Trusted Operating System Design I

Tom Kelliher, CS 325

Oct. 25, 2006

1 Administrivia

Announcements

Assignment

Read 5.3.

From Last Time

Exam I.

Outline

1. Trust
2. Security Policies

Coming Up

Security Models.

2 Trust

Operating systems enforce most of a system's security policies.

What is trust? How does one go about designing a trusted operating system? Elements of the process:

1. A stated security policy, as found in a requirements specification.
2. A model to study the interaction of "ordinary" system functionality with the security policy. Verify that *all* of the requirements specification is adhered to.
3. Design and implementation.
4. Observation that the working operating system has the features necessary to support the given security policy and development of confidence in those features.

Typical software engineering?

There are degrees of trust, based upon:

1. Enforcement of the given security policy.
2. Sufficiency of measures and mechanisms.
3. Evaluation — sense of trust developed over time.

3 Security Policies

1. Military — centralized and hierarchical.
2. Commercial — no so centralized and hierarchical.

3.1 Military Security Policy

A mechanism for maintaining confidentiality.

1. Sensitivity ranks: Unclassified, Restricted, Confidential, Secret, Top Secret.

2. Need to know rule: Access to sensitive data is allowed only to subjects who need to know the data to perform their jobs.

3. How do you prevent someone with Top Secret clearance from having access to all data, then?

4. Compartments (projects).

5. Classification of an object, or clearance of an individual is expressed as: $\langle \text{rank}; \text{compartments} \rangle$.

   Example: $\langle \text{Confidential}; \{\text{Red}, \text{Green}\} \rangle$.

6. For a given subject, S, to gain access to a given object, O, S must dominate O, as defined by: $S \geq O$ if and only if $rank_s \geq rank_o$ and $compartments_o \subseteq compartments_s$.

   Can S access O for each of the following?

   (a) S: $\langle \text{Restricted}; \{\text{Red}\} \rangle$. O: $\langle \text{Secret}; \{\text{Red}\} \rangle$.

   (b) S: $\langle \text{Top Secret}; \{\text{Red}\} \rangle$. O: $\langle \text{Secret}; \{\text{Red}, \text{Green}\} \rangle$.

   (c) S: $\langle \text{Secret}; \{\text{Red}, \text{Green}, \text{Blue}\} \rangle$. O: $\langle \text{Secret}; \{\text{Red}, \text{Green}\} \rangle$.

3.2 Commercial Security Policies

3.2.1 Clark-Wilson

A mechanism for maintaining integrity.

1. A policy for *well-formed transactions*, maintaining integrity between data items.

2. The well-formed transactions themselves have an integrity property — who is authorized to perform them?

3. Definitions:

   (a) Constrained data item (CDI): data subject to integrity controls.

       Example: Bank account balances.

   (b) Integrity verification procedures (IVP): test CDIs to ensure they conform to integrity guidelines.

       Example: Confirming that accounts are balanced.

   (c) Transformation procedures (TP): procedures that transform a set of CDIs from one valid state to another.

       Example: Transferring money from one account to another.

4. A well-formed transaction captures all of this in a triple:

$$\langle \text{user}, TP_i, \{CDI_j, CDI_k, \ldots\} \rangle$$

3.2.2 Separation of Duty

This is the idea of separating responsibilities so as to avoid the possibility of fraud.

Example: In a business, more than one person should handle invoice receipt, purchase confirmation, and check cutting.

3.2.3 Chinese Wall

A mechanism for ensuring confidentiality and the prevention of conflicts of interest.

1. Partition information (accounts) into conflict of interest sets.

   Example:

   (a) Set 1: Bank of America, M and T Bank, Wachovia Bank.

   (b) Set 2: Mobil, Shell, Sunoco, Amoco.

2. Once a subject accesses information from one element in a set, it may not access information from any other element in that set.