Policy Verifier Extension

This extension lets a developer provide a JAR file containing the programmatic implementation of a set of policy rules. When running in proxy mode, ZAP will check requests and responses against these rules. If a rule flags a request, an alert is raised. This last can be consulted by the developer from within the ZAP UI, in the « Alerts » tab.

How to implement a Policy

Pre-requisites:

1) JAR of ZAP. To create a JAR of ZAP in IntelliJ IDEA:
    1) Surf into the « zaproxy » directory
    2) Click « File », « Project Structure », « Artifacts », « + » , « Jar », « From module with dependencies »
    3) In the new window, select the « zaproxy.zap.main » module, and « ZAP » as Main Class, then click « OK » and « Apply »
    4) Once the Artifact has been defined, click on the menu « Build », « Build Artifacts… » and build the previously defined artifact.
    5) The JAR can now be found in the directory « out/artifacts »
2) JAR of the extension: Follows the steps explained before, however for the gradle project located in « Group19/zap-extensions ». **Note**: As Module you have to select « zap-extensions.addOns.policyverifier.main », and no Main Class needs to be selected.
3) Build the extension in order to obtain the .zap loadable extension file:
    1) In the same project as in prerequisite 2, compile the extension with the following command « ./gradlew clean && ./gradlew :addOns:policyverifier:build »
    2) You will find the extension having path « Group19/zap-extensions/addOns/policyverifier/build/zapAddOn/bin/policyverifier-alpha-1.zap »

Implementing Rules

A policy is defined as the Jar file containing the programmatic implementation of a set of policy rules. In practice, to create a Policy, you first need to implement a set of rules (each one in a separate java class) starting from an empty Java project, then create a Jar of those rules.

STEP-BY-STEP TUTORIAL

1. Create a new Java empty project. Often, default packages are created by IDEs such as IntelliJ IDEA. Therefore, make sure to delete everything from the « src » directory.
2. Before implementing your rules, you have to add the JAR of ZAP and the one of the extension as dependencies to your new project, otherwise you will not be able to compile because some packages and classes will be missing.
   In IntelliJ Idea:
   1. Click on « File » -> « Project Structure » -> « Modules »
   2. In the « Dependency » tab, click « + » in the bottom of the window, then « JARs or directories » and select the JAR of ZAP, and the one of the extension.
3. Every Rule you write must follow a strict implementation design and structure. From a programmatic perspective this implies implementing the interface rule as defined below.

```java
/** A Rule is an entity which can be valid or not according to the checked HttpMessage */
public interface Rule {
    default String getName() {
        return this.getClass().getSimpleName();
    }

    /**
     * The implementation of this method embeds the algorithm to check if the rule is valid or not.
     *
     * @param msg HttpMessage to check against validity
     * @return Boolean telling if the Http message follows this rule
     */
    boolean isValid(HttpMessage msg);
}
```

4. Now you can start coding your rules which implements the « Rule » interface. Note that:
   1. The name of the rule that will be eventually displayed in the ZAP UI is the name of the class, therefore make sure to use an explanatory name.
   2. The most important part of a rule is the method « isValid » which returns true if the HttpMessage received as argument follows the rule, false otherwise.
5. Creating the JAR of your policy:
   1. In IntelliJ IDEA, you can create an artifact, however you need to make sure that ZAP or the extension are excluded from the output. You can delete everything in the « Output Layout » tab of the menu « Artifact », by selecting everything except « [NAME OF PROJ] compile output, and clicking « - » .
   2. Build the artifact and you will obtain the JAR representing your policy.

How to load a Policy

STEP-BY-STEP TUTORIAL

1) Launch ZAP
2) To import the extension: Click on « File », then « Load Add-on File » and select the previously built extension (it can be found in « Group19/zap-extensions/addOns/policyverifier/build/zapAddOn/bin/policyverifier-alpha-1.zap »). The UI will now show an additional menu called « Policy Verifier ».
3) To load a policy: Click on « Policy verifier », « Load new policy », select the JAR representing your policy. If something goes wrong during the loading phase, an error will be displayed.

   Note that, if you load a policy with the same name as one previously loaded, the extension will update the oldest one by replacing the oldest one with the new one.

What happens if policies are violated ?

Once you have loaded a policy, zap proxy will constantly check if traffic follows your policy rules. If some rule is violated, you will find an alert under « JARPolicyVerifier » in the « Alerts » tab. If clicked the alert, represented by the response or request that has been violated, will display different information, one of which is the « Description : ». The format of this last is « PolicyX.RuleY violated ».