Trusted Product Evaluation Questionnaire

National Computer Security Center
9800 Savage Road
Fort George G. Meade, MD 20755-6000

May 2,1992

NCSC-TG-019
Library No. 5-232,458
Version 2

FOREWORD

The National Computer Security Center is publishing the Trusted Product Evaluation Questionnaire as part of the "Rainbow Series" of documents our Technical Guidelines Program produces. In the Rainbow Series, we discuss in detail the features of the Department of Defense Trusted Computer System Evaluation Criteria (DoD 5200.28-STD) and provide guidance for meeting each requirement. The National Computer Security Center, through its Trusted Product Evaluation Program, evaluates the security features of commercially-produced computer systems. Together, these programs ensure that organizations are capable of protecting their important data with trusted computer systems.

The Trusted Product Evaluation Questionnaire is a tool to assist system developers and vendors in gathering data to assist evaluators and potentially certifiers in their assessment of the system.

As the Director, National Computer Security Center, I invite your recommendations for revision to this technical guideline. We plan to review and update this document periodically in response to the needs of the community. Please address any proposals for revision through appropriate channels to:

National Computer Security Center
9800 Savage Road
Ft. George G. Meade, MD 20755-6000

Attention: Chief, Standards, Criteria, and Guidelines Division

Patrick R. Gallagher
. May 1992
Director
National Computer Security Center

ACKNOWLEDGMENTS

The National Computer Security Center expresses appreciation to Dr. Santosh Chokhani and Harriet Goldman, of the MITRE Corporation, as the principal authors of version I of this document; Mr. Kenneth B. Elliott III and Dr. Dixie Baker, of The Aerospace Corporation, as the principal authors of version 2 this document; and ENS Susan L. Mitchell as project manager.

We also thank the evaluators, vendors, and users in the United States computer security community who contributed their time and expertise to the review of this document.

Contents

Chapter 1

INTRODUCTION

One of the principal goals of the National Computer Security Center (NCSC) is to encourage the widespread availability of trusted computer systems. In support of this goal a metric was created, the Department of Defense Trusted Computer System Evaluation Criteria (TCSEC), against which computer systems could be evaluated. The TCSEC was originally published on 15 August 1983 as CSC-STD-001-83. In December 1985 the DoD adopted it, with a few changes, as a DoD Standard, DoD 5200.28-STD. DoD Directive 5200.28, "Security Requirements for Automatic Information Systems (AISs)," has been written to require, among other things, the Department of Defense Trusted Computer System Evaluation Criteria to be used throughout the DoD. The TCSEC is the standard used for evaluating the effectiveness of security controls built into ADP systems. The TCSEC is divided into four divisions: D, C, B, and A, ordered in a hierarchical manner with the highest division (A) being reserved for systems providing the best available level of assurance. Within divisions C, B, and A there are subdivisions known as classes, which are also ordered in a hierarchical manner to represent different levels of security in these classes.

The National Security Agency (NSA) has established an aggressive program to study and implement computer security technology and to encourage the widespread availability of trusted computer products for use by any organization desiring better protection of their important data and information processing services. The Trusted Product Evaluation Program and the open and cooperative business relationship being forged with the computer and telecommunications industries will result in the fulfillment of our country's computer security requirement. We are resolved to meet the challenge of identifying trusted computer products suitable for use in processing all types and classifications of information.

For definition and clarification of tile terms used in this document, please see the glossary section of this questionnaire.

Sub-questions within the numbered questions have been designated with letters (e.g., (a), (b), ...) so that answers to all parts of the main question can be identified.

Review of this document will occur periodically or when the need arises. Address all proposals for revision through appropriate channels to:

National Computer Security Center
9800 Savage Road
Fort George G. Meade, MD 20755-6000

Attention: Chief, Standards, Criteria, and Guidelines Division

1.1 PURPOSE

The NSA is responsible for evaluating commercial products through an independent evaluation based on TCSEC requirements by a qualified team of experts and maintaining a list of those products on the Evaluated Products

List (EPL). To accomplish this mission, the NSA Trusted Product Evaluation Program has been established to assist vendors in developing, testing, and evaluating trusted products for the EPL. During the evaluation process, the TCSEC for classes C1 through Al requires a determination that the security features of a system are implemented as designed and that they are adequate for the specified level of trust. In addition, the TCSEC also requires documentation to support a system's security. During the various phases of the evaluation process, the vendor supplies to an evaluation team certain information on system security and documentation. The purpose of the Trusted Product Evaluation Questionnaire (product questionnaire) is to assist system developers and vendors as a data gathering tool for formalizing the data gathering process for the various phases of the Trusted Products Evaluation process. Additionally, the product questionnaire may be used as a data gathering tool to assist certifiers in the evaluation process for certification and accreditation if the Final Evaluation Report is unavailable. Examples in this document are not to be construed as the only implementations that may answer the question. The examples are suggestions of appropriate implementations. The recommendations in this document are also not to be construed as supplementary requirements to the questionnaire.

1.2 SCOPE

The product questionnaire addresses the TCSEC Criteria Classes C1 through Al. In an effort to gather a better understanding of the system security, some questions in the product questionnaire address information in addition to that required in the Department of Defense Trusted Computer Systems Evaluation Criteria. This document is generally organized by Criteria subject area; within each subject area, the questions are broken down in a manner similar to Appendix D of the Criteria. This breakdown readily allows the reader to discern the questions that are appropriate to each of the Criteria levels. Of course, all of the questions at levels lower than the target level are applicable.

The information provided in the product questionnaire by the vendor is to assist the evaluator in obtaining an initial understanding of the system applying for evaluation and its security features of the respective Criteria class. The product questionnaire is not a statement of requirements, just an information gathering tool. This document should give the vendor an idea of the information required by the evaluator during the evaluation process and prepare the vendor for additional information needed by the evaluation team later on in the evaluation process.

The product questionnaire will be initially sent out to the vendor prior to the Preliminary Technical Review (PTR). The vendor can point to appropriate documents for the answers. The vendor need not answer the questions that are not pertinent. Some of the questions may be applicable at the later stages of the evaluation process and thus may be deferred until the appropriate time (e.g., a finished copy of the Verification Plan). Although the vendor is not obligated to send a completed product questionnaire prior to the PTR, the vendor should have the questionnaire substantially completed by the PTR date so that the PTR team can use the Questionnaire as in input into determining the vendor's ability to support an evaluation. The PTR team will also use the product questionnaire during the PTR to seek additional information to be used later on in the evaluation process. When an evaluation team has reached the Design Analysis Phase and is preparing the Initial Product Assessment Report, it will use the product questionnaire to seek specific references in vendor documentation for further details on the answers to these questions.

The completed document is to provide the evaluator an understanding of the various hardware and software configurations, architecture and design, testing, and documentation, system security features and their applicability to security and accountability policy, Trusted Computing Base (TCB) isolation and non-circumventability, and covert channel analysis methods. This product questionnaire also requests information on penetration testing and specification-to-code correspondence for systems to which these requirements are applicable.

While this product questionnaire is designed for operating systems and dbes not specifically address networks, subsystems, or database management systems, vendors participating in these areas may find it useful to review this document and answer any applicable questions.

Chapter 2

QUESTIONNAIRE

2.1 SUBJECTS

A subject is an active entity in the system, generally in the form of a person, process, or device that causes information to flow among objects or changes the system state. A subject can be viewed as a process/domain pair whose access controls are checked prior to granting the access to objects.

CI:

1. (a) List and (b) describe the subjects in your system.
2. (a) When and (b) how are the subjects created? (For example, they can be created or activated when a user logs on or when a process is spawned.)
3. (a) When and (b) how are the subjects destroyed? (For example, they can be destroyed or deactivated when a process terminates or when the user logs off.)
4. (a) What are the security attributes of a subject? (Examples of security attributes are user name, group id, sensitivity level, etc.) For each type of subject in your system (e.g., user, process, device), what mechanisms are available to (b) define and (c) modify these attributes? (d) Who can invoke these mechanisms?
5. (a) What are other security-relevant privileges a subject can have? (Examples of such privileges are: super user, system operator, system administrator, etc. Your operating system may assign numerous other privileges to the subjects, such as the ability to use certain devices.) For each type of subject in your system, what mechanisms are available to (b) define and (c) modify these pnv)leges? (d) Who can invoke these mechanisms? (e) Provide a list of subjects within the TCB boundary and (f) the list of privileges for each of them.

6. When a subject is created, where do its (a) security attributes and (b) privileges originate; i.e., how are the security attributes and privileges inherited?
7. List the subjects, if any, which are not controlled by the TCB.

2.2 OBJECTS

An object is a passive entity that contains or receives information. Access to an object potentially implies access to the information it contains. Examples of objects are: records, blocks, pages, segments, files, directories, directory tree, and programs, as well as bits, bytes, words, fields, processors, video displays, keyboards, clocks, printers, network nodes.

CI:

1. Provide a list of objects within the TCB (e.g., authentication database, print queues).
2. List the objects in your system that are protected by the Discretionary Access Control (DAC) mechanisms.
3. (a) List the objects that are not protected by the DAC mechanism. (b) Why are they not protected? (c) Describe other mechanisms used to isolate and protect objects.
4. (a) List other resources which are not protected by the DAC mechanism (Examples include temporary data files accessible only to the file's owner). (b) Why are they not protected by DAC? (c) Describe the mechanisms that are used to isolate and protect these resources.
5. How are the various types of objects created (e.g., directories, files, devices)?
6. How are the various types of objects destroyed?
7. (a) What are the security attributes of an object? For each type of object in your system, what mechanisms are available to (b) define and (c) modify these attributes?
   (d) Who can invoke these mechanisms?
8. When an object i$ created, where do its security attributes originate (i.e., how are the
   security attributes inherited?)

BI:

9. List the objects in your system that are protected by the Mandatory Access Control (MAC) mechanisms.
10. (a) List the objects that are not protected by the MAC mechanism. (b) Why are they not protected? (c) Describe other mechanisms used to isolate and protect objects.
11. (a) List other resources which are not protected by the MAC mechanism. (b) Why are they not protected? (c) Describe the mechanisms that are used to isolate and protect these resources.

2.3 HARDWARE ARCHITECTURE

If this evaluation is for a family of hardware, the following questions should be answered for each member of the hardware family. You may choose to answer each question for each member of the family, or answer each question for a baseline family member and point out the difference for each of the remaining family members.

CI:

1. Provide a high-level block diagram of the system. The diagram should at least depict various Central Processor Units (CPUs), memory controllers, memory, I/0 processors, I/0 controllers, I/0 devices (e.g. printers, displays, disks, tapes, communications lines) and relationships (both control flow and data flow) among them.

2. (a) Describe the portions of the system (if any) which contain microcode. (b) How is this microcode protected and loaded?

3. (a) Provide a list of privileged instructions for your hardware. (b) Provide a brief description of each privileged instruction.

4. For each privileged instruction, provide the privileges required to execute the instruction. (Examples of privileges include the machine state, the executing ring/segment/domain/privilege level, physical memory location of the instruction, etc.)

5. How does the process address translation (logical/virtual to physical) work in your system?

6. (a) How does 1/0 processing address translation work for the Direct Memory Access (DMA) controllers/devices? (b) Identify if the address translation is done through the memory address translation unit or if the logic is part of the controller. (c) How are the address translation maps and/or tables initialized?

7. Describe the hardware protection mechanisms provided by the system.

8. Describe what hardware mechanisms are used to isolate the TCB from untrusted applications.

9. (a) What are the machine/processor states supported by the system? (b) How are the states changed? (c) What data structures are saved as part of the processor state?

10. List all the (a) interrupts and (b) traps (hardware and software). (c) How are they serviced by the system?

BI:

11. Provide a high-level block diagram of a CPU. The diagram should explain the relationship among elements such as: Instruction Processor, Microsequencer, Microengine, Memory, Cache, Memory Mapping or Address Translation Unit, I/0 devices and interfaces.

12. Describe the hardware isolation mechanisms for the process memory (e.g., rings, segments, privilege levels).

13. (a) Provide a description of the hardware process address space. (b) When and (c) how is it formed? (d) How does the software use this mechanism, if it does at all?

2.4 SOFTWARE

The TCB software consists of the elements that are involved in enforcing the system security policy. Examples of TCB elements include: kernel, interrupt handlers, process manager, I/0 handlers, I/0 manager, user/process interface, hardware, and command languages/interfaces (for system generation, operator, administrator, users, etc.). The security kernel consists of the hardware, firmware and software elements of the TCB that are involved in implementing the reference monitor concept, i.e., the ones that mediate all access to objects by subjects.

CI:

1. Provide a (a) description and (b) architecture of the Trusted Computing Base (TCB) at the element level (see above for examples of elements).

2. Describe the interface between the TCB and user processes that are outside the TCB.

3. Describe the hardware ring/domain/privilege level/memory segment/physical location where each TCB element resides.

4. Describe the hardware ring/domain/privilege level/memory segment/physical location where the user processes reside.

5. (a) List software mechanisms that are used to isolate and protect the TCB. (b) Provide a brief description of each mechanism.

6. List all the privileges a process can have. Include the privileges based on the process or user profile, process or user name, or process or user identification.

7. How are a process's privileges determined?

8. (a) List the process states and (b) briefly state conditions under which a transition from one state to another occurs.

9. Briefly describe process scheduling.

10. Describe all interprocess communications mechanisms.

11. (a) Describe the file management system. This should include the directory hierarchy, if any, directory and file attributes. (b) Also identify all system directories and files and (c) their access attributes.

12. How are (a) I/0 devices and (b) their queues (if any) managed?

13. How are the (a) batch jobs and (b) their queues managed?

14. What software engineering tools and techniques were used for the TCB design and implementation?

C2:

15. Describe the interfaces (control and data flow) among the TCB elements.

16. Describe the interface between the kernel and the rest of the TCB elements.

17. Describe how the process states are manipulated by the TCB.

18. (a) Describe the data structures for a process context. Describe both (b) hardware and (c) software mechanisms used to manipulate/switch the process context.

BI:

19. (a) List software mechanisms that are used to isolate and protect user processes. (b) Provide a brief description of each mechanism.

20. (a) Describe various elements of the process address space and (b) their location in terms of ring/domain/privilege level/segment/physical memory.

21. How is a process' sensitivity level determined?

B2:

22. How was the modularity requirement achieved and implemented?

23. (a) For each TCB element, identify protection-critical portions of the code. (b) Describe the protection-critical functions performed by the code.

24. (a) Is the TCB layered? (b) If yes, how many layers are in the TCB? Provide a brief description of (c) modules and (d) functions in each layer. (e) How are the lower layers protected from higher layers?

B3:

25. How does the architecture limit or restrict the ability of untrusted code to exploit covert channels?

26. How is the least privilege requirement achieved and implemented?

27. (a) For each TCB element, identify non-protection-critical portions of the code. (b) Explain why the code is part of the TCB.

28. How was the data abstraction and information hiding requirement achieved and implemented?

2.5 DISCRETIONARY ACCESS CONTROL

CI:

1. What mechanisms are used to provide discretionary access controls? (Examples of mechanisms are: access control lists, protection bits, capabilities, etc.)

2. (a) Can the access be granted to the users on an individual user basis? (b) If so, how?

3. (a) How is a group defined? (b) What mechanisms are used to administer groups (i.e., to create or delete groups or to add or delete individual users from a group)? (c) Who can invoke these mechanisms? (d) What privileges are necessary to invoke these mechanisms?

4. How can the access be revoked on an individual user basis?

5. How can the access be revoked on a group basis?

6. List any objects that can be accessed by users excluded from the DAC policy (e.g., IPC files, process signaling/synchronization flags)?[1]

7. For each TCB object identified in question 1, section 2.2, describe the DAC mechanism which protects that object.

8. (a) List the access modes supported by the system (e.g., read, write, delete, owner, execute, append). (b) Briefly describe the meaning of each access mode for each object.

9. (a) Are conflicts between user and group access detected? (b) If so, how are the conflicts resolved?

10. For each object, list when changes in DAC permissions become effective.

C2:

11. (a) Can access be granted to groups of individuals? (b) If so, how?

12. (a) What are the initial access permissions when an object is created? (b) Can the initial access permission be changed? If so, (c) by whom (e.g., user/owner, system administrator, others) and (d) how.

13. (a) Can different initial access permissions be specified for different users, or is this is a system-wide setting? If the former, (b) by whom and (c) how?

1 This question is not applicable above class BI, because then all objects have to be protected.

14. (a) Who can grant the access permissions to an object after the.object is created? (Examples include creator, current owner, system administrator, etc.) (b) How is the permission granted?

15. (a) Can the ability to grant permissions be passed to another user? If so, (b) by whom and (c) how? (d) Under what circumstances can the previous owner of the privilege retain it?

B3:

16. (a) Can access be denied to the users on an individual user basis, i.e., exclude individual users? (b) If so, how?

17. (a) Can access be denied to groups of individuals? (b) If so, how?

2.6 IDENTIFICATION & AUTHENTICATION

CI:

1. (a) Does the system require the users to provide identification at login? (b) If yes, what information is requested by the system?

2. Is there any additional device or physical security required for user identification and authentication (I&A) (e.g., terminal ID, pass key, smart card, etc.)?

3. (a) Does the system authenticate this identity at the time of login? (b) If yes, what information is requested by the system? (c) How does the system use this information to authenticate the identity?

4. (a) Describe the algorithms used in user authentication. (b) Where in the system are the code and data for authentication (e.g., user/ password data base) stored?

5. How are the authentication code and data protected?

6. (a) Does the I&A process associate privileges with the user? If so, (b) what and (c) how?

C2:

7. Describe how each user is uniquely identified.

BI:

8. How does the I&A process associate a sensitivity level with the user?

2.7 OBJECT REUSE

C2:

1. How is reuse of data in the storage resources (e.g., memory page cache, CPU registers, disk sectors, magnetic tapes, removable disk media, terminals) of the system prevented? (Examples include writing predefined patterns, writing random patterns, preventing reading before writing, etc.)

2. When do these actions take place: prior to allocation or after deallocation and/or release?

3. Describe the TCB (a) hardware, (b) software and (c) procedural mechanisms used to accomplish the clearing for each type of storage resource.

4. Is it possible to read data that have been "logically" deleted, but not physically removed (e.g., attempting to read past the end-of-file mark)?

2.8 AUDIT

C2:

1. Provide a brief description (preferably in block diagram form) of audit data flow in terms of how the data are created, transmitted, stored, and viewed for analysis.

2. How are the audit logs protected?

3. (a) How can the audit log be read? (b) Who can invoke these mechanisms? (c) What privileges are required to invoke these mechanisms?

4. (a) What tools are available to output raw or processed (i.e., analyzed and reduced) audit information? (b) Who can invoke these tools? (c) What do the tools do in terms of audit data reduction? (d) What are the formats of the reports/outputs generated by these tools?

5. (a) How can the audit log be written or appended? (b) Who can invoke these mechanisms? (c) What privileges are required to invoke these mechanisms?

6. (a) How can the audit log be deleted? (b) Who can invoke these mechanisms? (c) What privileges are required to invoke these mechanisms?

7. What are the internal formats of audit records?

8. Provide a list of auditable events (examples include attempted logins, logouts, creation of subjects, deletion of subjects, assignment of privileges to subjects, change of subject privileges, use of privileges by subjects, creation of objects, deletion of objects, initial access to objects (introduction of the object into a user's address space), assumption of the role of security administrator).

9. (a) Which actions by the trusted users are auditable? (b) Which are not? (Examples of trusted users are system operator, account administrator, system security officer/administrator, auditor, system programmer, etc. Trusted users almost always have at least one privilege.)

10. (a) What data are recorded for each audit event? (b) Which of the following data (if any) are not recorded for each event: date, time, user, object, object DAC information (e.g., ACL), type of event, invoked or not invoked, why not invoked, success or failure in execution,terminal identification?

11. (a) Can the password ever become part of the audit record? (b) If yes, under what circumstances can this occur?

12. (a) What mechanisms are available to designate and change the activities being audited? (b) Who can invoke these mechanisms? (c) What privileges are needed to invoke these mechanisms?

13. (a) What mechanisms are available for selective auditing (i.e., selection of events, subjects, objects, etc., to be audited)? (b) What parameters (e.g., individual or group of subjects, individual objects, subjects within a sensitivity range, objects within a sensitivity range, event type) or combination of parameters can be specified for the selective auditing? (c) Who can invoke these mechanisms? (d) What privileges are needed to invoke these mechanisms?

14. When do changes to the audit parameters take effect (e.g., immediately for all processes, for new processes)?

15. (a) Are the audit reduction tools part of the TCB? (b) If not, what trusted mechanism is used to view/output the audit log?

16. (a) Does the system produce multiple audit logs? (b) If yes, what tools, techniques and methodologies are available to correlate these logs?

17. (a) Who (e.g., operator, system administrator or other trusted user) is notified when the audit log gets full? (b) What options are available to handle the situation?

18. What other action does the TCB take when the audit log becomes full (e.g., halt the system, do not perform auditable events, overwrite oldest audit log data).

19. (a) In the worst case, how much audit data can be lost (e.g., when audit log overflows, system goes down with audit data in memory buffers)? (b) Describe the worst case scenario. (c) When can it occur?

BI:

20. Which of the following events auditable: change in the device designation of single-level or multilevel, change in device level, change in device minimum or maximum level, override of banner page or page top and bottom markings?

21. Are the (a) subject and (b) object sensitivity level recorded as part of the audit event?

B2:

22. Are events that exploit covert storage channels auditable?

B3:

23. How does the TCB (a) designate and (b) change the occurrence or accumulation of events that require real-time notification? (c) Who can invoke these mechanisms? (d) What privileges are needed to invoke these mechanisms? (e) Who (e.g., system administrator, president of the company) gets the real-time notification? (f) What actions/options are available to the individual being notified? What does the TCB do about (g) the event and (h) the process that caused this alert?

2.9 LABELS

BI:

1. (a) How many hierarchical sensitivity classifications (such as unclassified, confidential, secret, top secret), does your system provide for? (b) What mechanisms are available to define the internal/ storage and external/print format? (c) What mechanisms are available to change them? (d) Who can invoke these mechanisms?

2. (a) How many non-hierarchical sensitivity categories (such as FOUO) does your system provide for? (b) What mechanisms are available to define the internal/storage and external/print format? (c) What mechanisms are available to change them? (d) Who can invoke these mechanisms?

3. (a) What is the internal TCB storage format of the sensitivity label? (b) If different for different subjects or objects, give all formats.

4. For each type of subject, where is the subject sensitivity label stored?

5. For each type of object, where is the object sensitivity label stored?

6. (a) List any subjects and objects that are not labeled. (b) Why are they not labeled? How are these subjects and objects (c) accessed and (d) controlled?

7. (a) How is imported data labeled? (b) How is this label determined? Is a human being involved in (c) the determination or (d) the actual labeling? (e) If so, what is the role of the person involved (e.g., system administrator, system operator)? (f) Does the labeling require special privileges? (g) If so, what are those privileges?

8. (a) Who can change the labels on a subject? (b) How?

9. (a) Who can change the labels on an object? (b) How?

10. How are the labels associated with objects communicated outside the TCB?

11. (a) How does the system designate each device to be single-level or

multilevel? (b) List the ways this designation can be changed. (c) List the users who can perform this designation.

12. (a) How does the TCB designate the sensitivity level of a single-level device? (b) List the ways this designation can be changed. (c) List the users who can do this.

13. (a) How does the TCB export the sensitivity label associated with an object being exported over a multilevel device? (b) What is theJormat for the exported label? (c) How does the TCB ensure that the sensitivity label is properly associated with the object?

14. (a) What mechanisms are available to specify the human-readable print label associated with a sensitivity label? (b) Who can invoke these mechanisms?

15. (a) Is the beginning and end of each hardcopy output marked with the human-readable print label representing the sensitivity level of the output (i.e., does each hardcopy output have banner pages)? (b) What happens if a banner page output is longer and/or wider than a physical page?

16. (a) Is the top and bottom of each hardcopy output page marked with the human-readable print label representing the sensitivity level of the output? (b) What happens if the print label is wider and/or longer than the space available for the top and/or the bottom?

17. How does the TCB mark the top and bottom page of non-textual type of output such as graphics, maps, and images?

18. (a) How can the top and bottom page markings be overridden? (b) Who can override the markings?

19. How can an operator distinguish the TCB-generated banner pages from user output?

B2:

20. (a) How does the TCB acknowledge a change in the sensitivity level associated with an interactive user? (b) Is the user notification posted on the user terminal? (c) How immediate is this change?

21. (a) How does a user query the system TCB for his or her current sensitivity label? (b) What part of the sensitivity label is output? (c) Where is this output posted?

22. (a) How does the TCB designate the minimum and maximum sensitivity levels of a device? (b) List the ways these designations can be changed. (c) List the users who can invoke these mechanisms.

23. List the circumstances under which the TCB allows input or outj~ut of data that fall outside a device's sensitivity range.

2.10 MANDATORY ACCESS CdNTROL

BI:

1. Define the MAC policy for the possible access modes such as read, write, append, delete.

2. (a) Does the system use sensitivity labels to enforce the MAC? (b) If not, what information is used to make the MAC decisions?

3. (a) List the subjects, objects, and circumstances under which the MAC policy is not enforced.[2] (b) Why is it not enforced in these cases?

4. In what sequence does the system perform access mediation? (An example sequence might be a. check for privileges that supersede MAC and DAC, then b. check for DAC, then c. check for MAC.)

5. (a) Does the TCB support system-low and system-high sensitivity levels? If yes, how can they be (b) designated and (c) changed? Who can invoke the functions to (d) designate and (e) change them? How are these levels used by the system in (f) various labeling functions and (g) MAC decisions?

2This question is not applicable above class BI, because then all objects have to be protected.

2.11 TESTING

CI:

1. (a) What routines are available to test the correct operation of the system hardware and firmware? (b) What elements of the system hardware are tested through these routines? (c) What elements of the system firmware are tested through these routines? (d) What elements of the system hardware and firmware are not tested through these routines? (e) Does the testing include boundary and anomalous conditions? (f) Is the emphasis on diagnosing and pinpointing faults or is it on ensuring the correct operation of the system hardware and firmware?

2. (a) How are the routines in the previous question invoked? (b) Who can invoke these routines? (c) Do they run under the control of the operating system or do they run in stand-alone mode?

3. (a) When can these routines be run? (b) When should these routines be run? (c) If they run automatically, when do they run (e.g., powerup, booting, rebooting)?

4. Describe the software development testing methodology. In this description, include a discussion of various testing steps such as unit, module, integration, subsystem, system testing. This discussion should include a description of test coverage criteria and test cases development methodology.

5. Provide (a) a copy of the security test plan, a brief description of its contents, or an annotated outline. (b) Does the test plan include the following information: system configuration for testing, procedures to generate the TCB, procedures to bring up the system, testing schedule, test procedures, test cases, expected test results? (c) Provide a schedule for development of the security test plan if such a test plan doesn't already exist.

6. (a) How thorough is the security testing? (b) Do the test cases include nominal, boundary, and anomalous values for each input? (c) What about the combinations of inputs? (d) Describe the test coverage criteria.

7. (a) How are the test cases developed? (b) Are they based on the concept of functional testing, structural testing, or a combination of the two?

8. What tools and techniques (automated, manual, or a combination of the two) will be used to do the functional and/or structural analysis in order to develop a thorough set of test cases?

BI:

9. How do you plan to ascertain that errors have been minimized in the system?

B2:

10. What is the role of the descriptive top-level specification (DTLS) in the functional and/or structural analysis done in order to develop a thorough set of test cases?

11. (a) Do you plan to develop scenarios for penetration testing? (b) If so, what methodologies will be used?

12. How do you plan to compute and verify the bandwidths of covert channels?

Al:

13. What is the role of the formal top-level specification (FTLS) in the functional and/or structural analysis done in order to develop a thorough set of test cases?

2.12 MODELING AND ANALYSIS

BI:

1. Describe the system security policy.

2. How is the system security policy represented in the informal model?

3. What policies are represented in the informal model (e.g., MAC, DAC, privileges, other protection mechanisms, object reuse)?

4. What tools, techniques and methodologies are used to demonstrate the model consistent with its axioms?

B2:

5. (a) Provide a copy of the Verification Plan, a brief description of its contents, or an annotated outline. (b) Provide a schedule for completion of the Verification Plan.

6. What tools, techniques and methodologies are used to represent the formal model of the system security policy?

7. What policies are represented in the formal model (e.g., MAC, DAC, privileges, other protection mechanisms, object reuse)?

8. What tools, techniques and methodologies are used to prove the model consistent with its axioms?

9. (a) What tools, techniques and methodologies are used to represent the descriptive top-level specificatidn (DTLS)? (b) What portions of the TCB are represented by the DTLS?

10. What tools, techniques and methodologies are used to identify, analyze, calculate, and reduce the bandwidths of covert channels?

B3:

11. What tools, techniques and methodologies are used to show that the DTLS is consistent with the formal security policy model?

12. (a) What tools, techniques and methodologies are used to represent the formal top-level specification (FTLS)? (b) What portions of the TCB are represented by the FTLS?

13. What tools, techniques and methodologies are used to verify or show that the FTLS is consistent with the formal security policy model?

14. What tools, techniques and methodologies are used to identify the implemented code modules that correspond to the FTLS?

15. What tools, techniques and methodologies are used to show that the code is correctly implemented vis-a-vis the FTLS?

2.13 OTHER ASSURANCES

Although the configuration management criteria do not appear until class B2 in the TCSEC, the questions pertaining to configuration management below are relevant to all classes because of the NSA's Ratings Maintenance Phase (RAMP) program.

CI:

1. (a) Describe the Configuration Management (CM) system in place in terms of organizational responsibilities, procedures, and tools and techniques (automated, manual, or a combination of the two). (b) Describe the version control or other philosophy to ensure that the elements represent a consistent system, i.e., object code represents the source code, and the design documentation accurately describes the source code. (c) If the CM system is different for some of the elements listed in question 1 in section 2.4, answer this question for each of the elements.

2. (a) When was this system placed under configuration management? (b) Provide the approximate date, as well as the life-cycle phase (e.g., design, development, testing). Answer this question for each system element so controlled (as listed in the previous question).

3. List the elements that are and are not under the Configuration Management (e.g., hardware, firmware, formal security policy model, FTLS, DTLS, design data and documentation, source code, object code, test plans, Security Features User's Guide, Trusted Facilities Manual).

4. Describe the protection mechanisms in place to safeguard the CM elements.

5. (a) List separately the functions that can be performed by each of the trusted users (e.g., operator, security administrator, accounts administrator, auditor, systems programmer). (b) For each of these

persons/roles, list the system data bases that can be accessed and their access modes. (c) Also list the privileges provided to each of these roles.

6. (a) How does the TCB recognize that a user has assumed one of the above-mentioned trusted roles? (b) Which of the above-mentioned functions can be performed without the TCB recognizing this role?

7. (a) Does the system have a degraded mode of operation? (b) What can cause this to occur? (c) How long can the system keep running in this mode? (d) How does an operator get the system back to full operation? (e) What security-related services are provided in the degraded mode? (f) What security-related services are not provided?

B2:

8. Describe the version control or other philosophy to ensure that the object code corresponds to the correct source code, which in turn is accurately abstracted in the DTLS.

9. (a) When (e.g., before user authentication) and (b) how (e.g., by typing a specific control character sequence) can the trusted path be invoked by the user? (c) What TCB elements are involved in establishing the trusted path?

10. How does the TCB ensure that the trusted path is unspoofable?

11. How do you plan to show consistency between the DTLS and the code?

B3:

12. What security relevant actions are able to be performed under trusted path?

13. Are there other system interfaces which support the same functionality as provided in the trusted path?

14. (a) How does the system recovery work? What system resources (e.g., memory, disks blocks, files) are protected (b) prior to and (c) during the system recovery? (d) How are they protected? (e) What resources are not protected?

A1:

15. Describe the version control or other philosophy which ensures that the FTLS continues to accurately describe the system through system changes.

16. How do you plan to show consistency among the FTLS, DTLS and the code?

17. Describe the tools, techniques and procedures used to ensure the integrity of the TCB elements (hardware, firmware, software, documents, etc.) supplied to the customers (e.g., trusted courier, electronic seals, physical seals).

2.14 OTHER DOCUMENTATION

CI:

1. (a) Describe the methodology used in the design of the system. (b) Provide a list of documents that capture the system design. (c) For each document, provide a copy, a brief description of its contents, or an annotated outline. (d) Provide a schedule for development of the design documents.

2. Does the SFUG describe (a) the protection mechanisms provided by the TCB, (b) guidelines on their use, and (c) how they interact?

3. Does the SFUG explain to users the underlying philosophy of protection for the system?

4. Does the SFUG discuss the need for exercising sound security practices in protecting the information processed and/or stored in the system, including all input and output?

5. Does the SFUG describe users' responsibilities with respect to assuring the effectiveness of the protective features (e.g., password selection and protection)?

6. Does the SFUG describe security-related commands available to users?

7. Does the SFUG explain how to use the DAC mechanism(s) provided by the system to protect objects?

8. Does the SFUG explain how removable media are to be handled (if applicable)?

9. Does the SFUG discuss the auditing of security-relevant events?

10. Does the SFUG include and clearly highlight warnings where needed?

11. (a) Does the TFM ~~ontain procedures to configure the secure system? (b) Does it list the devices and hardware elements that are part of the evaluated configuration? Does it contain procedures (c) for configuring each of the devices, (d) for connecting them, and (e) for configuring the entire system? (f) Does it list the devices that are not part of the evaluated configuration? (g) Does it list the procedures for securely configuring them out and for disconnecting them?

12. Does the TFM list the (a) functions, (b) privileges, and (c) data bases that are to be controlled? (d) Does it describe how these are controlled? (e) Does it describe the consequences of granting access to them as warnings?

13. (a) Does the TFM contain the procedures and warnings relating to the secure operation of the computing facility? (b) Does it address the physical, personnel, and administrative aspects of security in order to

ensure the protection of computing hardware, firmware, software, and privileged devices such as the operator terminals?

14. Does the TFM contain the procedures for securely starting/booting/ initializing the system?

C2:

15. (a) Does the TFM provide procedures for maintaining the audit log? (b) Does it describe how ihe audit log can be turned on, turned off, combined with other audit logs, and backed up? (c) Does it describe how to detect that the audit log is getting full, or is full, and what actions to take in order to minirnize the loss of audit data?

16. Does the TFM contain the (a) structure of the audit log file and the (b) format of the audit records? (c) Does it describe how the audit records can be viewed? Does it (d) describe the capabilities of the audit reduction tool, (e) how to invoke these capabilities, and (f) the format of the tool output?

B1:

17. Does the TFM address the protection of hard-copy outputs?

18. (a) Does the TFM provide a list of trusted users (e.g., system operator, security administrator, accounts administrator, auditor) and trusted processes (device queue manipulation, user profile editor)? (b) For each trusted user or process, does it list the functions (e.g., creating and deleting users, changing user security profile, setting up defaults for discretionary and mandatory access controls, selecting auditing events), privileges, and data bases (e.g., user security profiles, authentication data base) to be accessed?

B2:

19. (a) Does the TFM contain procedures to generate the TCB from source code? (b) For each system parameter or input, does the TFM list valid values for a secure TCB generation?

20. Does the TFM include a list of TCB modules that make up the security kernel?

21. Are the separate operator and administrator functions clearly identified and described?

B3:

22. Does the TFM contain the procedures for secureJy restarting/resuming the system after a lapse in system operation, or a system failure?

Chapter 3

GLOSSARY

Access

A specific type of interaction between a subject and an object that results in the flow of information from one to the other.

Access List

A list of users, programs, and/or processes and the specifications of access categories to which each is assigned.

Administrative User

A user assigned to supervise all or a portion of an ADP system.

Audit

To conduct the independent review and examination of system records and activities.

Audit Trail

A chronological record of system activities that is sufficient to enable the reconstruction, reviewing, and examination of the sequence of environments and activities surrounding or leading to an operation, a procedure, or an event in a transaction from its inception to final results.

Auditor

An authorized individual, or role, with administrative duties, which include selecting the events to be audited on the system, setting up the audit flags that enable the recording of those events, and analyzing the trail of audit events.

Authenticate

(1) To verify the identity of a user, device, or other entity in a computer system, often as a prerequisite to allowing access to resources in a system. (2) To verify the integrity of data that have been stored, transmitted, or otherwise exposed to possible unauthorized modification.

Authenticated User

A user who has accessed an ADP system with a valid identifier and authentication combination.

Authorization

The granting of access rights to a user, program, or process.

Bandwidth

A characteristic of a communication channel that is the amount of information that can be passed through it in a given amount of time, usually expressed in bits per second.

Bell-LaPadula Model

A formal state transition model of computer security policy that describes a set of access control rules. In this formal model, the entities in a computer system are divided into abstract sets of subjects and objects. The notion of a secure state is defined, and it is proven that each state transition preserves security by moving from secure state to secure state, thereby inductively proving that the system is secure. A system state is defined to be "secure" if the only permitted access modes of subjects to objects are in accordance with a specific security policy. In order to determine whether or not a specific access mode is allowed, the clearance of a subject is compared to the classification of the object, and a determination is made as to whether the subject is authorized for the specific access mode. The clearance/classification scheme is expressed in terms of a lattice. See Star Property (*-property) and Simple Security Property.

Channel

An information transfer path within a system. May also refer to the mechanism by which the path is effected.

Covert Channel

A communication channel that allows an untrusted subject with legitimate access to information to transfer that information in a manner that violates the system's security policy, using a mechanism in some way not intended by the system developers.

Covert Storage Channel

A covert channel that involves the direct or indirect writing of a storage location by one process and the direct or indirect reading of the storage location by another process. Covert storage channels typically involve a finite resource (e.g., sectors on a disk) that is shared by two subjects at different security levels.

Covert Timing Channel

A covert channel in which one process signals information to another by modulating its own use of system resources (e.g., CPU time) in such a way that this manipulation affects the real response time observed by the second process.

Coverage Analysis

Qualitative or quantitative assessment of the extent to which the test conditions and data show compliance with required properties (e.g., security model and TCB primitive properties). See: Test Condition, Test Data, Security Policy Model.

Data

Information with a specific physical representation.

Data Integrity

The property that data meet an a priori expectation of quality.

Degauss

To reduce magnetic flux density to zero by applying a reverse magnetizing field.

Descriptive Top-Level Specification (DTLS)

A top-level specification that is written in a natural language (e.g.,English), an informal program design notation, or a combination of the two.

Discretionary Access Control (DAC)

A means of restricting access to objects based on the identity and need-to-know of the user, process and/or groups to which they belong. The controls are discretionary in the sense that a subject with a certain access permission is capable of passing that permission (perhaps indirectly) on to any other subject.

Dominate

Security level S1 is said to dominate security level S2 if the hierarchical classification of S1 is greater than or equal to that of S2 and the non-hierarchical categories of S1 include all those of 52 as a subset. Exploitable Channel Any channel that is usable or detectable by subjects external to the Trusted Computing Base whose purpose is to violate the security policy of the system.

Flaw

An error of commission, omission, or oversight in a system that allows protection mechanisms to be bypassed.

Flaw Hypothesis Methodology

A system analysis and penetration technique in which specifications and documentation for the system are analyzed and then flaws in the system are hypothesized. The list of hypothesized flaws is prioritized on the basis of the estimated probability that a flaw actually exists and, assuming a flaw does exist, on the ease of exploiting it and on the extent of control or compromise it would provide. The prioritized list is used to direct a penetration attack against the system.

Formal Proof

A complete and convincing mathematical argument, presenting the full logical justification for each proof step, for the truth of a theorem or set of theorems.

Formal Security Policy Model

A mathematically precise statement of a security policy. To be adequately precise, such a model must represent the initial state of a system, the way in which the system progresses from one state to another, and a definition of a "secure" state of the system. To be acceptable as a basis for a TCB, the model must be supported by a formal proof that if the initial state of the system satisfies the definition of a "secure" state and if all assumptions required by the model hold, then all future states of the system will be secure. Some formal modeling techniques include: state transition models, temporal logic models, denotational semantics models, algebraic specification models.

Formal Top-Level Specification (FTLS)

A top-level specification that is written in a formal mathematical language to allow theorems showing the correspondence of the system specification to its formal requirements ~o be hypothesized and formally proven.

Formal Verification

The process of using formal proofs to demonstrate the consistency between a formal specification of a system and a formal security policy model (design verification) or between the formal specification and its program implementation (implementation verification).

Functional Testing

The segment of security testing in which the advertised mechanisms of a system are tested, under operational conditions, for correct operation.

Identification

The process that enables recognition of an entity by a system, generally by the use of unique machine-readable user names.

Integrity

Sound, unimpaired or perfect condition.

Internal Security Controls

Hardware, firmware, and software features within a system that restrict access to resources (hardware, software, and data) to authorized subjects only (persons, programs, or devices).

Isolation

The containment of subjects and objects in a system in such a way that they are separated from one another, as well as from the protection controls of the operating system.

Lattice

A non-empty set X with a reflexive partial order such that for every pair x,y of members X, there is a unique smallest element greater than each x and y and a unique largest element that is smaller than each x and y.

Least Privilege

This principle requires that each subject in a system be granted the most restrictive set of privileges (or lowest clearance) needed for the performance of authorized tasks. The application of this principle limits the damage that can result from accident, error, or unauthorized use.

Mandatory Access Control (MAC)

A means of restricting access to objects based on the sensitivity (as represented by a label) of the information contained in the objects and the formal authorization (i.e., clearance) of subjects to access information of such sensitivity.

Multilevel Device

A device that is used in a manner that permits it to simultaneously process data of two or more security levels without risk of compromise. To accomplish this, sensitivity labels are normally stored on the same physical medium and in the same form (i.e., machine-readable or human-readable) as the data being processed.

Object

A passive entity that contains or receives information. Access to an object potentially implies access to the information it contains. Examples of objects are: records, blocks, pages, segments, files, directories, directory tree, and programs, as well as bits, bytes, words, fields, processors, video displays, keyboards, clocks, printers, network nodes.

Object Reuse

The reassignment and reuse of a storage medium (e.g., cage frame, disk sector, magnetic tape) that once contained one or more objects. To be securely reused and assigned to a new subject, storage media must contain no residual data (magnetic remanence) from the object(s) previously contained in the media.

Partial Ordering

A partial order on a set X is a relation R having the property that if (x,y) is in R and (y,z) is in R, then (x,z) is in R. A partial order is reflexive if (x,x) is in R for each x in X.

Penetration

The successful act of bypassing the security mechanisms of a system.

Process

A program in execution.

Protection-Critical Portions of the TCB

Those portions of the TCB whose normal function is to deal with the control of access between subjects and objects. Their correct operation is essential to the protection of data on the system.

Read

A fundamental operation that results only in the flow of information from an object to a subject.

Read Access (Privilege)

Permission to read information.

Reference Monitor Concept

An access-control concept that refers to an abstract machine that mediates all accesses to objects by subjects.

Security Level

The combination of a hierarchical classification and a set of non-hierarchical categories that represents the sensitivity of information.

Security Policy

The set of laws, rules, and practices that regulate how an organization manages, protects, and distributes sensitive information.

Security Policy Model

A formal presentation of the security policy enforced by the system. It must identify the set of rules and practices that regulate how a system manages, protects, and distributes sensitive information. See Bell-LaPadula Model and Formal Security Policy Model.

Security-Relevant Event

Any event that attempts to change tide security state of the system, (e.g., change discretionary access controls, change the security level of the subject, change user password). Also, any event that attempts to violate the security policy of the system, (e.g., too many attempts to log in, attempts to violate the mandatory access control limits of a device, attempts to downgrade a file).

Security Testing

A process used to determine that the security features of a system are implemented as designed. This includes hands-on functional testing, penetration testing, and verification.

Simple Security Property

A Bell-LaPadula security model rule allowing a subject read access to an object only if the security level of the subject dominates the security level of the object. Also called simple security condition.

Single-Level Device

An automated information systems device that is used to process data of a single security level at any one time.

Spoofing

An attempt to gain access to a system by posing as an authorized user. Synonymous with impersonating, masquerading or mimicking.

Star Property

A Bell-LaPadula security model rule allowing a subject write access to an object only if the security level of the object dominates the security level of the subject. Also called confinement property, *-property.

Subject

An active entity, generally in the form of a person, process, or device, that causes information to flow among objects or changes the system state. Technically, a process/domain pair.

Subject Security Level

A subject's security level is equal to the security level of the objects to which it has both read and write access. A subject's security level must always be dominated by the clearance of the user the subject is associated with.

Terminal Identification

The means used to provide unique identification of a terminal to a system.

Test Condition

A statement defining a constraint that must be satisfied by the program under test.

Test Data

The set of specific objects and variables that must be used to demonstrate that a program produces a set of given outcomes.

Test Plan

A document or a section of a document which describes the test conditions, data, and coverage of a particular test of group of tests. See also: Test Condition, Test Data, Coverage Analysis.

Test Procedure (Script)

A set of steps necessary to carry; out one or a group of tests. These include steps for test environment initialization, test execution, and result analysis. The test procedures are carried out by test operators.

Test Program

A program which implements the test conditions when initialized with the test data and which collects the results produced by the program being tested.

Top-Level Specification

A nonprocedural description of system behavior at the most abstract level, typically, a functional specification that omits all implementation details.

Trusted Computer System

A system that employs sufficient hardware and software integrity measures to allow its use for processing simultaneously a range of sensitive or classified information.

Trusted Computing Base (TCB)

The totality of protection mechanisms within a computer system-including hardware, firmware, and software-the combination of which is responsible for enforcing a security policy. It creates a basic protection environment and provides additional user services required for a trusted computer system. The ability of a trusted computing base to correctly enforce a security policy depends solely on the mechanisms within the TCB and on the correct input by system administrative personnel of parameters (e.g., a user's clearance) related to the security policy.

Trusted Path

A mechanism by which a person at a terminal can communicate directly with the Trusted Computing Base. This mechanism can only be activated by the person or the Trusted Computing Base and cannot be imitated by untrusted software. Person or process accessing an AIS either by direct connections (i.e., via terminals), or indirect connections (i.e., prepare input data or receive output that is not reviewed for content or classification by a responsible individual).

Verification

The process of comparing two levels of system specification for proper correspondence (e.g., security policy model with top-level specification, top-level specification with source code, or source code with object code). This process may or may not be automated.

Verification Plan

A deliverable as specified in the Trusted Product Evaluation Management Plan. It indicates how the system design will be verified. It should include identification of the specification language/system to be used, an indication of any special features of the language that will be used, and the planned number of levels that specifications will be written for. The method to be used for theorem proving, either manual, interactive 6r automated, should be indicated. The plan will be submitted to the team for review.

Write

A fundamental operation that results only in the flow of information from a subject to an object.

Write Access (Privilege)

Permission to write an object.

Chapter 4

REFERENCES

1. Department of Defense, Trusted Computer System Evaluation Criteria, DoD 5200.28-STD, December 1985.
2. Department of Defense, Security Requirements for Automated Information Systems (AISs), DoD Directive 5200.28, 21 March 1988.
3. Aerospace Report No. TOR-0086 (6777-25)1, Trusted Computer System Evaluation Management Plan, 1 October 1985.
4. National Computer Security Center, NCSC-TG-002 Version-I, Trusted Product Evaluations - A Guide For Vendors, 1 March 1988(DRAFT).
5. National Computer Security Center, NCSC-TG-004 Version l, Glossary of Computer Security Terms, 21 October 1988.
6. National Computer Security Center, NCSC-TG-013 Version I, Rating Maintenance Phase - Program Document, 23 June 1989.