

Application Boundaries Enforcer (ABE) NoScript Module

Rules Syntax And Capabilities

Version 0.5 – 2009-02-02

Author: Giorgio Maone – g.maone@informaction.com

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 *requests* 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 only at the beginning of a line, start with a '#' character and 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
- **Logout** – strips Authorization and Cookie headers from the request, then sends it
- **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 two “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

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
- **^https?://some\.site\.com/.*** – regular expression
- ***.some.site.com** – glob expression
- **www.some.site.com** – domain literal
- **http://www.somesite.com** – URI literal
- **LOCAL** – special token for configurable local network
- **SELF** – special token for strict domain match

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 SUBDOC 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
Logout
```

3. EBNF Grammar Reference

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

```
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)
;
origin       : T_FROM resources
;
subject      : T_SITE resources
;
resources    : (resource+ | ALL)
;
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' ;
A_SANDBOX  : 'Sandbox' ;
A_ACCEPT   : 'Accept' ;

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

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