A Proofs

To prove the soundness theorem, we show that each execution step of a node under the monitor generates secure events.

Lemma 1. *Let* $N_k = \langle config, wires, l, P, V, S\rangle_k$ *be a node. Any semantic step of* N_k *under the monitor produces a secure trace with regard to* $\langle P_k, V_k, S_k\rangle$*, i.e.,* $\forall N_k.\ config_k \xrightarrow{T_k}_{\mathcal{M}} config'_k \Rightarrow secure(T_k)$.

Proof. First we show that any trace produced from the expression evaluation rules is secure. By induction on the derivation $\langle e, M_k\rangle \Downarrow_{\mathcal{M}} v$:

- The rule (Value) generates an empty (secure) trace.

- The rule ($\text{Read}_{\mathcal{M}}$) only generates the event $R_k(x)$ if it meets the security condition for reading a variable, i.e., $secure(R_k(x), \langle P_k, V_k, S_k\rangle)$.

- In the rule ($\text{Call}_{\mathcal{M}}$), by the induction hypothesis, $\langle e, M_k\rangle \Downarrow^{T_k}_{\mathcal{M}} v \Rightarrow secure(T_k)$. Then, the trace $T_k.f_k(v)$ is generated if the API call and the value of the expression e obeys the security condition for API calls, i.e., $secure(f_k(v), \langle P_k, V_k, S_k\rangle)$. Therefore, $secure(T_k) \wedge secure(f_k(v), \langle P_k, V_k, S_k\rangle) \Rightarrow secure(T_k.f_k(v), \langle P_k, V_k, S_k\rangle)$.

Next, by induction on the derivation $config_k \xrightarrow{T_k}_{\mathcal{M}} config'_k$, we prove the lemma:

- Rules (Input), (Skip), and (Seq-2) generate empty traces, which are trivially secure.

- Rules (If), (While-T), (While-F) and (Output) generate the same trace resulting from the expression evaluation $\langle e, M_k\rangle \Downarrow^{T_k}_{\mathcal{M}} v \Rightarrow secure(T_k)$, because of the proof above.

- The trace T_k generated in Rule (Seq-1) is secure, based on the induction hypothesis.

- The rule ($\text{Write}_{\mathcal{M}}$) emits a secure trace since $\langle e, M_k\rangle \Downarrow^{T_k}_{\mathcal{M}} v \Rightarrow secure(T_k)$, and $secure(T_k) \wedge secure(W_k(x), \langle P_k, V_k, S_k\rangle) \Rightarrow secure(T_k.W_k(x), \langle P_k, V_k, S_k\rangle)$. Because any trace generated by the rules of expression evaluation $\langle e, M_k\rangle \Downarrow_{\mathcal{M}} v$ is secure, and the write event is produced only if it complies with the security condition for writing into a variable, i.e., $secure(W_k(x), \langle P_k, V_k, S_k\rangle)$.

We have proved the node-level security as a corollary of Lemma 1. Hence, the generated trace from a transition between any two node configurations is secure. Next, we prove that any trace generated by a flow execution under the monitor is secure.

Lemma 2. *Any semantic step of a flow* F_l *under the monitor produces a secure trace,* $\forall F_l, F'_l.\ F_l \xrightarrow{T_F}_{\mathcal{M}} F'_l \Rightarrow secure(T_F)$.

Proof. By case analysis on the flow semantics rules:

- The rules (Init) and (Term) yield empty (secure) traces, which are trivially secure.

- The rules (STEP) and (SEND) repeat the same trace generated from the corresponding transition between node configurations. Lemma 1 demonstrates that $\forall N_k.\ config_k \xrightarrow{T_k}_{\mathcal{M}} config'_k \Rightarrow secure(T_k)$. Thus, the theorem also holds for these cases.

Lemma 3. *Let G be a global configuration. Any semantic step of G under the monitor is secure, $\forall G, G'.\ G \xrightarrow{T_G}_{\mathcal{M}} G' \Rightarrow secure(T_G)$.*

Proof. The single rule in the global semantics replicates the trace produced by the transition between the two flow configurations. Lemma 2 shows flow transitions are secure under the monitor, thus the global transitions. Because $\big(\forall G, G'.\ G \xrightarrow{T_G}_{\mathcal{M}} G' \Rightarrow secure(T_G)\big) \Leftrightarrow \big(\forall F_l, F'_l.\ F_l \xrightarrow{T_F}_{\mathcal{M}} F'_l \Rightarrow secure(T_F)\big)$.

Proof (Theorem 1). By using the lemma 3 and multiple repetitions of the single rule of the global semantics, the soundness theorem is proven as a corollary.

To prove the transparency theorem, we show that the monitor preserves the secure events emitted from a node.

Lemma 4. *Any semantic step in the original execution of a node that emits a secure trace remains the same in the monitor semantics, $\forall N_k, N'_k.conf_k \xrightarrow{T_k} conf'_k \wedge secure(T_k) \Rightarrow conf_k \xrightarrow{T_k}_{\mathcal{M}} conf'_k$.*

Proof. By induction on $\langle e, M_k\rangle \Downarrow v$, we observe that there is a one-to-one mapping from the rules for $\Downarrow$ and $\Downarrow_{\mathcal{M}}$ if the security conditions $secure(R_k(x), \langle P_k, V_k, S_k\rangle)$ and $secure(f_k(v), \langle P_k, V_k, S_k\rangle)$ hold.

By induction on the derivation $conf_k \xrightarrow{T_k} conf'_k$, again we can see a one-to-one correspondence between the rules for $\rightarrow$ and $\rightarrow_{\mathcal{M}}$, as a result of the induction on $\langle e, M_k\rangle \Downarrow v$, and the comparison between the rule (WRITE) in the standard semantics and the rule ($\text{WRITE}_{\mathcal{M}}$) in the monitor semantics, which requires $secure(W_k(x), \langle P_k, V_k, S_k\rangle)$ to be held.

We assume utilizing a deterministic order-preserving scheduler that both the original semantics and the monitor employ. The non-deterministic scheduler might affect the order of events generated by the global and flow transitions.

Lemma 5. *Any semantic step of the global configuration that generates a secure trace remains the same in the monitor semantics, $\forall G, G'.\ G \xrightarrow{T_k} G' \wedge secure(T_k) \Rightarrow G \xrightarrow{T_k}_{\mathcal{M}} G'$.*

Proof. The standard and the monitor semantics use the same global and flow semantics. With the assumption of employing an identical deterministic scheduler and using lemma 4, we can write $\forall G, G'.\ G \xrightarrow{T_k} G' \wedge secure(T_k) \Rightarrow \exists! F_l, N_k, F'_l, N'_k.\ F_l \in Flows(G) \wedge N_k \in Nodes(F_l) \wedge F'_l \in Flows(G') \wedge N'_k \in Nodes(F'_l) \wedge conf_k \xrightarrow{T_k}_{\mathcal{M}} conf'_k$. Similarly, the statement holds for $\xrightarrow{T_k}_{\mathcal{M}}$ in the other way.

Proof (Theorem 2). Starting with the initial configuration (G_0, V_{init}) and using the global semantics, there are two cases:

- Case 1 (the trace is secure): If $secure(T)$, using the lemma 5 for n-times results $T = T' \wedge n = m$.

- Case 2 (the trace is not secure): If $T = T_{pre} \,.\, T_i \,.\, T_{post}$ where $secure(T_{pre}) \wedge \neg secure(T_i)$, then using the lemma 5 for i times concludes $T' = T_{pre} \wedge i = m$. Thereafter, no semantic rule applies for the transition $G_i \xrightarrow{T_{pre}}{}^{i} G_{i+1}$ in the monitor semantics.