LIO Language Update

Wesley Nuzzo

May 23, 2023

1 Update of figure 2 (syntax)

Bool: b
Policy:

$$p ::= l \mid p \nearrow^a p \mid p \searrow^a p$$

Term:

$$v ::= \texttt{true} \mid \texttt{false} \mid () \mid p \mid a \mid x \mid \lambda x.e \mid (e, e) \mid \texttt{fix}\ e \mid \texttt{Lb}\ v\ e \mid (e)^{\texttt{LIO}} \mid \bullet \mid \texttt{Cnd}\ v\ b$$

Expression:

$$e ::= \ldots \mid \texttt{newCond}\ e\ e \mid \texttt{readCond}\ e \mid \texttt{setCondTrue}\ e \mid \texttt{labelOfCond}\ e$$

2 Two different paradigms

There are two different possible paradigms for implementing this concept.

The first paradigm is the one presented in Chong's thesis: no memory location can change it's policy, but a declassification policy can be flow to it's secondary policy if it's condition is met, an erasure policy can flow to the join of its primary and secondary policy at all times, and data must be deleted if it requires erasure (i.e., it has an erasure policy whose condition is met or its primary policy requires erasure).

The second paradigm is a bit simpler: the data doesn't change, but the policy does change "under" the values as the conditions update.

3 Semantics

3.1 paradigm 2 (updating policy)

I'm still working on this one.

$$\text{Label: } \frac{\Sigma.lbl \sqsubseteq l \sqsubseteq \Sigma.clr}{\langle \Sigma, E[\texttt{label } p\ e] \rangle \longrightarrow \langle \Sigma, E[\texttt{return } (\texttt{Lb } p\ e)] \rangle}$$

$$\text{Unlabel: } \frac{p' = \Sigma.lbl \sqcup update(p, \Sigma) \quad p' \sqsubseteq \Sigma.clr \quad \Sigma' = \Sigma[lbl \mapsto p']}{\langle \Sigma, E[\texttt{unlabel } (\texttt{Lb } p\ e)] \rangle \longrightarrow \langle \Sigma', E[\texttt{return } e)] \rangle}$$

—

$$\text{SetCondTrue } \frac{\Sigma.\phi(a) = \texttt{Cond } p\ e \qquad \Sigma' = \Sigma.\phi[a \mapsto \texttt{Cond } p\ \texttt{true}] \quad \Sigma.lbl \sqsubseteq p \sqsubseteq \Sigma.clr \quad \Sigma'' = \Sigma[lbl \mapsto update(\Sigma'.lbl, \Sigma')]}{\langle \Sigma, E[\texttt{setCondTrue } a] \rangle \longrightarrow \langle \Sigma'', E[\texttt{return } ()] \rangle}$$

3.1.1 helper function: *update*

$$update(l, \Sigma) = l$$

When c is true:

$$update(p \searrow^c q, \Sigma) = p \sqcap (pol(c) \sqcup q)$$

$$update(p \nearrow^c q, \Sigma) = p \sqcup q$$

When c is false:

$$update(p \searrow^c q, \Sigma) = p \searrow^c q$$

$$update(p \nearrow^c q, \Sigma) = q \sqcap (pol(c) \sqcup p \nearrow^c q)$$