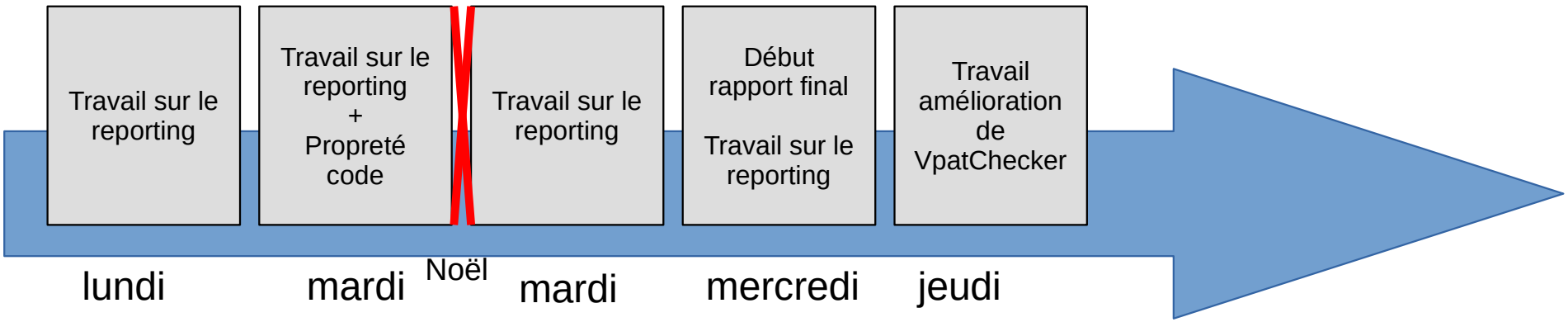
Réunion hebdomadaire 13 05/01/2023

Bonne année !

Overview de la semaine



Ivan BAHEUX *Semaine 52/01*

Reporting : La dernière fois

Exemple de rapports positifs :
- Vulnérabilité « log.d leak »

```
Report is positive for Vulnerability [["Log.d Leak"]
    {"Log.d kept in code makes it vulnerable to leakage of data"}
    NO STATICMODEL IMPLEMENTATION FOR NOW
    Sink -> SinkModel [name=log_d, parameters:
---  SourceModel [name=private, AnyValue]]
] - TestPath {fr.castav.lcis.Checker.model.TestPaths$TestPath@4417af13}

Report is positive for Vulnerability [["Log.d Leak"]
    {"Log.d kept in code makes it vulnerable to leakage of data"}
    NO STATICMODEL IMPLEMENTATION FOR NOW
    Sink -> SinkModel [name=log_d, parameters:
---  SourceModel [name=private, AnyValue]]
] - TestPath {fr.castav.lcis.Checker.model.TestPaths$TestPath@67594471}
```

Reporting : Maintenant

```
------------ Debug information for ABSTRACT_SM ------------
 =====> vulnerableFunc has :
  Total tested 35
  Negatives    5
  Positives    30
  Positive -> <OUTPUT_FOLDER>/vulnerableFunc/ABSTRACT_SM/vulnerableFunc_0
  Positive -> <OUTPUT_FOLDER>/vulnerableFunc/ABSTRACT_SM/vulnerableFunc_1
  Positive -> <OUTPUT_FOLDER>/vulnerableFunc/ABSTRACT_SM/vulnerableFunc_2
  Positive -> <OUTPUT_FOLDER>/vulnerableFunc/ABSTRACT_SM/vulnerableFunc_3
  Positive -> <OUTPUT_FOLDER>/vulnerableFunc/ABSTRACT_SM/vulnerableFunc_4
  Positive -> <OUTPUT_FOLDER>/vulnerableFunc/ABSTRACT_SM/vulnerableFunc_5
  Positive -> <OUTPUT_FOLDER>/vulnerableFunc/ABSTRACT_SM/vulnerableFunc_7
  Positive -> <OUTPUT_FOLDER>/vulnerableFunc/ABSTRACT_SM/vulnerableFunc_8
  Positive -> <OUTPUT_FOLDER>/vulnerableFunc/ABSTRACT_SM/vulnerableFunc_10
  Positive -> <OUTPUT_FOLDER>/vulnerableFunc/ABSTRACT_SM/vulnerableFunc_11
  Positive -> <OUTPUT_FOLDER>/vulnerableFunc/ABSTRACT_SM/vulnerableFunc_12
  Positive -> <OUTPUT_FOLDER>/vulnerableFunc/ABSTRACT_SM/vulnerableFunc_13
  Positive -> <OUTPUT_FOLDER>/vulnerableFunc/ABSTRACT_SM/vulnerableFunc_14
  Positive -> <OUTPUT_FOLDER>/vulnerableFunc/ABSTRACT_SM/vulnerableFunc_16
  Positive -> <OUTPUT_FOLDER>/vulnerableFunc/ABSTRACT_SM/vulnerableFunc_17
  Positive -> <OUTPUT_FOLDER>/vulnerableFunc/ABSTRACT_SM/vulnerableFunc_18
  Positive -> <OUTPUT_FOLDER>/vulnerableFunc/ABSTRACT_SM/vulnerableFunc_19
  Positive -> <OUTPUT_FOLDER>/vulnerableFunc/ABSTRACT_SM/vulnerableFunc_20
  Positive -> <OUTPUT_FOLDER>/vulnerableFunc/ABSTRACT_SM/vulnerableFunc_21
  Positive -> <OUTPUT_FOLDER>/vulnerableFunc/ABSTRACT_SM/vulnerableFunc_22
  Positive -> <OUTPUT_FOLDER>/vulnerableFunc/ABSTRACT_SM/vulnerableFunc_23
  Positive -> <OUTPUT_FOLDER>/vulnerableFunc/ABSTRACT_SM/vulnerableFunc_24
  Positive -> <OUTPUT_FOLDER>/vulnerableFunc/ABSTRACT_SM/vulnerableFunc_25
  Positive -> <OUTPUT_FOLDER>/vulnerableFunc/ABSTRACT_SM/vulnerableFunc_26
  Positive -> <OUTPUT_FOLDER>/vulnerableFunc/ABSTRACT_SM/vulnerableFunc_28
  Positive -> <OUTPUT_FOLDER>/vulnerableFunc/ABSTRACT_SM/vulnerableFunc_29
  Positive -> <OUTPUT_FOLDER>/vulnerableFunc/ABSTRACT_SM/vulnerableFunc_31
  Positive -> <OUTPUT_FOLDER>/vulnerableFunc/ABSTRACT_SM/vulnerableFunc_32
  Positive -> <OUTPUT_FOLDER>/vulnerableFunc/ABSTRACT_SM/vulnerableFunc_33
  Positive -> <OUTPUT_FOLDER>/vulnerableFunc/ABSTRACT_SM/vulnerableFunc_34
```

```
=====> Log.d_Leak has :
 Total tested 35
 Negatives    25
 Positives    10
 Positive -> <OUTPUT_FOLDER>/Log.d_Leak/ABSTRACT_SM/Log.d_Leak_2
 Positive -> <OUTPUT_FOLDER>/Log.d_Leak/ABSTRACT_SM/Log.d_Leak_6
 Positive -> <OUTPUT_FOLDER>/Log.d_Leak/ABSTRACT_SM/Log.d_Leak_8
 Positive -> <OUTPUT_FOLDER>/Log.d_Leak/ABSTRACT_SM/Log.d_Leak_9
 Positive -> <OUTPUT_FOLDER>/Log.d_Leak/ABSTRACT_SM/Log.d_Leak_11
 Positive -> <OUTPUT_FOLDER>/Log.d_Leak/ABSTRACT_SM/Log.d_Leak_15
 Positive -> <OUTPUT_FOLDER>/Log.d_Leak/ABSTRACT_SM/Log.d_Leak_18
 Positive -> <OUTPUT_FOLDER>/Log.d_Leak/ABSTRACT_SM/Log.d_Leak_22
 Positive -> <OUTPUT_FOLDER>/Log.d_Leak/ABSTRACT_SM/Log.d_Leak_27
 Positive -> <OUTPUT_FOLDER>/Log.d_Leak/ABSTRACT_SM/Log.d_Leak_30
-------------------------------------------------------------
```

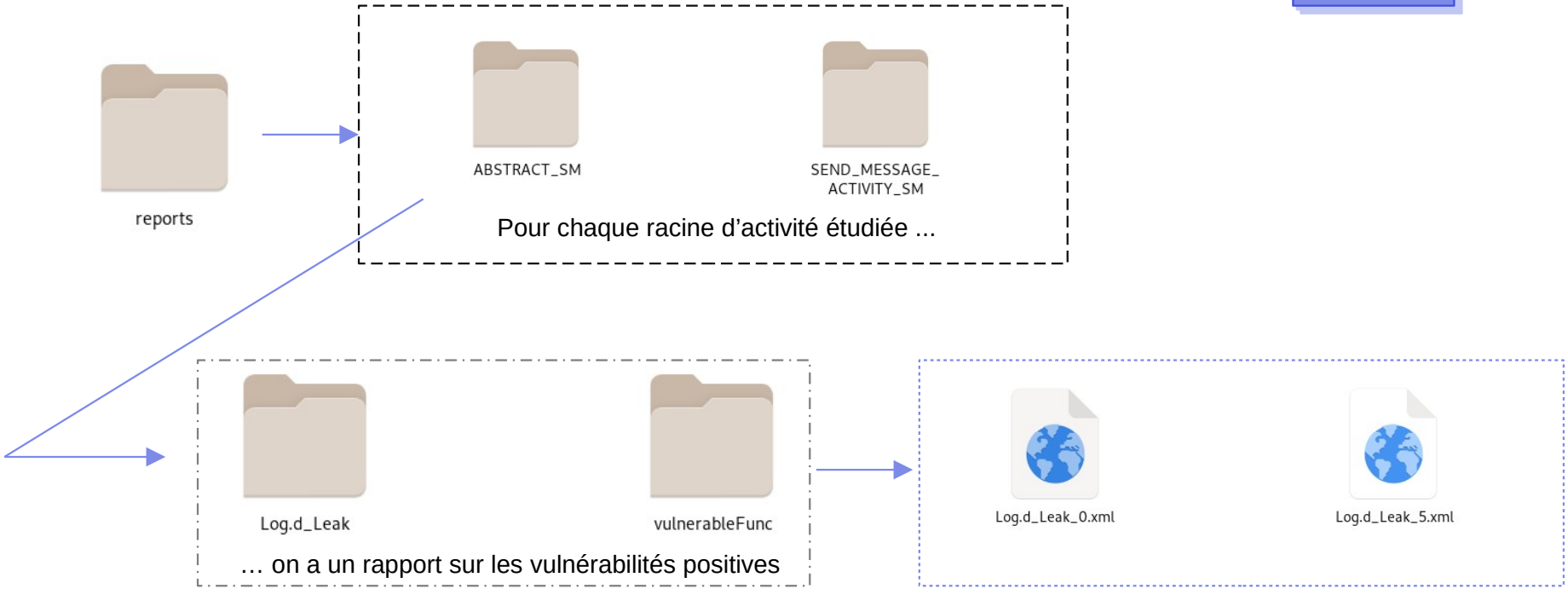
Ceci est l'affichage dans la console lorsqu'on active le debug

Informations :

- Par « racine » :

Nombre de positifs et négatifs à une vulnérabilité

Reporting : Maintenant



Rapport final : plan

Chapter 3

Methods

3.1 Vulnerability Focus

3.2 Design Methodology
3.2.1 Base work
3.2.2 Design goals
3.3 Data collection and testing methodology

Chapter 4

Contribution

4.1 Global Architecture

4.2 Test Generation

4.2.1 ConTest - Architecture

4.2.2 ConTest - Implementation

4.2.2.1 Context DSL

4.2.2.2 Behaviour DSL

4.2.2.3 Generator

4.3 Vulnerability detection

4.3.1 VPAT - Architecture

4.3.2 VPAT - Implementation

4.3.2.1 Vulnerability patterns

4.3.2.2 Vulnerability detection

Chapter 5

Results and analysis

Chapter 6

Discussions

Chapter 7

Conclusions and future work

Travail à venir

Prévisionnel pour la suite :

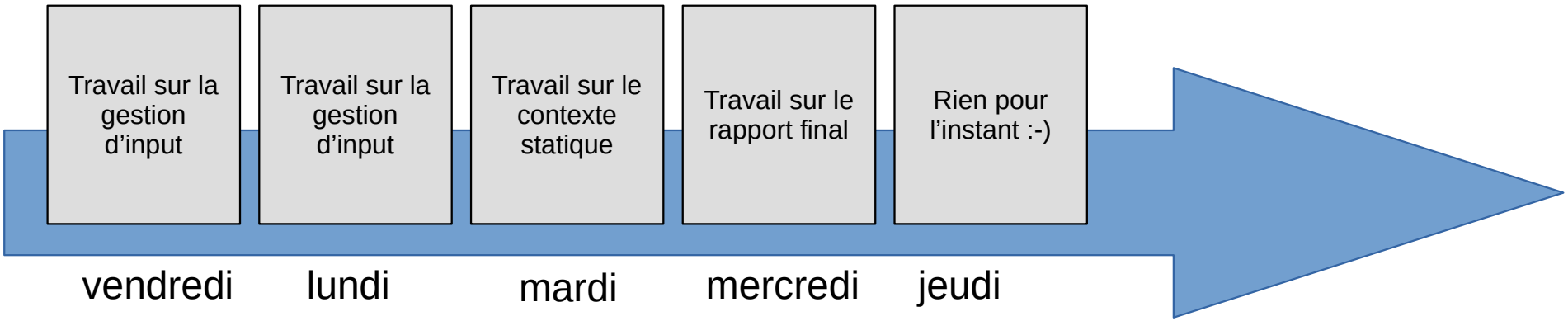
Avancer le rapport final

Travail sur le tri / la classification des patterns
Fin du travail sur l'input
Faire plus de tests
Ajouter le contexte statique : Version android (minsdk, target…)
Configuration réseau
→ peut se faire en ayant juste l'accès au manifest
→ « trivial » mais prend du temps

Réunion hebdomadaire 14 12/01/2023

Bonne année !

Overview de la semaine



Ivan BAHEUX *Semaine 01/02*

Gestion des inputs

Ce qu'il manquait :

- Jusqu'ici les inputs n'étaient jamais modifiés dans le test
  i.e s'il y avait un input dans le programme, il ne se passait rien dans le test

Ce qu'on à fait :

- Les inputs peuvent être utilisés lorsqu'une fonction nécessite une valeur spécifique (Cas le plus courant)

ex :

```
Vulnerability "log.t API 26" {
    description "log.t can be exploited by giving his first argument 'OUTPUsT'"

    function {
        main Sink "log.t" {
            parameter {
                private,
                static "OUTPUsT"
            }
        },
        Source private *
    }
}
```

Divulgâchage

Fig 1 : Vulnérabilité log.t API 26

```
<state name="SENDER">
    <transition name="WRONG"/>
    <dataflows>
        <dataflow name="inputFunction" type="Input" value="OUTPUsT"/>
    </dataflows>
</state>
```

Fig 2 : État d'un test positif à la vulnérabilité ci-contre

Contexte statique

Afin d'ajouter le contexte statique :

- Mise-à-jour de CBML pour suivre le design original (générateur de tests)

```
model TranslationApp {

    contexts {
        INTERNET_CONNECTIVITY
    }

    static contexts {
        minSdk = "26",
        maxSdk = "",
        targetSdk = "32"
    }

    situations {
        INTERNET_DISCONNECTED : INTERNET_CONNECTIVITY,
        INTERNET_SLOW : INTERNET_CONNECTIVITY
    }
```

Ceci génère les tests normaux et puis multiplie les tests pour chaque contexte statique.
i.e : Pour chaque test on crée le même test dans le contexte APIVersion 26, 27, 28… 32

Avantage pour le générateur tester plusieurs versions, et pour VpatChecker on a une information de plus

Contexte statique

Limites :

- Pour l'instant seul trois contextes statiques sont implémentés (pour VPat car c'est générique dans CBML) dont deux de façon incomplète :

Fig 3 : Définition de VPat

```
Context returns Context:
    Permission?='android.permission.' value=Permission | Network?='network' value=Network | Version?="apiversion" value=Version
;

Version returns Version:
    name=STRING
;

Permission returns Permissions:
    {Permission} name=permissionID
;

//TODO : ADD every permission in android
permissionID returns PermissionID:
    name='ACCESS_MEDIA_LOCATION' |
    name='ACCESS_NETWORK_STATE'  |
    name='ACCESS_WIFI_STATE'     |
    name='INTERNET'
;

//TODO : ADD network configurations (Trusted CA...)
Network returns Network:
    {Network} 'default'
;
```

Avancement rapport final

Chapitre 3 bien avancé (Methods)

Résumé :

- Vulnerability focus
- Design methodology
- Data collection and testing methodology (à paufiner)

Travail à venir

Prévisionnel pour la suite :

Avancer le rapport final

Travail sur le tri / la classification des patterns
Faire plus de tests
Ajouter le contexte statique : Configuration réseau