WIKIPEDIA

Clark–Wilson model

The **Clark–Wilson** integrity model provides a foundation for specifying and analyzing an integrity policy for a computing system.

The model is primarily concerned with formalizing the notion of information integrity. Information integrity is maintained by preventing corruption of data items in a system due to either error or malicious intent. An integrity policy describes how the data items in the system should be kept valid from one state of the system to the next and specifies the capabilities of various principals in the system. The model uses security labels to grant access to objects via transformation procedures and a restricted interface model.

Contents

Origin

The model was described in a 1987 paper (*A Comparison of Commercial and Military Computer Security Policies*) by David D. Clark and David R. Wilson. The paper develops the model as a way to formalize the notion of information integrity, especially as compared to the requirements for multilevel security (MLS) systems described in the Orange Book. Clark and Wilson argue that the existing integrity models such as Biba (read-up/write-down) were better suited to enforcing data integrity rather than information confidentiality. The **Biba** models are more clearly useful in, for example, banking classification systems to prevent the untrusted modification of information and the tainting of information at higher classification levels. In contrast, **Clark–Wilson** is more clearly applicable to business and industry processes in which the integrity of the information content is paramount at any level of classification (although the authors stress that all three models are obviously of use to both government and industry organizations).

Basic principles

According to Stewart and Chapple's *CISSP Study Guide Sixth Edition*, the Clark–Wilson model uses a multi-faceted approach in order to enforce data integrity. Instead of defining a formal state machine, the model defines each data item and allows modifications through only a small set of programs. The model uses a three-part relationship of subject/program/object (where program is interchangeable with transaction) known as a *triple* or an *access control triple*. Within this relationship, subjects do not have direct access to objects. Objects can only be accessed through programs. Look here (http://www.pearsonitcertification.com/articles/article.aspx?p=1998558&seqNum=4) to see how this differs from other access control models.

The model's enforcement and certification rules define data items and processes that provide the basis for an integrity policy. The core of the model is based on the notion of a transaction.

- A **well-formed** transaction is a series of operations that transition a system from one consistent state to another consistent state.
- In this model the integrity policy addresses the integrity of the transactions.
- The principle of separation of duty requires that the certifier of a transaction and the implementer be different entities.

The model contains a number of basic constructs that represent both data items and processes that operate on those data items. The key data type in the Clark–Wilson model is a Constrained Data Item (CDI). An Integrity Verification Procedure (IVP) ensures that all CDIs in the system are valid at a certain state. Transactions that enforce the integrity policy are represented by Transformation Procedures (TPs). A TP takes as input a CDI or Unconstrained Data Item (UDI) and produces a CDI. A TP must transition the system from one valid state to another valid state. UDIs represent system input (such as that provided by a user or adversary). A TP must guarantee (via certification) that it transforms all possible values of a UDI to a “safe” CDI.

Rules

At the heart of the model is the notion of a relationship between an authenticated principal (i.e., user) and a set of programs (i.e., TPs) that operate on a set of data items (e.g., UDIs and CDIs). The components of such a relation, taken together, are referred to as a *Clark–Wilson triple*. The model must also ensure that different entities are responsible for manipulating the relationships between principals, transactions, and data items. As a short example, a user capable of certifying or creating a relation should not be able to execute the programs specified in that relation.

The model consists of two sets of rules: Certification Rules (C) and Enforcement Rules (E). The nine rules ensure the external and internal integrity of the data items. To paraphrase these:

> C1—When an IVP is executed, it must ensure the CDIs are valid.
>
> C2—For some associated set of CDIs, a TP must transform those CDIs from one valid state to another.

Since we must make sure that these TPs are certified to operate on a particular CDI, we must have E1 and E2.

> E1—System must maintain a list of certified relations and ensure only TPs certified to run on a CDI change that CDI.
>
> E2—System must associate a user with each TP and set of CDIs. The TP may access the CDI on behalf of the user if it is "legal".
>
> E3-The system must authenticate the identity of each user attempting to execute a TP.

This requires keeping track of triples (user, TP, {CDIs}) called "allowed relations".

> C3—Allowed relations must meet the requirements of "separation of duty".

We need authentication to keep track of this.

> C4—All TPs must append to a log enough information to reconstruct the operation.

When information enters the system it need not be trusted or constrained (i.e. can be a UDI). We must deal with this appropriately.

> C5—Any TP that takes a UDI as input may only perform valid transactions for all possible values of the UDI. The TP will either accept (convert to CDI) or reject the UDI.

Finally, to prevent people from gaining access by changing qualifications of a TP:

> E4—Only the certifier of a TP may change the list of entities associated with that TP.

CW-lite

A variant of Clark-Wilson is the CW-lite model, which relaxes the original requirement of formal verification of TP semantics. The semantic verification is deferred to a separate model and general formal proof tools.

See also

- Confused deputy problem

References

- Clark, David D.; and Wilson, David R.; *A Comparison of Commercial and Military Computer Security Policies* (http://theory.stanford.edu/~ninghui/courses/Fall03/papers/clark_wilson.pdf); in *Proceedings of the 1987 IEEE Symposium on Research in Security and Privacy (SP'87), May 1987, Oakland, CA*; IEEE Press, pp. 184–193
- Chapple, Mike; Stewart, James and Gibson Darril ; *Certified Information Systems Security Professional; Official Study Guide (8th Edition) 2018, John Wiley & Sons, Indiana*
- Shankar, Umesh; Jaeger, Trent; and Sailer, Reiner ; "Toward Automated Information-Flow Integrity Verification for Security-Critical Applications" (http://www.cse.psu.edu/~trj1/papers/ndss06.pdf); in "Proceedings of the 2006 Network and Distributed Systems Security Symposium (NDSS '06), February 2006, San Diego, CA“; Internet Society, pp. 267-280

External links

- Slides about Clark–Wilson used by professor Matt Bishop to teach computer security (http://nob.cs.ucdavis.edu/book/book-intro/slides/06.pdf)
- http://doi.ieeecomputersociety.org/10.1109/SP.1987.10001

Retrieved from "https://en.wikipedia.org/w/index.php?title=Clark–Wilson_model&oldid=1061054543"