Language Based Technology For Security

Cybersecurity LM-66

Samuele Padula

Giugno 2021

1 Risposta alle domande

1.1 Che cosa raccolgono le componenti ρ e κ nella CFA vista del Pi Calcolo?

ρ contiene per ogni etichetta del sistema il binding ai nomi che puo' assumere in qualsiasi momento durante l'esecuzione mentre κ rappresenta i nomi che possono essere inviati sul canale. Chiaramente il valore di ρ dipende sostanzialmente da cosa può essere spedito sul canale (κ) che è quello che fa da soggetto all'azione di input. Il ρ di un nome libero mi restituirà lo stesso nome ma il ρ di una variabile invece mi restituirà i possibili legami che avrà a tempo di esecuzione

1.2 Quale potrebbe essere il risultato dell'analisi nell'esempio sotto nelle componenti ρ e κ ?

$$P \mid Q \mid R = (\overline{a}d.P' + \overline{a}b.P'' \mid R \mid a(w).\overline{w}c.Q')$$

- ρ(a) $\supseteq$ {a},
- ρ(b)$\supseteq${b},
- ρ(c)$\supseteq${ c },
- ρ(d)$\supseteq${ d },
- ρ(w)$\supseteq${ d,b }
- κ(a)$\supseteq${d,b},
- κ(b)$\supseteq${c},
- κ(c)$\supseteq${ 0 },
- κ(d)$\supseteq${ c },
- κ(w)$\supseteq${ c }

1.3 Se SecretNames = {a, d} and that PublicNames = {b, c}. Cosa ci dice l'analisi della proprietà di segretezza discussa a lezione?

La nozione statica ci dice che ogni elemento segreto non puo' fluire su canali pubblici. Inoltre se è un sistema è confinato (nozione statica) allora tale sistema è sicuro (nozione dinamica). Essendo:

$\kappa(\mathrm{b}) \subseteq PublicNames, \kappa(\mathrm{c}) \subseteq PublicNames| = \Phi \implies Segretezza\ rispettata$

Il canale risulta dunque sicuro.

1.4 A cosa serve la componente ψ?

ψ è l'insieme delle violazioni alla sicurezza (nel contesto specifico del tracciamento di origine e destinazione indica un eventuale incongruenza nell'origine e/o destinazione attesa) che consente di analizzare un flusso di informazioni e verificarne la conformità con i risultati attesi. Se al termine del processo di analisi (ρ , κ)|= P : $\emptyset$, ovvero non sono presenti mismatch di origine e destinazione dell'informazione, abbiamo la certezza che in fase dinamica (a run time) un eventuale reference monitor non bloccherà l'esecuzione.

1.5 Quali sono le maggiori differenze tra le soluzioni CFA di LySa e quelle di IoT LySa?

IoT Lysa permette di verificare le interazioni tra diversi nodi di una rete, di tracciarne il flusso di informazioni e di valutarne le proprietà di confidenzialità e integrità. L'analisi della stima di LySa si basa su (ρ , κ)|= $\in \psi$ mentre l'analisi di Iot LySa si basa su ($\hat{\Sigma}$, κ, Θ) valutato nel nodo l. In particolare $\hat{\Sigma}$ è la abstract store che rappresenta, per ogni variabile di ciascun nodo, un astrazione dei legami possibili a run-time e Θ che rappresenta un astrazione dei dati che vengono raccolti e calcolati da ciascun nodo. Quindi sostanzialmente l'algebra IoT LySa estende LySa attraverso reti di nodi, quindi aggiungendo un livello di gerarchia utilizzando e modellando sensori e attuatori, avendo comunicazioni di tipo multi output e avendo una comunicazione locale tracciando i dati per capire come interagiscono i nodi, ma soprattutto quali sono i flussi dei dati indipendentemente dai valori concreti che i dati hanno, sulla base di criteri di sicurezza che possono essere ad esempio dati provenienti o diretti verso nodi untrusted. .

1.6 Che cosa ci permettono di tracciare i valori astratti?

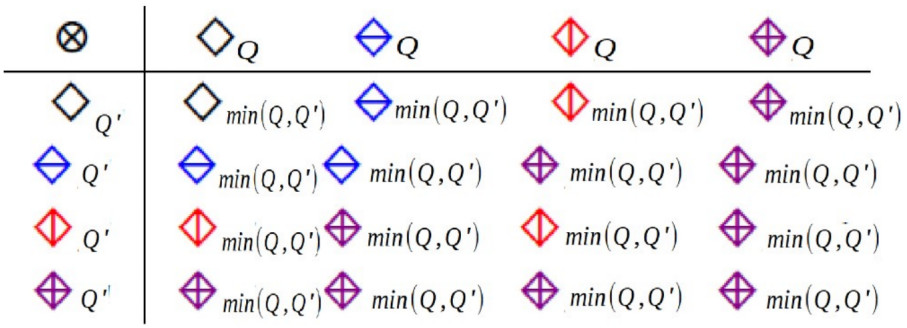
I valori astratti che Iot LySa comprende sono:

- Costanti
- Dati dai sensori
- Attributi dei dati (i.e. tag)
- Funzioni e aggregazioni di dati astratti

In base a questo IoT LySa rispetto a LySa ci permette di tracciare il flusso dell'informazione tramite i tag che vengono trasportati fino ai nodi supervisori della gerarchia. In questo modo possiamo valutare proprietà come la taintness.

2 Taint and Resilience Analysis a là qualità

Nell'ottica di integrare l'analisi statica relativa alla qualità e alla taintess, usando i costrutti definiti è stato possibile integrare la CFA di IoT LySa al concetto di fault tolerance, mantenendo invariati i concetti di tamperability e sensitivity. L'operazione di join è stata modificata come segue:



Sono stati estesi i termini astratti con i tag di qualita' "q" e il valore astratto () che implementa la funzione di OR logico sull'insieme di valori astratti esaminati valutando il tag "q":

$$\hat{V} \ni \hat{v} = \begin{cases} (T, b, q) \\ (v, b, q) \\ (f(\hat{v_1}, ..., \hat{v_k}), b, q) \\ (\{\hat{v_1}, ..., \hat{v_k}\}_{k0}, b, q) \\ (\&_{\exists}(\hat{v_1}, ..., \hat{v_r}), b, q) \end{cases}$$

Sono stati definiti due nuovi insiemi, una nuova funzione di assegnamento e un ordinamento. Gli insiemi Hi e Lo contengono i sensori di qualita' alta(affidabile) e bassa(inaffidabile):

$$y \in Lo < y' \in Hi$$

Basandosi sulla funzione di assegnamento della taintness originale:

$$\tau(y, \ell) = \begin{cases} \textcircled{\Diamond} & if\ y \in \mathcal{S}_\ell \\ \textcircled{\Diamond} & if\ y \in \mathcal{T}_\ell \\ \textcircled{\Diamond} & if\ y \in \mathcal{S}_\ell \cap \mathcal{T}_\ell \\ \Diamond & o.w. \end{cases} \qquad where\ y \in \mathcal{I}_\ell \cup \mathcal{X} \qquad \tau(v, \ell) = \Diamond$$

è stata definita una nuova funzione di assegnamento che tiene conto anche della qualità dei sensori:

$$\tau'(y,l) = \begin{cases} (\tau(y,l), hi) & \text{if } y \in Hi \\ (\tau(y,l), lo) & \text{if } y \in Lo \end{cases}$$

$$\tau'(v,l) = (\tau(v,l), hi)$$

Si è deciso di aggiungere la seguente policy di propagazione dove viene eseguito un prodotto parziale tra i valori di livello "hi" (se ne esiste almeno uno) altrimenti viene eseguito il prodotto parziale normalmente su tutti i valori:

$$\&_{(\exists\tau)}(f, \hat{v_1}, ..., \hat{v_r}) = \begin{cases} \otimes(\hat{v_{1\downarrow(2,3)}}, ..., \hat{v_{r\downarrow(2,3)}}) & \forall \hat{v_{i\downarrow 3}} = hi \ if \ \exists \hat{v_{j\downarrow 3}} = hi \ , 1 \leq i, j \leq r \\ \otimes(\hat{v_{1\downarrow(2,3)}}, ..., \hat{v_{r\downarrow(2,3)}}) & \text{o.w.} \end{cases}$$

Questo è un esempio di propagazione dei valori:

$$v^{(\&_{\exists\tau}(avg, v_0^{(\Phi, lo)}, v_1^{(\diamond, lo)}, v_2^{(\diamond, hi)}, v_3^{(\diamond, lo)}))} = v^{(\diamond, hi)}$$

Le clausole di stima sono state modificate nel modo seguente:

$$\frac{\tau'(x,l) \otimes \hat{\Sigma}_l(x) \subseteq \Theta(l)(a)}{(\hat{\Sigma}, \Theta) \models_l x^a}$$

Nella semantica è stata modificata la premessa relativa all'input, lasciando invariato l'output. Tale decisione deriva dal fatto che il controllo viene eseguito solo in fase di ricezione del valore:

(1)

$$\frac{\begin{array}{c} \bigwedge_{i=1}^{j} (\hat{\Sigma}, \Theta) \models_l M_i^{a_i} \wedge \forall (l', \langle \hat{v_1}, ..., \hat{v_r} \rangle) \in \\ \kappa(\mathrm{l}) : \mathrm{Comp}(l', l) \implies (\bigwedge_{i=j+1}^{r} \hat{v_i} \otimes \tau'(x_i, l) \in \hat{\Sigma}_l(x_i) \wedge (\hat{\Sigma}, \kappa, \Theta) \models_l P) \end{array}}{(\hat{\Sigma}, \kappa, \Theta) \models_l (M_i^{a_1}, ..., M_j^{a_j}; x_{j+1}^{a_{j+1}}, ..., x_r^{a_r}).P}$$

[Definizione di validità] Dato N un sistema di nodi con etichetta l in L e Hi l'insieme dei sensori affidabili ed L_c l'insieme dei nodi critici di etichette l_c. Il sistema di nodi N è valido se vale la seguente condizione:

$$\frac{\forall l_c \in L_c : (x)_{\downarrow 3} \in \{hi\}\ \forall x \in \hat{\Sigma}_{l_c}}{(\hat{\Sigma}, \kappa, \Theta) \models_{l\ c} N}$$

Nell'esempio grafico, sulla sinistra abbiamo che in nessun nodo è presente la $\&_\exists()$. I sensori S_0 ,S_1,S_3 sono di qualità lo mentre S_2 è di qualità hi. Il tag di taint viene propagato fino al nodo l_3 in quanto basta un sensore taggato taint per "sporcare" la propagazione. Diversamente, invece, è l'esempio di destra perchè secondo quanto detto fin ora, un nodo è valido solo se tutti i valori nella abstract store sono hi (vedi la Definizione di validità) dunque il nodo $l1$ non è valido in quanto non soddisfa tale definizione. Nonostante ciò, la presenza dell'$\&_\exists()$ gli consente di trasmettere un valore qualitativamente buono. Tale valore di taintness proviene esclusivamente da S_2 essendo l'unico sensore con un valore di qualità hi. Inoltre essendo $l1$ non critico il sistema non si ferma, si fermerebbe solamente se diventasse critico $l3$. Dunque tramite il tag di qualità è possibile riuscire a mettere in atto un meccanismo di fault tolerance rispetto ad eventi di tampering dei sensori.