Application Boundaries Enforcer (ABE) NoScript Module

Rules Syntax And Capabilities

Version 1.2.2 – 2017-07-27

Author: Giorgio Maone – giorgio@maone.net

Table of Contents

1. Ruleset, Rules and Predicates....................2
   1.1. Actions....................3
   1.2. Methods....................4
   1.3. Resources....................5
2. Examples....................6
3. EBNF Grammar Reference....................7

1. Ruleset, Rules and Predicates

This is a functional description of the ABE rules capabilities. For a formal syntactic specification please see *3. EBNF Grammar Reference*.

ABE rules are meant to enforce specific *actions* modifying HTTP *request*s identified by their destination *site* (an URI pattern), and optionally by their HTTP *method* and/or their origin site.

Rules are collected in a ruleset, which is just a text file containing a list of rules and optionally comments:

```
<rule>
[[#comment]
[<rule>]
...]
```

Comments are ignored: they can be placed anywhere, start with a '#' character and always finish at the end of the line.

Rule priority goes down from top to bottom: processing stops as soon as a rule matches current request. Therefore highest priority (most specific) rules should be placed topmost.

A rule looks like this:

```
Site <resource>
<predicate>
[<predicate>
...]
```

A *<resource>* is typically an URI pattern (literal, glob or regexp) designing a request destination (site), or a wildcard token such as ALL or SELF.

A *<predicate>* is made of an *<action>* (e.g. Allow, Deny) which must be enforced on certain requests. Requests are identified by their HTTP *<method>*s (e.g. GET, POST or ALL) and optionally by their origin, specified as "*from <resource>*" :

```
<action> [<method>[ <method>...]] [from <resource>]
```

A missing *<method>* or *from* clause implies *ALL* or *from ALL*, respectively.

Predicate priority goes down from top to bottom: processing stops as soon as a predicate matches current request. Therefore most specific predicates (e.g. "Accept GET from somesite.com") should be placed topmost, while most general ones (e.g. "DENY from ALL" or just "DENY") should go on the bottom.

1.1. Actions

Available ABE <*predicate*> actions are:

- **Accept –** lets the request pass through as it is
- **Sandbox** – sends the requests as it is, but disables JavaScript and other active content (e.g. plugin embeddings) in the landing page
- **Anonymize** *(synonyms are **Anon** and **Logout**)* – strips Authorization and Cookie headers, turns methods different than GET, HEAD and OPTIONS into GET, remove upload data, then sends the modified request.
- **Deny** – completely blocks the request, preventing it from being sent

1.2. Methods

The <*method*> component of a <*predicate*> can be any HTTP method (GET, POST, HEAD, PUT, DELETE, TRACE, OPTIONS) with the addition of 3 "pseudo" methods:

- **ALL** – the <*action*> of this <*predicate*> must be enforced independently from the HTTP method of the requests (i.e. for all methods)
- **SUB** – the <*action*> of this <*predicate*> must be enforced only if this is a subdocument request, i.e. if the requested resource is going to be shown in a frame or iframe
- **INCLUSION** (alias **INC**) – the <*action*> of this <*predicate*> must be enforced only if this is an inclusion sub-request (i.e. not a top-level load). The inclusion type(s) to match can be listed as optional comma-separated arguments inside parentheses, e.g. `INCLUSION(SCRIPT, OBJ)`.
  If no type is specified, this pseudo-method matches any sub-request.
  Valid types are **SCRIPT**, **CSS**, **IMAGE**, **OBJ** (plugin objects and sub-requests from plugin objects), **OBJSUB** (just sub-request from plugin objects), **MEDIA**, **FONT**, **SUBDOC** (subdocuments, i.e. documents loaded in frames and iframes), **XBL**, **PING**, **XHR** (XMLHttpRequest and Fetch loads), **DTD**, **OTHER** and **UNKNOWN. Starting with NoScript version 5.0.8** all the "external" types defined in Mozilla's nsIContentPolicy.TYPE_* constants are dynamically supported: the name of the ABE type is the same as the constant's, but without the TYPE_ prefix. **UNKNOWN** matches any request not mapped yet to a specific nsIContentPolicy.TYPE_ (like TYPE_OTHER), while **OTHER**, for historical/compatibility reasons, matches the same requests as UNKNOWN plus any TYPE_* (like TYPE_**WEBSOCKET** or TYPE_**CSP_REPORT**) not matched by the original "static" ABE types.

1.3. Resources

A <*resource*> is an URI pattern (a literal string designating a full URI, a domain, a glob, a regular expression or a special token) which is matched to designate the destination site(s) which a rule applies to, and optionally the origin of the requests which a certain <*predicate*> refers to.

A resource can be expressed the following ways:

- **ALL**
  special token matching any URI (it can be omitted)
- **^https?://some\.site\.com/.***
  regular expression
- ***.some.site.com** (matches anything.some.site.com but *not* some.site.com) or
  .some.site.com (matches anything.some.site.com *and* some.site.com)
  glob expression; a glob expression starting with "." will match both the subdomains having it as a suffix and the domain stripped of the leading dot.
- **www.some.site.com**
  domain literal
- **http://www.somesite.com**
  URI literal with "starts with" matching
- **LOCAL**
  special token for local network (private IPv4 and IPv6, see RFC 3330 and RFC 4193)
- **SELF**
  special origin-only token for strict prepath (scheme://auth@host:port) matching
- **SELF+**
  special origin-only token for host matching
- **SELF++**
  special origin-only token for base (2nd level) domain matching

2. Examples

```
# This is a Ruleset example, made of comments (like this line)
# and rules, like the ones following.

# This one defines normal application behavior, allowing hyperlinking
# but not cross-site POST requests altering app status
# Additionally, pages can be embedded as subdocuments only by documents from
# the same domain (this prevents ClickJacking/UI redressing attacks)
Site *.somesite.com
Accept POST SUB from SELF https://secure.somesite.com
Accept GET
Deny

# This one guards logout, which is foolish enough to accept GET and
# therefore we need to guard against trivial CSRF (e.g. <img>)
Site www.somesite.com/logout
Accept GET POST from SELF
Deny

# This one guards the local network, like LocalRodeo
# LOCAL is a placeholder which matches all the LAN
# subnets (possibly configurable) and localhost
Site LOCAL
Accept from LOCAL
Deny

# This one strips off any authentication data
# (Auth and Cookie headers) from requests outside the
# application domains, like RequestRodeo
Site *.webapp.net
Accept ALL from *.webapp.net
Anonymize

# This one allows Facebook scripts and objects to be included only
# from Facebook pages
Site .facebook.com .fbcdn.net
Accept from .facebook.com .fbcdn.net
Deny INCLUSION(SCRIPT, OBJ, SUBDOC)
```

3. EBNF Grammar Reference

Follows the grammar of an ABE ruleset, expressed in EBNF notation:

```
tokens {
  T_ACTION;
  T_METHODS;
}

ruleset   : rule* EOF ;
rule      : subject predicate+ -> subject predicate+ ;
predicate : action methods? origin? -> T_ACTION action T_METHODS methods?
origin? ;
methods   : (method+ | ALL) ;
method    : (HTTPVERB | SUB | inclusion) ;
inclusion : INC (LPAR (INC_TYPE COMMA)* INC_TYPE? RPAR)? ;
origin    : T_FROM oresources ;
subject   : T_SITE resources ;
oresources: (oresource+ | ALL) ;
resources : (resource+ | ALL) ;
oresource: resource | 'SELF' | 'SELF+' | 'SELF++' ;
resource  : REGEXP | GLOB | URI | LOCATION ;
action    : A_DENY | A_LOGOUT | A_SANDBOX  | A_ACCEPT ;

T_SITE    : 'Site' ;
T_FROM    : ('f' | 'F') 'rom' ;
A_DENY    : 'Deny' ;
A_LOGOUT  : 'Logout' | 'Anon' 'ymize'? ;
A_SANDBOX : 'Sandbox' ;
A_ACCEPT  : 'Accept' ;

fragment URI_START : 'a'..'z' | '0'..'9' ;
fragment URI_PART  : 'a'..'z' | 'A'..'Z' | '0'..'9' | '_' | '-' | '.' |
        '[' | ']' | ':' | '/' | '@' | '~' | ';' | ',' |
        '?' | '&' | '=' | '%' | '#' ;
LOCATION  : 'LOCAL' ;
URI       : URI_START URI_PART+ ;
GLOB      : (URI_START | '*' | '.') (URI_PART | '*')* ;
REGEXP    : '^' ~'\n'+ ;

ALL       : 'ALL' ;
SUB       : 'SUB' ;
INC       : 'INC' 'LUSION'? ;
HTTPVERB  : 'GET' | 'POST' | 'PUT' | 'PATCH' | 'DELETE';
INC_TYPE  : 'A'..'Z' ('A'..'Z' | 'A'..'Z' '_' 'A'..'Z')+ ;

COMMA     : ',' ;
LPAR      : '(' ;
RPAR      : ')' ;

WS        :  (' '|'\r'|'\t'|'\u000C'|'\n') {$channel=HIDDEN;} ;
COMMENT : '#' ~'\n'* {$channel=HIDDEN;} ;
```