernels, networking protocols, and high availability. He received his B.S.C.S. from India, M.S.C.S. from SUNY Albany, and an M.B.A. from Santa Clara University.*

For more information about LynuxWorks, contact the company directly.

LynuxWorks
855 Branham Lane East
San Jose, CA 95138-1018
Tel.: 408-979-3900
E-mail: asubbarao@lnxw.com
Web site: www.lynuxworks.com