Review of Applying the TCSEC Guidelines to a Real-time Embedded System Environment

Nathan Balon

SN# 3210 1717

CIS 544

Title

Applying the TCSEC Guidelines to a Real-time Embedded System Environment by Jim Alves-Foss, Deborah Frincke, and Gene Saghi

Introduction

The Department of Defense created the Trusted Computer System Evaluation Criteria (TCSEC) in 1985, as a means of assessing the security of a computer system. The Military produced a series of books called the rainbow series, and each has it own color for the cover. TCSEC is also informally known as "the Orange Book" because the cover of the manual is cover. The military envisioned various levels of trust in the "Orange Book" going all the way up the Trusted Computing Base that every line of code is mathematically proven to be correct [Cheswick, Bellovin and Rubin 2003].

Systems are evaluated on the level of security offered by the system and are given a rating. The ratings of the system are ranked from D, C1, C2, B1, B2, B3, to A1, with D being the least secure and A1 being the most secure. The security ratings give some assurance that the system is secure since it meets the guidelines developed in the TCSEC.

Problem

Many different types of systems need a way to define security. The typical security policy is defined for a computer with an operating system. Security is needed for more than just the typical operating systems on computers. Not all systems can be evaluated using the guidelines of TCSEC. With real-time embedded systems, the execution of code and the operating environment is contained in chips and doesn't use a typical operating system. Security policies and mechanisms are needed by more then the typical operating system. Real-time embedded systems also need a way to verify the security of the system. This paper explores the ways in which TCSEC can be modified to be used by real-time embedded systems. The authors of this paper explore ways that TCSEC can be modified to be used with embedded systems. In the paper the authors look at the ways that TCSEC can be used to evaluate the avionic system of the F11 aircraft.

Review

The paper ***Applying the TCSEC Guidelines to a Real-time Embedded System Environment*** deals with the ways in which TCSEC can be modified to be used with embedded systems. The avionic system of the F11 is explored. The F11's avionic system uses different guidelines to evaluate its security rather than a typical computer system. The F11 uses a distributed real-time embedded system.

TCSEC

The TCSEC was developed by the Department of Defense as method to evaluate a systems security. A system is evaluated on the level of assurance that the system provides. The system is given a rating, ranging from D to A, depending on the level of security that the system offers.

- Class D systems offer the minimum level of security for the system.

- Class C systems offer Discretionary Access Control and Auditing. Class C systems can be divided into C1 and C2 levels of security.

- Class B systems offer Mandatory Access Control. Class B systems can be further subdivided into B1 through B3.

- Class A systems offer verified security; it offers assurance as B3 but adds the additional assurance requirements that the system has been designed correctly.

Each of these divisions has different requirements. There are four areas of requirements; they are security policy, accountability, assurance, and documentation. Each class of security has different provisions that must be met for these requirements. The higher the rating the more strict the security requirements are for the system and the greater the level of security offered by the system.

Multilevel Security Policy

The TCSEC describes Multilevel Security Policies (MLS). The main purpose of describing this Multilevel Security Policy is to prevent unauthorized access to information. In a Multilevel Security Policy there are different security levels for the subjects and objects of the system. Lower level subjects are not allowed to observe higher level subjects. The system is then not allowed to access objects that would violate the security policy.

If the policy is not strictly enforced it is possible that covert channels can arise. Two ways that covert channels can develop are through the sharing of resources or with timing. If covert channels are developed it is possible that information can leak and the security policy would be violated.

Real-time Embedded Systems

The paper then explores what is a real-time embedded system. First, the authors explain that a distributed system is one that contains hosts, servers and a network through which the host and servers communicate [Alves-Foss 1996]. In a distributed system, security is important because there are many different users with different security levels accessing the system. Next, the authors define the real-time system explored in the paper, as one which provides mechanisms to ensure that the executing tasks will meet performance and

deadline criteria [Alves-Foss 3 1996]. It is important for a real-time system to complete the tasks of the system in a timely manner. Last, an embedded system is one that monitors and controls attached peripherals [Alves-Foss 1996]. Adding security features to a distributed real-time embedded system come at a great monetary and performance cost. It also looks at ways of reducing the security overhead of the F11 system.

The avionic system of the F11 uses integrated circuits. The system incorporates both data processors and signal processor. Because of cost constraints, it is possible for different tasks to use the same processor. An important point is that different processes with an assortment of security levels do not use the same processor at the same time and/or covert channels can be established. It is also possible for many different processors to be used at the same time with different security levels but they shouldn't share the same processor.

Differences between Systems

Differences exist between traditional computer systems and real-time embedded systems. For this reason, different methods should be established for measuring the assurance of the system. The authors explore the ways in which these two systems differ. In a traditional computer system the security is typically controlled by the operating system. The operating system enforces the access control policies of the system. Also, in a traditional system it is not uncommon for a user to create new objects in the system such as files.

A traditional system is very dynamic because new objects are created frequently. On the other hand, a real-time embedded system is much more static. It is highly unlikely that a user will create new files in an embedded real-time system. The system has well defined set of processes that will be used by the system. The authors state, the system may execute processes dynamically but these processes are well defined. These are a few of the differences that exist between the systems.

The cost of implementing security is much greater in a real-time embedded system. The authors state, that real-time embedded systems such as the F11's are moving from being proprietary to open systems. The avionics system discussed in the paper is much different then the typical computer system. Because of the differences and cost of implementing security the TCSEC must be modified. Below is a diagram of the avionics system of the F11 which was taken from the paper ***Applying the TCSEC Guidelines to a Real-time Embedded System***.

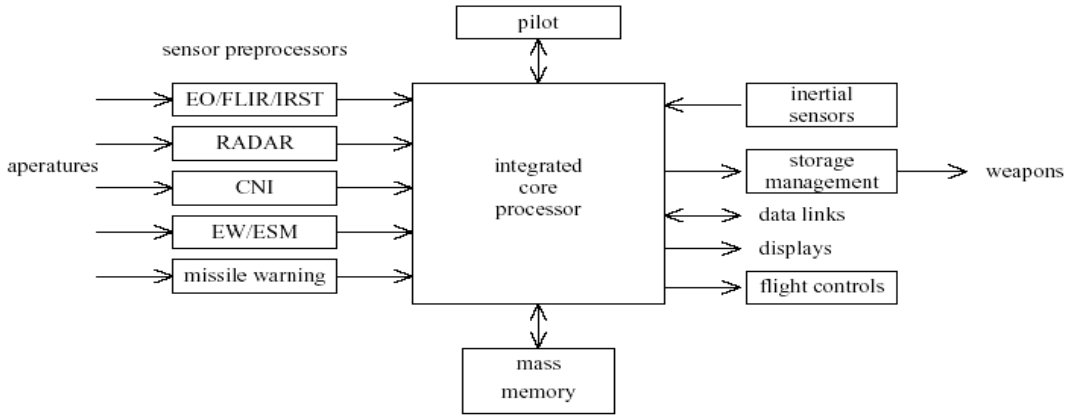

Fig. 1 F11 avionics system [Alves-Foss 1996]

Modification of TCSEC

The authors purpose ways of modifying the TCSEC. Because differences exist between the two types of systems, the TCSES standards for embedded real-time systems should be modified. The difference needs for a real-time embedded system should be taken into account.

TCSEC Components that Need to be Changed

Discretionary Access Control is not needed in real-time embedded systems. The security of the system can be enforced with Mandatory Access Control rather than Discretionary Access Control. The authors' state that in real time embedded systems there are well-defined process with well-defined roles. In typical system it is not known in advance what objects will be created by the users of the system, so Discretionary Access Control is necessary. For real-time embedded systems the objects that will exist in the system are known in advance. There is no need for using Discretionary Access Controls in a real-time embedded system.

Identification and Authentication is another area where the TCSEC needs to be modified. In this paper, they assume that the system will not be physically tampered with. Also, the typical authentication for a military pilot to access the F11 is by walking past armed military guards. So it is unlikely that the pilot will need to be authenticated to the F11's avionic system.

TCSEC Components that May Need to be Changed

There are also some components of the TCSEC that may also need to be changed depending on the situation. They are subject sensitivity labels, trusted path, and audit. First, the authors declare that subject sensitivity labels may not be necessary if they are

defined by human users because only one person, the pilot, will interact with the system. The labels may be needed when a real-time embedded system has more then one subject. Second, the trusted path may or may not need to be used depending on the trust of the vendor components. Last, the type of auditing that is done by the system may need to be modified. The authors state that auditing should probably only be done for maintenance and performance checks and validating vendor components. It may not be necessary to audit records from the users of the system.

The rest of the TCSEC guidelines should remain unchanged. The table in the appendix Fig. 2 shows the solution the authors propose to modify the TCSEC for real-time embedded systems.

Discussion

The TCSEC guidelines should be changed for certain types of systems. Not all systems should have to follow the same requirements to be assured of the level of security provided by the system. A system should not have to follow the same guidelines when the criteria are not applicable.

In the future, as processing power increases, some of the features that the authors suggest eliminating may be incorporated back into the real-time systems. As the cost of providing the additional security features decreases, it is likely that they will be implemented into the system. The authors state that the identification and authentication of the system is not necessary for the F11. The authors declare that having armed guards used for the authentication of the pilot is a quality solution. As devices such as biometric authentication become more common place, this would be a good solution to authenticating and identifying the pilot to the avionic system. If someone were to compromise the authentication and identification that the authors propose, a great deal of damage could be done. It is not out of the question today for someone to try to take control of a plane and an F11 would be a prime target. An extra layer of security would be beneficial.

Today, computers are being integrated into more and more devices. In the future it is possible that the avionics system will be more computer based rather then using an embedded system. Today, common devices, such as refrigerators, are available with computer technology and Internet access. More and more devices are adding hard drives as cost continues to decrease. In the future, it would not be out of the question for the F11 to have a hard drive and use Discretionary Access Control.

Overall, the paper makes a lot of good points about the security needs of embedded real-time systems. It seems like some of the decisions that the authors make are based on cost factors. The security of an F11 is an issue of national defense, and as such the security of an F11 should be scrutinized much more then a computer such as a web server.

Summary

The paper, "Applying the TCSEC Guidelines to a Real-time Embedded System Environment" looks at the ways of modifying the TCSEC for real-time embedded systems. Many of the criteria established by the TCSEC do not apply to a real-time embedded system. The authors suggest ways that the TCSEC can be modified such as removing discretionary access control, authentication and identification. The authors look at some portions of the TCSEC that may need to be changed depending on the circumstances.

Appendix

Modifications to be made to the TCSEC for a real-time embedded system.

Criteria	Appropriate for Real-Time	Comments
	Security Policy	
Discretionary Access Control	**No**	Substitute MAC
Object Reuse	Yes	Memory, processors shared.
Labels, Label Integrity	Yes	Needed for MAC.
Exportation of Labeled Information	Yes	Vendor-supplied components along common bus
Exportation to Multi/Single-level Devices	Yes	"
Labeling Human Readable Output	Yes	Flight data recorders, printouts.
Mandatory Access Control	Yes	Predefined relationships.
Subject Sensitivity Labels	**Probably Not**	*Subjects* are defined as *human users*
Device Labels	Yes	Proprietary and vendor-supplied.
	Accountability	
Identification and Authentication	**None or Limited**	Assumption 2; also, components untrusted, subjects cannot create processes, hence objects will be statically identifiable
Audit	**Limited**	Functionality/performance checks, covert channel detection.
Trusted Path	**None or Limited**	Unnecessary if no human users;
	Assurance	
System Architecture, Integrity	Yes	
Security Testing	Yes	
Design Specification and Verification	Yes	
Covert Channel Analysis	Yes	
Trusted Facility, Config Management	Yes	
Trusted Recovery	Yes	
	Documentation	
Security Feature User's Guide	Yes	
Trusted Facility Manual	Yes	
Test, Design Documentation	Yes	

Fig. 2 Applying TCSEC to a real-time embedded system [Alvess-Foss 1996]

References

Alves-Foss, Jim. Frincke, Deborah. and Saghi, Gene. Applying the IC SEC Guidelines to a Real-Time Embedded System Environment. 19th National Information Systems Security Conference Proceedings. http://csrc.nist.gov/nissc/1996/papers/NISSC96/.

Cheswick, William R., Bellovin, Steven M., and Aviel, Rubin D. Firewalls and Internet Security: Repelling the Wily Hacker Second Edition. Reading, MA.: Addison-Wesley, 2002.

TCSEC Criteria Concepts: Frequently Asked Questions(v4) http://www.radium.ncsc.mil/tpep/process/faq-sect4.html (29 Sep 2003)