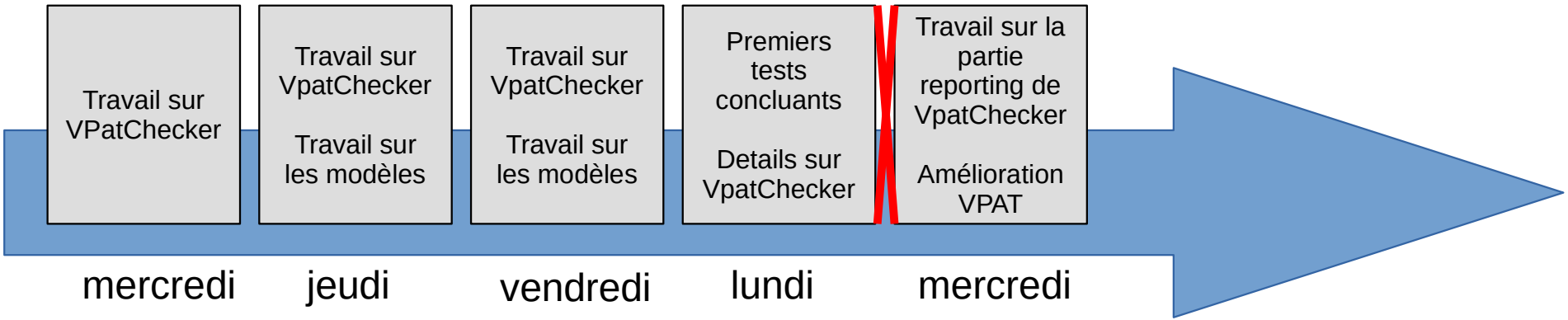
Réunion hebdomadaire 12 15/12/2022

Overview de la semaine

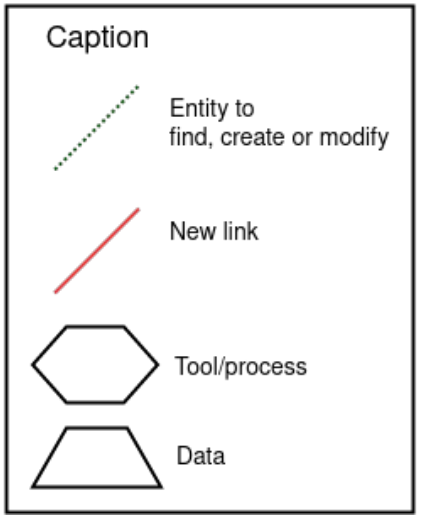


Ivan BAHEUX *Semaine 49*

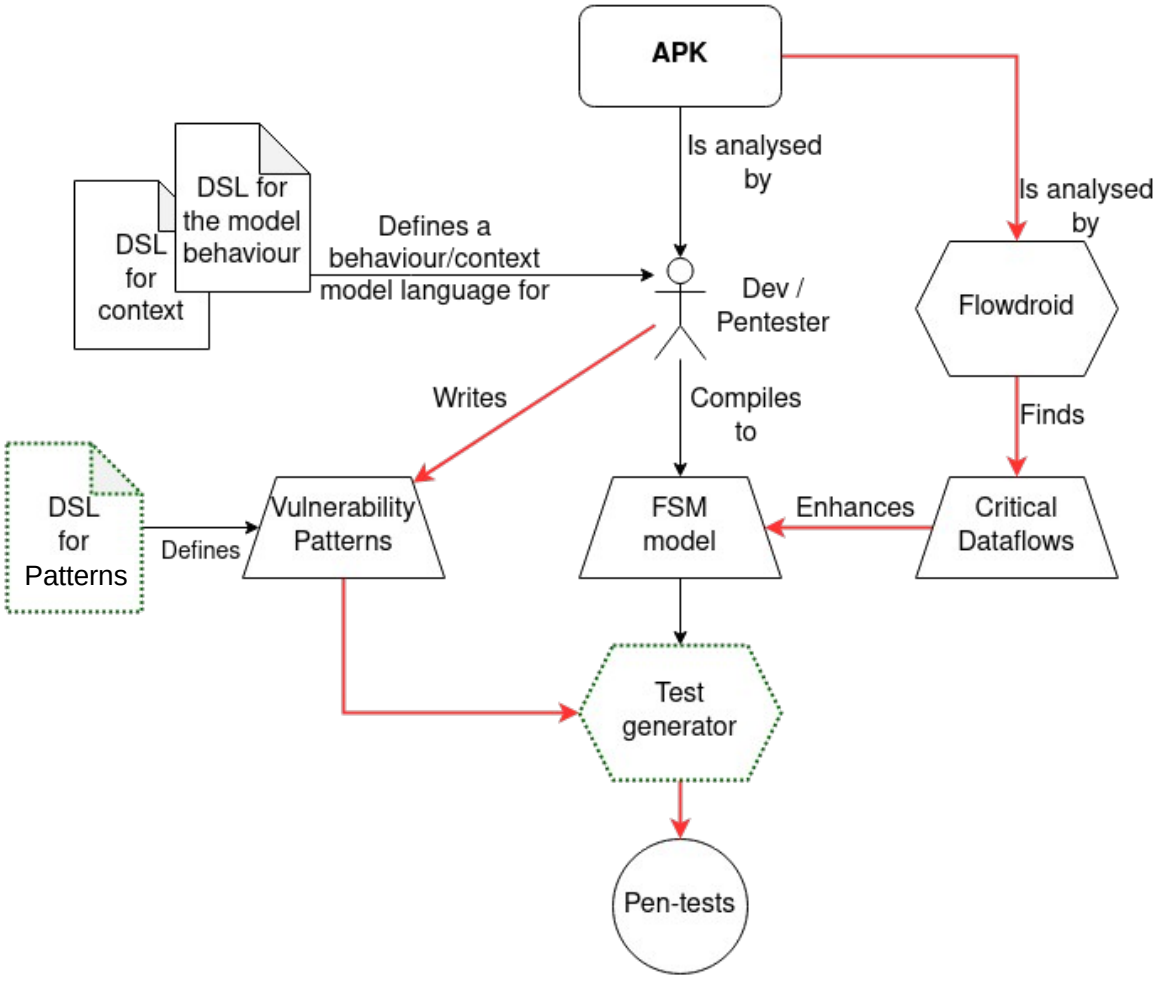
Rappel

But :

- Permettre à un développeur de tester puis de démontrer la présence des vulnérabilités définies par les patterns en fonction du contexte de l'application



White box idea



Vpat : pour l'instant

- Update :
  - Un pattern est défini par une fonction principale « main »

Exemple simpliste

```
Vulnerability "Log.d Leak" {
    description "Log.d kept in code makes it vulnerable to leakage of data"

    function {
        main Sink "log.d" {
            parameter {
                private
            }
        } ,
        Source private *
    }
}
```

Le pattern est valide quand :

Une source privée est redirigée vers le paramètre de la méthode « log.d »

Vpat : à venir

- Définir la classification dans le langage :
  - Trier des vulnérabilités générales par CWE
  - Trier des vulnérabilités spécifiques par CVE
- Définir dans le langage comment exploiter la vulnérabilité

(Un autre exemple pour la route..)

```
Vulnerability "vulnerableFunc" {
    description "vulnerableFunc can be exploited by giving his first argument 'evilPayload'"

    function {
        main Sink "vulnerableFunc" {
            parameter {
                static "evilPayload"
            }
        }
    }
}
```

Vpat : à venir

- Prendre en compte des motifs contextuels
  - Concerne Vpat et VpatChecker
- Petit objectif personnel :
  - Possiblement héberger les motifs de façon open-source suite à l’idée de Pascal André

Vpat Checker

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}
```

Vpat Checker

Le testpath vulnérable en question (reconstruit par vpatchecker) :

Explication :

State Start

→ AppStarted

State Send_message

HAS SOURCE INTERNET

→ NONE (context = slow3G)

State Handle_slow_internet

Sinks source from « SendMessage »

…

...

```
<TestPath>
    <state name="START">
        <transition name="APP_STARTED"/>
    </state>
    <state name="SEND_MESSAGE">
        <transition name="NONE">
            <contexts>
                <context origin="INTERNET_CONNECTIVITY">slow3G</context>
            </contexts>
            <situations>
                <situation origin="DEFAULT_ORIGIN">INTERNET_SLOW</situation>
            </situations>
        </transition>
        <dataflows>
            <dataflow name="internet" type="Source"/>
        </dataflows>
    </state>
    <state name="HANDLE_SLOW_INTERNET">
        <transition name="NONE"/>
        <dataflows>
            <dataflow name="log_d" type="Sink">
                <parameters>
                    <parameter origin="source">internet</parameter>
                </parameters>
            </dataflow>
            <dataflow name="log_t" type="Sink">
                <parameters>
                    <parameter origin="source">internet</parameter>
                    <parameter origin="value">OUTPUT</parameter>
                </parameters>
            </dataflow>
        </dataflows>
    </state>
    <state name="SENDER">
        <transition name="WRONG"/>
        <dataflows>
            <dataflow name="enter_value" type="Input"/>
        </dataflows>
    </state>
    <state name="DISPLAY_WARNING">
        <transition name="BACK_BUTTON_PRESSED"/>
    </state>
    <state name="SEND_MESSAGE">
        <transition name="SEND_MESSAGE_CLICKED"/>
        <dataflows>
            <dataflow name="internet" type="Source"/>
        </dataflows>
    </state>
```

Travail à venir

Prévisionnel pour la suite :

Faire des tests

Travailler sur un meilleur reporting.

Ajouter une information « oubliée » dans le modèle CBML
→ « nom de l'activité » (rien de compliqué)