Application-Oriented Policies and their Composition

Virgil D. Gligor and Serban I. Gavrila

April 16, 1998

• Systems and Applications

• Property Types; Dependencies

• Policy Structure

• Policy Composition
Systems

- *state machine*
  
  STATES, SUBJECTS, USERS, OPERATIONS, OBJECTS

- *state transitions*
  
  - commands: \( op(s_1, S, obj, s_2) \)
  
  - command sequence: \( op_1(s_0, S_1, obj, s_1)op_2(s_1, S_2, obj_2, s_2)... \)
  
  - tranquil commands: do not alter security attributes

- *system*: a set of command sequences with start states \( s_0 \) in STATES\(_0\).

- *secure state, commands*: those that satisfy properties

- *reachable state*: a state appearing in a command sequence of a system

- *secure system*: all state transitions and reachable states are secure

- \( \Omega \): set of all command sequences of a secure system
Applications and Executability

• application: $\text{App} = \{\text{ObjSet}, \text{OpSet}, \text{Plan}\}$

  - plan: a finite set of pairs $\{(\text{obj}_i, \text{op}_i)\}$
  - ordered plan: an ordered set of pairs $\{(\text{obj}_i, \text{op}_i)\}$
  - plans with “operation bracketing” (e.g., least-privilege princ.)

• $\text{App}_1 \cup \text{App}_2 = \{\text{ObjSet}_1 \cup \text{ObjSet}_2, \text{OpSet}_1 \cup \text{OpSet}_2, \text{Plan}_1 \cup \text{Plan}_2\}$

• command sequence $\sigma$ executes $\text{App}$ if for any pair $(\text{obj}_i, \text{op}_i)$ in Plan there is a command $\text{op}_i(s_k, S, \text{obj}_i, s_{k+1})$ in $\sigma$
Property Types

\[ P = \text{Attribute (AT) properties} \land \text{Access Management (AM) properties} \land \text{Access Authorization (AA) properties} \]
Examples of Property Types

• Attribute (AT) Properties
  – security (integrity) levels, partial order, lattice property
  – roles, hierarchy, permissions, membership, inheritance

• Access Management (AM) Properties
  – distribution, review, revocation of permissions
    • selectivity, transitivity, independence ...
  – object / subject creation and destruction
  – object encapsulation

• Access Authorization (AA) Properties
  – required subject and object attributes for access
    • BLP, Biba, RBAC, UNIX ...
Property Dependencies

“uses”

other types of dependencies exist

Individual policy properties cannot be composed independently
Policy Structure

\[ P = P \land \text{Admin} (P) \land \text{Compat}(P, \text{App}) \]

- access management
- access authorization
- attribute properties

Safety Properties

Safety or Liveness Properties ?
Admin(P)

P: a set of tranquil command sequences with the start state in STATES_0

for all

Admin(P) = “for each s in STATES, there exists s_0 ∈ STATES_0, there exists ω ∈ Ω such that: ω starts in s, and ω reaches s_0 and s_0* is in P”

Compat(P, App)

Compat(P) = “there exists s_0 ∈ STATES_0 and σ ∈ P starting in s_0 such that σ executes App”

.... neither Safety nor Liveness ....
Mandated Compatibility

 USERS

 Policy Definition and Administration

 Application Operations

 Application Definition and Administration

 OBJECTS

 Permissions

 Compatibility
Types of Compatibility

- **Totally multi-path Compatible**
- **Multi-path Compatible**
- **Machine Closed Compatible**
- **Strongly Compatible**

Compat(P, App)
Totally Multi-path Compatible
For each start state $s_0$ there is a command sequence $\sigma$ in $P$ starting in $s_0$, and for each finite command sequence $\sigma$ in $P$ there is a command sequence $\tau$ such that $\sigma \tau$ is in $P$ and executes $App$.

Machine-Closed Compatible
For each finite command sequence $\sigma$ in $P$ there is a command sequence $\tau$ such that $\sigma \tau$ is in $P$ and executes $App$.

Multi-path Compatible
There is a start state $s_0$ such that for each finite command sequence $\sigma$ in $P$ starting in $s_0$ there is $\tau$ such that $\sigma \tau$ is in $P$ and executes $App$.

Totally Compatible
For each start state $s_0$ there is a command sequence $\sigma$ in $P$ starting in $s_0$ such that $\sigma$ executes $App$.

Strongly Compatible
For each start state $s_0$ such that $s_0^*$ is in $P$, there is a command sequence $\sigma$ in $P$ starting in $s_0$ that executes $App$.

Compatible
There is a start state $s_0$ and a command sequence $\sigma$ in $P$ starting in $s_0$ that executes $App$. 
Types of Compatibility

- **Totally multi-path Compatible**
- **Multi-path Compatible**
- **Machine-Closed Compatible**
- **Overly Restrictive STATES \( \sigma \)**
- **Overly Restrictive \( \sigma \)**s
- **Strongly Compatible**

**Compat(P, App)**

*May Require Administrative Work for App’s Execution in P*
Overly Restrictive $\sigma_s$

Example:

$App = \{ \{obj\}, \{op_1, op_2\}, plan\}; plan = \{(obj, op_1), (obj, op_2)\}$

$P:$ “$u_1$ and $u_2$ are the only users who may execute $App$ and

a user may not execute two distinct operations on the same object”

Compat($P$, $App$) is true

\[
\begin{align*}
\sigma &= S_1: (op_1, obj) \\
S_0 &\rightarrow s_1 \\
S_1 &\rightarrow s_2
\end{align*}
\]

$u_1: (op_1: obj)$, $S_1$ = subject

$u_2: (op_1, op_2 : obj)$, $S_2$, $S_2'$ = subjects

Compat$_M$(P, App) is false

\[
\begin{align*}
\sigma' &= S_2': (op_1, obj) \\
S_0 &\rightarrow s_1' \\
\tau &= S_2': (op_2, obj) \\
S_1 &\rightarrow \times
\end{align*}
\]

$u_1: (op_1: obj)$, $S_1$

$u_2: (op_1, op_2 : obj)$, $S_2$, $S_2'$

$S_1: (op_2, obj)$
Policy Composition

\[ P_1 = P_1 \land \text{Admin}(P_1) \land \text{Compat}(P_1, \text{App}_1) \]
\[ P_2 = P_2 \land \text{Admin}(P_2) \land \text{Compat}(P_2, \text{App}_2) \]

Let \( \text{CS}(P_1), \text{CS}(P_2) \) denote sets of command sequences

\( P_1, P_2 \) are composable if and only if
\( \text{CS}(P_1 \cap P_2) \neq \emptyset \) whenever \( \text{CS}(P_1), \text{CS}(P_2) \neq \emptyset \)

Emerging policy \( P_1 \circ P_2 = \)
\[ = P_1 \land P_2 \land \text{Admin}(P_1 \land P_2) \land \text{Compat}(P_1 \land P_2, \text{App}_1 \cup \text{App}_2) \]
Example: Non-Composable Separation-of-Duty Policies

Static SoD

R2
- verify
- read
- write
- sign

R1
- read
- write

R3
- sign

Operational Static SoD

R2
- verify
- read
- write
- sign

R1
- read
- write

R3
- sign

Purchasing Staff Department

Purchasing Staff Central Administration

Policy-Management Change