

Questo homework implementa un semplice linguaggio funzionale dotato di ricorsione, di un costrutto “Guard(policy, exp)” per controllare una politica history-based access control a runtime, un costrutto “Define(name, transitions, final, exp)” per definire nuove politiche. È possibile fornire all’interprete politiche già definite. L’interprete tiene attiva una sola istanza di ogni politica. Le politiche riguardano un set definito di eventi di sicurezza sul filesystem.

Le politiche sono definite da automi a stati finiti che accettano le politiche violate. Per rendere compatta la definizione degli automi l’alfabeto è unico ed è composto dagli eventi di sicurezza, gli stati sono impliciti nella tabella di transizione, lo stato iniziale è sempre 0, bisogna dichiarare lo stato di accettazione (violazione della politica), bisogna dichiarare solo le transizioni rilevanti, la transizione di default rimane nello stato corrente.

A runtime è presente una struttura dati “state” che rappresenta lo stato dell’analisi, composto dalla pila delle politiche attive, la pila delle politiche disponibili e la storia degli eventi rappresentata da una coda unica. Quando un nuovo evento si verifica gli automi che rappresentano le politiche attive vengono aggiornati in ordine LIFO.

Quando una nuova politica viene richiesta, se non già presente, il suo automa viene inizializzato e aggiornato con gli eventi della storia.