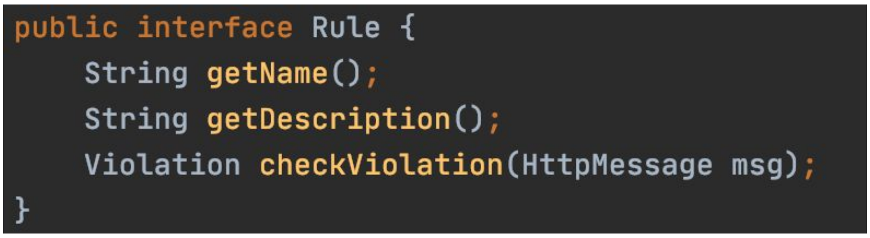
Guide for developing rules

This guide explains how to write extension rules for the ReportingProxy ZAP add on.

1 - Writing rules:

To write new rules, the developer implements the Rule interface.

The rule interface is defined as follows:

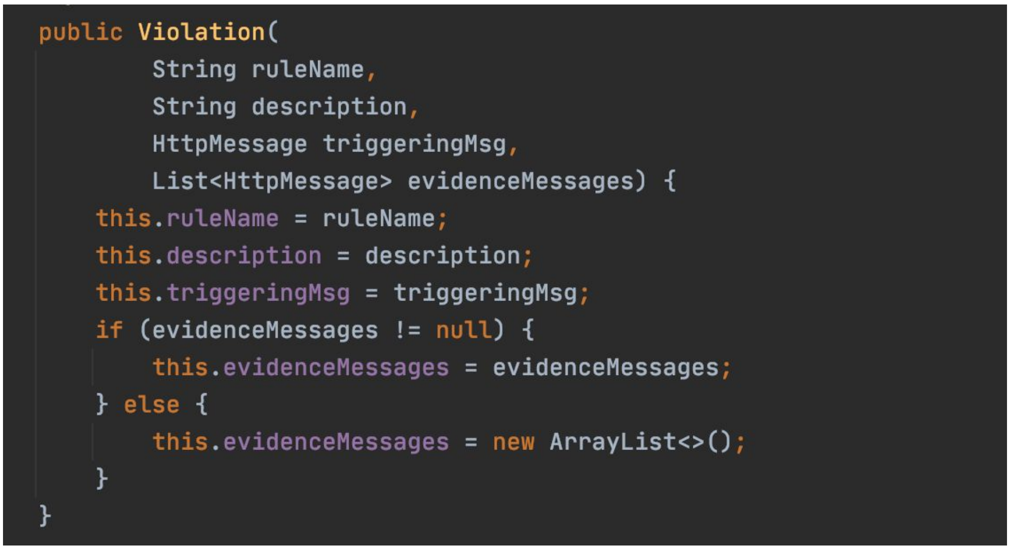


```
public interface Rule {
    String getName();
    String getDescription();
    Violation checkViolation(HttpMessage msg);
}
```

● *String getName()* : The name of your rule. This name is used to represent the rule in the report and in alerts.

● *String getDescription()* : A description for the rule.

● *Violation checkViolation(HttpMessage msg)* : A violation test that returns a Violation object if there is violation of rule. If there is no violation, returns null. The method takes an HttpMessage as defined in org.parosproxy.paros.network.HttpMessage. We note that an HttpMessage contains both the request and response messages, it is up to the developer to enforce his rule on either or both of them.



```
public Violation(
        String ruleName,
        String description,
        HttpMessage triggeringMsg,
        List<HttpMessage> evidenceMessages) {
    this.ruleName = ruleName;
    this.description = description;
    this.triggeringMsg = triggeringMsg;
    if (evidenceMessages != null) {
        this.evidenceMessages = evidenceMessages;
    } else {
        this.evidenceMessages = new ArrayList<>();
    }
}
```

- *ruleName* : the rules name.
- *description* : the rules description.
- *triggeringMsg* : The HTTP message that triggered the violation alert.
- *evidenceMessages*: The HTTP message that contributed to the rule being violated.

2 - Compiling rules:

1- create a working space directory X and go to it

2- create the following hierarchy of directories :
org/zaproxy/zap/extension/reportingproxy/rules

3- Place your rules in X/org/zaproxy/zap/extension/reportingproxy/rules/ .

4 - Run the following commands:

```
javac -cp <ZAP.jar> <UserStory3Extension.jar> <rules_dir>/*.java
jar cMf rules.jar org/zaproxy/zap/extension/reportingproxy/rules/*.class
```

5- The created JAR file rules.jar contains the rules to be loaded in ZAP.

Where :

- <ZAP.jar> the ZAP program in a JAR format, in our case this file can be found in "Group17/zaproxy/zap/build/libs/"
- <UserStory3Extension.jar> : The JAR of our compiled user story. This file can be found in zap-addons/addOns/reportingproxy/build/libs/reportingproxy-1.jar after having compiled ran *zap-addons/gradlew clean build*

We note that we have provided a bash script to automatically compile our rules. The script is located in solutions/userstory3/build_rules.sh

3 - Loading policies to ZAP :
The developer is able to load his policies to ZAP by going to tools -> Policy Loader and selecting the .jar files of his policies. He should be presented with an alert message notifying him of the success (or failure) of the operation.