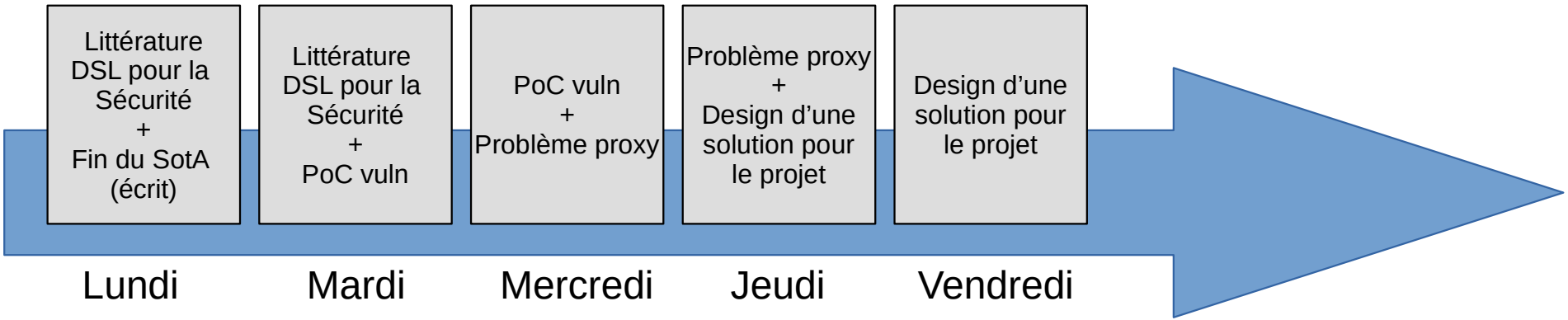
Réunion hébdomadaire 6 — 21/10/2022

Overview de la semaine



Ivan BAHEUX — *Semaine 42*

Travail de la semaine : s 42

State of the art

PDF produit

Changements :

- Fin de la partie introduction
- Ajout d'une section *adversary*
- Fix de quelques erreurs
- Ajout partie MBST et DSL pour la sécurité

DSL utilisation pour la sécurité :

- Plusieurs solutions éxistent, le DSL est un OUTIL
  - DSL définissant un modèle (https://www.researchgate.net/publication/259725434 : A DSL for Web Services Automatic Test Data Generation)

```
AST ::= BasicType | ComplexType | {Tag, Attributes, AST}

BasicType ::= {Tag, Attributes, Content}
  where Tag = empty | int | string | ..
    and Content = Literal | gen

ComplexType ::= {Tag, Attributes, Content}
  where Tag = sequence | union | list
    and Content = [AST]
```

  - DSL définissant un langage d'attaque (https://dl.acm.org/doi/10.1145/3465481.3470070 : An Agnostic Domain Specific Language for Implementing Attacks in an Automotive Use Case)

Listing 1: Automatic detection of unknown variables with ALIA

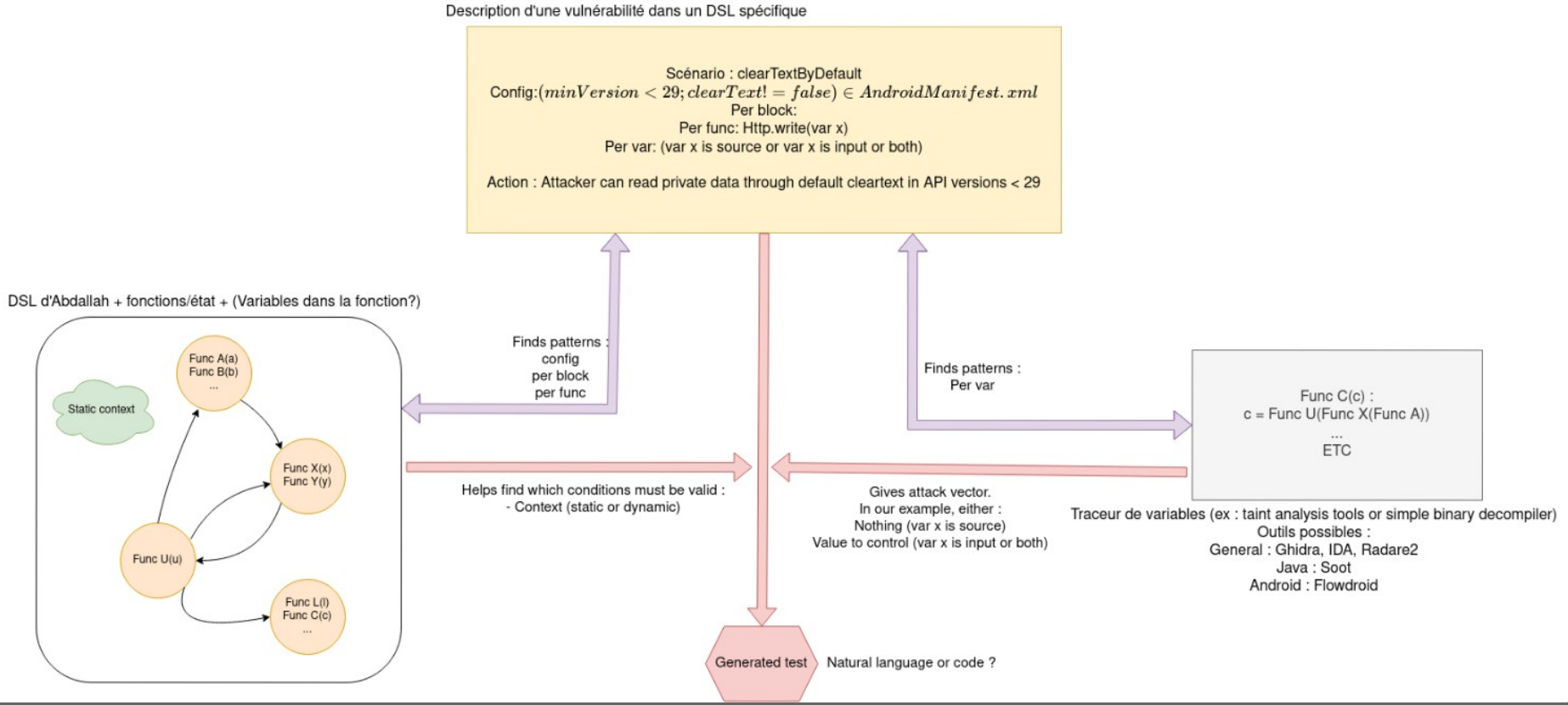
```
PreConditions:
  get_su_rights: con
Actions:
  get_con: con = exploit(type:OpenADB, target:
      ip_addr)
  get_su_rights: exploit(type:ScriptExecution,
      command:'su')
  exe_whoami: user = exploit(type:
      ScriptExecution, command:'whoami')
  list: exploit(type:ScriptExecution, command:
      'ls')
PostConditions:
  exe_whoami: uesr == "root"
```

Vulnérabilités intéressantes

- Vulnérabilités en lien avec la backward compatibility des API Android :
  - Exemple :
    - Avant API level 28 (~1/4 des appareils)
      - Configuration par défaut accepte HTTP
        → Man-in-the-Middle

- Intérêt :
  - Génération de tests par version prouvant la sécurité à toute version
  - Lié au contexte de l'appareil et semble implémentable avec le DSL de A.Adwan (après amélioration)

Idée de design pour notre cas :



Design : Critiques et choix

Critiques :

- Complexe (plusieurs outils)
- Briques non finies (Tout n’est pas déjà automatisé, donc il faudra ajouter cela dans la charge de travail)
  - DSL pas automatisé
  - Dans les cas complexes : difficile à implementer ?
  - DSL ne représente pas l’interconnection des apps android
- Travail à faire sur la sélection d’outils

Avantages :

- Très généralisable (Probablement à des patterns très divers)

Travail à venir

Prévisionnel pour la suite :

Faire fonctionner le proxy pour un PoC

Tester des outils