

Réunion choix

22/11/2022

Travail sur le projet d'Adwan Abdallah

Ivan BAHEUX

Semaine 47/48

Travail Abdallah

Après une bataille sans relâche :

```
statemachine ABSTRACT_SM {  
    state START {  
        transition on APP_STARTED -> ENTER_PHRASE_ACTIVITY  
    }  
  
    super state ENTER_PHRASE_ACTIVITY abstracts ENTER_PHRASE_ACTIVITY_SM {  
        transition on TRANSLATION_SUCCEEDED -> VIEW_TRANSLATION_ACTIVITY  
        transition on TERMINATE_BUTTON_CLICKED -> EXIT  
    }  
  
    super state VIEW_TRANSLATION_ACTIVITY abstracts VIEW_TRANSLATION_ACTIVITY_SM {  
        transition on TERMINATE_BUTTON_CLICKED -> EXIT  
        transition on BACK_BUTTON_CLICKED -> ENTER_PHRASE_ACTIVITY  
    }  
  
    state EXIT  
}
```

Exemple TranslationApp (une partie)

Le code est fonctionnel !

```
ABSTRACT_SM  
state -> START  
event -> APP_STARTED  
state -> ENTER_PHRASE_ACTIVITY  
event -> TERMINATE_BUTTON_CLICKED  
state -> EXIT
```

```
state -> START  
event -> APP_STARTED  
state -> ENTER_PHRASE_ACTIVITY  
event -> TRANSLATION_SUCCEEDED  
state -> VIEW_TRANSLATION_ACTIVITY  
event -> TERMINATE_BUTTON_CLICKED  
state -> EXIT
```

```
state -> START  
event -> APP_STARTED  
state -> ENTER_PHRASE_ACTIVITY  
event -> TRANSLATION_SUCCEEDED  
state -> VIEW_TRANSLATION_ACTIVITY  
event -> BACK_BUTTON_CLICKED  
state -> ENTER_PHRASE_ACTIVITY  
event -> TRANSLATION_SUCCEEDED  
state -> VIEW_TRANSLATION_ACTIVITY  
event -> TERMINATE_BUTTON_CLICKED  
state -> EXIT
```

Exemple des paths générées pour l'ABSTRACT_SM de TranslationApp

Ce qu'il manque

- La génération des chemins complets
- Les algorithmes de couverture de code (Peut-être fonctionnels)
 - Pas vérifiés et pas commentés
- Une documentation
- La génération des chemins à partir de tous les points d'entrée
 - Main de l'app + exported components
- Bien sûr ... ma partie

Ce qu'il manque de mon côté

- Mise au propre du git (pour postérité)
- Mise en lien avec Vpat + Enrichissement des sink+source
- Algorithme de recherche de pattern
- Génération de test de sécurité en langage réel

A faire : Gantt

VPAT : Vulnerability PATtern (Appellation à définir)

```
Vulnerability "private_to_okhttp.send()"
{
  description "An example of vulnerability : Outputting private sources to HTTP"
  function
  {
    PrivateSource ANY_PRIVATE,
    Sink "okhttp.send()" {
      parameter {
        isFunction
        Variable a {
          origin ANY_PRIVATE
        }
      }
    }
  }
  context {
    Context PERMISSION_INTERNET,
    Context PERMISSION_ANY_PRIVATE
  }
}
```

Objectif :

- AND, OR, XOR in context and origin
- Check isFunction
- Mots clés
ANY_PRIVATE,
ANY_PUBLIC
- Lecture de la machine à état par le dsl « compilé »

Travail à venir

Prévisionnel pour la suite :

Mettre au clair un git propre pour coder (pour l'instant une backup du code propre en version ZIP à été mise sur gricad)

Tester ses algorithmes + completer

Documenter sa partie