Usage guide for policy rule matching DSL

The DSL allows developers to easily write policies.
Policies are defined by are set of rules, each of which is responsible of verifying a predicate on the incoming/outgoing messages. The policy name is the one of the loaded txt file.

The DSL's syntax is defined as follows :

Defining a rule:
Rule "<name>" "<description>" :
<composed_predicate> ;

- <name>: the name of the rule
- <description>: the description of the rule
- <composed_predicate>: one or multiple simple predicates linked by operators {and | or | not}.

Defining a simple predicate:

<transmission_type>.<field_type>.<mode_of_operation> = <matcher>

- <transmission_type> : "request" or "response".
- <field_type> : "header" or "body" .
- <mode_of_operation> : "re" for regex, "value" for a single literal value, "values" for a list of values.
- <matcher> : represents the value to match, for "re" and "value" mode matcher is represented as a string. For "values" mode, matcher is a list of strings.

Note on composed predicates:
Composed predicates are a logical expression of simple predicates. Parenthesis are supported as well as the and, or , not operators.

Example policy mypolicy.txt :

Rule "keyword_AND_frenzy_rule" "response body contains both of the keywords":
(response.body.value="hacker" and response.body.value="zerohedge") or
(response.body.value="cern");

Rule "keyword_list_rule" "response body contains at least one of the keywords in the list":
response.body.values=["hacker","zerohedge"];