



The Goguen Meseguer Security Model

George W. Dinolt
CS4605

Elements

- Let \mathcal{S} be the set of *States* the system can take on
- Let $s_0 \in \mathcal{S}$ be the *initial state*
- Let \mathcal{U} be the set of *Users* of the system
- Let \mathcal{C} be the set of *Commands* that can be issued by *Users* to cause a change of state
- Let OUT be the user visible outputs of the system
- Let \mathcal{L} be the set of security labels.

Functions

do : The state transition function

$$do : \mathcal{S} \times \mathcal{U} \times \mathcal{C} \rightarrow \mathcal{S}$$

Constructs a new state from a state, a user and a command

out : The output visible from a state

$$out : \mathcal{S} \times \mathcal{U} \rightarrow \mathcal{OUT}$$

Sequence Notation

Sequences: We let $\mathcal{W} = (\mathcal{U} \times \mathcal{C})^*$ be the set of all possible sequences of pairs of elements (u, c) where $u \in \mathcal{U}$ and $c \in \mathcal{C}$. $w \in \mathcal{W}$ might be written as

$$w = \langle (u_0, c_0), (u_1, c_1), \dots, (u_n, c_n) \rangle$$

Null Sequence: The sequence consisting of no elements, the empty or *Null* sequence is sometimes denoted by $\langle \rangle$.

Concatenation: If $w = \langle (u_0, c_0), (u_1, c_1), \dots, (u_n, c_n) \rangle$ then we may write rewrite w as

$$w = \langle (u_0, c_0), (u_1, c_1), \dots, (u_{n-1}, c_{n-1}) \rangle \cdot (u_n, c_n)$$

where “.” is sequence concatenation operator.

Operating on Inputs

Extending do to sequences: Suppose $x \in \mathcal{W}$ and $w = x \cdot (u, c) \in \mathcal{W}$, then we can extend do to sequences, $do : \mathcal{S} \times \mathcal{W} \rightarrow \mathcal{S}$ by:

$$do(s_0, w) = \begin{cases} s_0 & \text{if } w = \text{Null}, \\ do(do(s_0, x), u, c) & \text{if } w = x \cdot (u, c) \end{cases} \quad (1)$$

do applied to a sequence of inputs is the state that results as a consequence of applying the inputs one after another in sequence.

We will use the notation:

$$[[w]] = do(s_0, w)$$

Outputs From Sequences

If $w \in \mathcal{W}$ then we will use the notation

$$[[w]]_u = \text{out}([[w]], u)$$

Purge Users

Let $G \subseteq \mathcal{U}$, $w \in \mathcal{W}$, the purge of G from w is:

$$P_G(w) = \begin{cases} \text{Null} & \text{if } w = \text{Null} \\ P_G(x) & \text{if } w = x \cdot (u, c) \text{ and } u \in G \\ P_G(x) \cdot (u, c) & \text{if } w = x \cdot (u, c) \text{ and } u \notin G \end{cases}$$

i.e. $P_G(w)$ is the subsequence of w that has had all references to commands that are issued by users in G removed.

Non-Interference

Suppose $\{\mathcal{S}, \mathcal{U}, \mathcal{C}, \mathcal{O}UT, do, out\}$ is a system and suppose $G \subseteq \mathcal{U}$ and $\mathcal{W} = (\mathcal{U} \times \mathcal{C})^*$, then the users in G do not interfere with the other users of the system if

$$\forall w \in \mathcal{W}, \forall u \in \mathcal{U} \setminus G, [[w]]_u = [[P_G(w)]]_u$$

i.e. purging the actions of the set of users G does not change the view of the system seen by the other users of the system.

If G are the “high users” of the system, then the actions of the high users will not be visible to the other (lower) users.