Global Information Assurance Certification Paper

The Clark-Wilson Security Model
Sonya Q. Blake
May 17, 2000
Version Number 1.2e

Research Paper Contents

Introduction
Security Terminology
Principles of Integrity
A Security Model for Integrity
Overview of Windows NT Security Model
Windows NT Interpretation of the Clark-Wilson Model
Summary

Introduction

Information systems security concerns itself with three essential properties of information: *confidentiality, integrity, and availability*. These three critical characteristics of information are major concerns throughout the commercial and military industry. Historically, confidentiality has received the most attention, probably because of its importance in military. The military environment main objective is to prevent disclosure of information. Unlike the military security systems, the main concern of commercial security systems is to ensure that the integrity of data is protected from improper modifications and inappropriate actions performed by unauthorized users. Confidentiality is equally important within the commercial environment, however, David D. Clark and David R. Wilson argue that the integrity of the information is more important than its confidentiality in most commercial systems. Since much of the attention in the security arena has been devoted to developing sophisticated models (e.g. Bell-LaPadula model) and mechanisms for confidentiality, capabilities to provide confidentiality in information systems are considerably more advanced than those providing integrity. Accordingly, recent efforts by National Institute of Science and Technology (NIST) are focused on the integrity issue. In this paper, we will explore the nature and scope of the Clark-Wilson (CW) model.

Security Terminology

As a background to the discussion of the CW model, a brief description of specific security terms used within the security world is in order. The terminology presented below came from the review of written material from various sources and is merely summarized here.

Integrity

There has been a large amount of debate over the meaning of integrity in the information security community. For the purposes of this paper, data integrity is defined as the, quality, correctness, authenticity, and accuracy of information stored within an information system. (Summers, p.141). Systems integrity is the successful and correct operation of information resources. Together, these definitions define *integrity* as information is not modified in unauthorized ways, that it is internally consistent and consistent with the real-world objects that it represents, and that the system performs correctly. (Summers, p.152).

Security Policy
The goal of information systems is to control or manage the access of subjects (users, processes) to objects (data, programs). This control is governed by a set of rules and objectives called a *security policy*. *Security policies* are governing principles adopted by organizations [NCSC 1988]. They capture the security requirements of an organization, specify what security properties the system must provide and they describe steps an organization must take to achieve security.

In the ANSA enterprise projection, a policy consists of an objective, missions and constraints. Each part of the policy has a security aspect [16].

The objective defines what future state is desired for security.

The security aspect of the missions will define the activities associated with security such as granting or revoking access rights.

The security aspect of the constraints defines the rules that must be followed to preserve security when carrying out the activities defined by the missions.

Security Models
Security models are often regarded as a formal presentation of the security policy enforced by the system [NCSC 1988] and are used to test a policy for completeness and consistency [17]. They describe what mechanisms are necessary to implement a security policy.

Security Mechanisms
A security mechanism enforces or implements some component of the security policy.

Principles of Integrity

Security principles are simply a collection of generally accepted standards of good practice that is thought to further the enforcement of security policies. Every organization is different and the interpretation and the adoption of principles will depend on specific circumstances. There are several principles for achieving and maintaining information integrity, but we are only going to focus on two basic principles that Clark and Wilson suggest are the most important. The principles are the well-formed transaction and separation of duty, which are abstracted from the Clark and Wilson papers [8].

- The principle of **separation of duty** states no single person should perform a task from beginning to end, but that the task should be divided among two or more people to prevent fraud by one person acting alone.
  Preserves the external consistency of data by ensuring that data in the system reflects the real-world data it represents.

- The principle of **well-formed transaction** is defined as a transaction where the user is unable manipulate data arbitrarily, but only in constrained (limitations or boundaries) ways that preserve or ensure the integrity of the data. A security system in which transactions are well-

formed ensures that only legitimate actions can be executed. Ensures the internal data is accurate and consistent to what it represents in the real world.

A Security Model for Integrity

Integrity models are used to describe what needs to be done to enforce the information integrity policies. There are three goals of integrity: (Summers, p.142)

- Prevent unauthorized modifications
- Maintain internal and external consistency
- Prevent authorized but improper modifications

To accomplish these goals, a collection of security services that embody the properties needed for integrity as well as a framework for composing them is needed. The needed security properties for integrity include integrity, access control, auditing, and accountability.

The Clark-Wilson (CW) model is an integrity, application level model which attempts to ensure the integrity properties of commercial data and provides a framework for evaluating security in commercial application systems. It was published in 1987 and updated in 1989 by David D. Clark and David R. Wilson (a computer scientist and an accountant). (Anderson, p.188)

Clark and Wilson partitioned all data in a system into two -constrained data items (CDI) and unconstrained items (UDI), data items for which integrity must be ensured. The (CDI) are objects that the integrity model is applied to and (UDI) are objects that are not covered by the integrity policy (e.g. information typed by the user on the keyboard). Two procedures are then applied to these data items for protection. The first procedure integrity verification procedure (IVP), verifies that the data items are in a valid state (i.e., they are what the users or owners believe them to be because they have not been changed). The second procedure is the transformation procedure (TP) or well-formed transaction, which changes the data items from one valid state to another. If only a transformation procedure is able to change data items, the integrity of the data is maintained. Integrity enforcement systems usually require that all transformation procedures be logged, to provide an audit trail of data item changes. [NCSC 1991] To provide a clear understanding of what this exactly mean, it is sufficiently valuable to have a look at a real world example:

1. Purchasing clerk creates an *order* for a supply, sending copies to the supplier and the receiving department.
2. Upon receiving the goods, a receiving clerk checks the delivery and, if all is well, signs a *delivery form*. Delivery form and original order go to the accounting dept.
3. Supplier sends an *invoice* to the accounting dept. Accounting clerk compares the invoice with original order and delivery form and issues a *check* to the supplier.

This example is presented in terms of **constrained data items** which are processed by **transformation procedures**. Data items are changed only by transformation procedures, thereby maintaining their integrity. The users are purchasing clerk, receiving clerk, supplier and accounting clerk. The transformation procedures (TP) are create order, send order, create delivery form, send delivery form, sign delivery form, create invoice, send invoice, compare invoice to order, and so on. The constrained data items are order, delivery form, invoice and

check. Users may only invoke some Transformation Procedures, and a pre-specified set of data objects or CDIs, as their duties see fit which enforces the notion of separation of duty.

To ensure that integrity is achieved and preserved, Clark and Wilson declare that certain integrity-monitoring (certification rules) and integrity-preserving rules (enforcement rules) are needed [6]. The integrity-monitoring rules are enforced by the administrator and the integrity-preserving rules are enforcement rules guaranteed by the system.

In the formulation in Amoroso [1] the nine rules for CW model are:

Certification Rules

- **C1 (IVP Certification)** - The system will have an IVP for validating the integrity of any CDI.
- **C2 (Validity)** - The application of a TP to any CDI must maintain the integrity of that CDI. CDIs must be certified to ensure that they result in a valid CDI
- **C3** - A CDI can only be changed by a TP. TPs must be certified to ensure they implement the principles of separation of duties & least privilege
- **C4 (Journal Certification)** - TPs must be certified to ensure that their actions are logged
- **C5** - TPs which act on UDIs must be certified to ensure that they result in a valid CDI

Enforcement Rules

- **E1 (Enforcement of Validity)** - Only certified TPs can operate on CDIs
- **E2 (Enforcement of Separation of Duty)** - Users must only access CDIs through TPs for which they are authorized.
- **E3 (User Identity)** - The system must authenticate the identity of each user attempting to execute a TP
- **E4 (Initiation)** - Only administrator can specify TP authorizations

The CW model differs from the other models that allow subjects to gain access to objects directly, rather than through programs. The *access* triple is at the heart of the CW model, (Summers, p.145) which prevents unauthorized users from modifying data or programs.
From what is presented above, we see that the CW model shows that the rules seek to enforce the needed security properties for integrity, which are described below: [18]

Integrity

An assurance that CDIs can only be modified in constrained ways to produce valid CDIs. This property is ensured by the rules: C1, C2, C5, E1 and E4.

Access control

The ability to control access to resources. This is supported by the rules: C3, E2 and E3.

Auditing

The ability to ascertain the changes made to CDIs and ensure that the system is in a valid state. This is ensured by the rules C1 and C4.

Accountability

The ability to uniquely associate users with their actions. This requires authentication of such users which is enforced by rule E3

Overview of Windows NT Security Model

Before we discuss an interpretation of the CW model, an overview of the Windows NT security model is in order.

Windows NT was built to incorporate networking, security and audit reporting as services within the operating system. The Windows NT Security Model was designed to monitor and regulate access to objects and it maintains security data for each user, group, and object.

Described in this section are the basic components of the Windows NT security model (Ivens & Hallberg, p. 40).

- Logon Process
- Local Security Authority (LSA)
- Security Account Manager (SAM)
- Security Reference Monitor (SRM)

Logon process, which accept logon request from users. It is the process that accepts the user's initial interactive logon, password, authenticates it, and grants entry into the system.

The **LSA** is the heart of the security subsystem. It verifies the logon information from the SAM database and ensures that the user has permission to access the system. It generates access token, administers the local security policy defined in the system and is responsible for auditing and logging security events.

Security Account Manager (SAM) is the database that contains information for all user and group account information and validates users.

Security Reference Monitor provides real-time services to validate every object access and action made by a user to ensure that the access or action is authorized. This part enforces the access validation and audit generation policy defined by the Local Security Authority.

Resources, such as processes, files, shares, and printers are represented in Windows NT as objects. Users never access these objects directly, but Windows NT acts as a proxy to these objects, controlling access to and usage of these objects. A subject in Windows NT is the combination of the user's access token plus the program acting on the user's behalf. Windows NT uses subjects to track and manage permission for the programs each user runs (Ivens & Hallberg, p. 42).

This is the most basic object in Windows NT, Security Identifiers (SIDs), are internal numbers used with a Windows NT system to describe a user and a group uniquely amongst other Windows NT systems. Owners, users or groups are assigned permissions to an object and are

identified by their SID. The security information for an object is encoded in a special data structure called the Security Descriptor (SD). The SD for an object contains the following components: (Minasi, p.1222)

- **Owner Security ID**
  This is the SID of the user or group who owns the object.

- **Group Security ID**
  This is a primary group associated with the object. It is optional and is not used by the Windows NT file-system system security. It is included to simplify the implementation of a POSIX-compliant file system.

- **Discretionary Access Control List**
  It identifies the user and group SIDs that are to be granted or denied access for the object

- **System ACL**
  This is what controls the auditing message that the system will generate.

Each user of Windows NT has a unique security ID (SID). When a user logs on, Windows NT creates a security access token. The token contains information about the user account which includes a security ID for the user, as well as other security IDs for the groups to which the user belongs, and permissions assigned to the user. The security access token created for the logged-in user is attached to all processes that are started by the user. When the process tries to access a particular object, the SRM checks to see whether any of the SIDs in the security access token attached to the process match a list called the access-control list (ACL) attached to that process. The ACL contains access-control entries (ACE) for each user authorized to access the object

Windows NT includes an auditing mechanism that can be used to audit successful and unsuccessful attempts for operations on files and directories. (Ivens & Hallberg, p.44) This mechanism enables you monitor events related to system security, to identify any security breaches, and to determine the extent and location of any damage.

Windows NT Interpretation of the Clark-Wilson model

An interpretation of the CW model in Windows NT is discussed in the following section [9].

Clark-Wilson Model	Windows NT Security Model
Rule 1. The system will have an IVP for validating the integrity of any CDI.	In Windows NT there is a local security authority (LSA) which checks the security information in the subject's access token with the security information in the object's security descriptor
Rule 2. The application of a TP to any CDI must maintain the integrity of that CDI	In Windows NT, most subjects cannot change the attribution of the objects, but some subjects have this privilege, such as administrator But this is only limited to some special users. So this rule is not applied to Windows NT strictly

Rule 3. A CDI can only be changed by a TP	As mentioned above some special users can change attribution of the objects, and no other methods can be applied to change objects
Rule 4. Subjects can only initiate certain TPs on certain CDIs	In windows NT, the subject's access token includes what kinds of operations are permitted. Only when information of the access token is consistent with the information in the object's security descriptor, the operation is allowed
Rule 5. CW-triples must enforce some appropriate separation of duty policy on subjects	In Windows NT, administrator can do anything. So this rule is not applied
Rule 6. Certain special TPs on UDIs can produce CDIs as output	In Windows NT, users can change the object from without ACL state to with ACL state. Generally, this operation is performed by Administrator
Rule 7. Each TP application must cause information sufficient to reconstruct the application to be written to a special append-only CDI	In Windows NT, audit services can collect information about how the system is being used
Rule 8. The system must authenticate subjects attempting to initiate a TP	In Windows NT, any user has her or his SID, and any process in behalf of this user copies the same SID. By this way, Windows NT can authenticate subjects attempting to initial a TP
Rule 9. The system must only permit special subjects (i.e., security officers) to make any authorization-related lists.	In Windows NT, only administrator can do and view some high security events

Based on the information presented above, it is easy to see that the security mechanisms of Windows NT satisfy the axioms of the CW model and that the CW model could be implemented with security mechanisms of Windows NT.

Summary

Integrity models may be implemented in several ways to satisfy the integrity requirements specified in a security policy. Model implementations describe how specific mechanisms can be employed in a system to ensure that the goals of the security policy are met. The Clark-Wilson model emphasizes how integrity is key to the commercial environment and it seeks to develop better security systems for that environment.

In general, it is important to recognize that by itself, a security model is not a panacea to information security issues. Security models have theoretical limits and do not establish security. So, why use models? The fact is that security models are generally used to evaluate existing secure system designs rather than a guide to developing secure systems. It is an effective method for verifying security. Security models are important and necessary, but focusing and relying only on a model can lead to a false sense of security.

Confidentiality, integrity, availability are very important and much related aspects of security. To achieve any of these goals, the objective is to strike a balance between applying generally accepted models and incorporating the latest security technologies and products, applying security patches, risk management, adhering to industry standards and guidelines, and implementing sound management principles to achieve secure systems. It is an on-going process.

References

Published Works:

[1] Amoroso, Edward. Fundamentals of Computer Security Technology. Prentice Hall, 1994.

[2] Summers, C. Rita. Computer Security: Threats and Safeguards. New York: McGraw Hill, 1997.

[3] Anderson, Ross, Security Engineering: A Guide to Building Dependable Distributed Systems. New York: Wiley Computer Publishing, 2001.

[4] Minasi, Mark. Windows NT Server 4, Sixth Edition. Alameda: SYBEX, Network Press, 1999.

[5] Ivens, Kathy and Hallberg, Bruce. Inside Windows NT Workstation 4. Indianapolis: New Riders Publishing, 1996.

[6] Krause, Micki, and Tipton F. Harold. Handbook of Information Security Management. CRC Press LLC, 1998.

Research Articles, White Papers and Workshops:

[7] National Computer Security Center Report 79–91."Integrity in Automated Information Systems." September 1991. URL: http://www.radium.ncsc.mil/tpep/library/rainbow/C-TR-79-91.pdf

[8] David D. Clark and David R. Wilson "A Comparison of Commercial and Military Computer Security Policies." IEEE Symposium of Security and Privacy, 1987, pages 184-194.

[9] Biba, K.J. "Integrity Considerations for Secure Computer Systems." Bedford, The MITRE Corporation, 1977.

[10] Xiao, Lei. "Clark-Wilson in Unix or NT. " Assignment 3. 9 January 1999. URL: http://www.tml.hut.fi/Opinnot/Tik-110.401/1999/Tehtavat/answer3.html

[11] Goguen, J.A. and J. Meseguer. Security Policies and Security Models. Proceedings of the 1982 Berkeley Conference on Computer Security, 11- 20. Berkeley, CA, 1982.

[12] Roskos, J.E., Welke, S.R., Boone, J., and Mayfield, T., "A Taxonomy of Integrity Models, Implementations and Mechanisms," Proceedings of the 13th National Computer Security Conference, pp. 541-551, October 1990. URL: http://www.radium.ncsc.mil/tpep/library/rainbow/NCSC-TG-010.pdf

[13] Welch, Ian "Reflective Enforcement of the Clark-Wilson Integrity Model", 2nd Workshop on Distributed Object Security. 2 November 1999. URL: http://www.cs.ncl.ac.uk/research/dependability/reflection/downloads/dos99.pdf

[14] SANS Related GSEC Practical

[15] Olovsson, Tomas. "A Structured Approach to Computer Security." Technical Report No. 122, 1992. URL: http://www.ce.chalmers.se/staff/ulfl/pubs/tr122to.pdf

[16] Bull, John and Rees, Owen. "A Framework for Federating Secure Systems." ANSA Phase III, 1994. URL: http://www.ansa.co.uk/ANSATech/93/Primary/10060002.pdf

[17] Pfleeger. "Trusted Operating System Design." ECE-C352 Lecture 8. (1999) URL: http://www.ece.drexel.edu/ECE/ECE-C352/lectures/lecture8.pdf

[18] "Prof. E. Stewart Lee, Director. "Essays about Computer Security." Centre for Communications Systems Research, Cambridge, 1999. URL: http://www.cl.cam.ac.uk/~mgk25/lee-essays.pdf

[19] MSDN Online Library, Microsoft Corporation, 1999.