Security Policy

https://campus.barracuda.com/doc/45026285/

A Security Policy determines what action to take when one or more of the rules match the request. All security policies are global and can be shared among multiple Services configured on the Barracuda Web Application Firewall.

To Create a Security Policy

URL: /v1/security_policies			
Method: POST			
Description: Creates a security policy with the default values.			
Parameter Name	**Data Type**	**Mandatory**	**Description**
Input Parameters:			
name	Alphanumeric	Yes	The name of the security policy that needs to be created.

Example:

Request:

curl http://192.168.0.1:8000/restapi/v1/security_policies -u 'eyJldCI6IjEzODAxNDg0NjgiLCJwYXNzd29yZCI6ImFjNGEzNDJmNjAzNTBhNWE3MTgxNjQ4Nzll\nOGJhMGY3IiwidXNlciI6ImFkbWluIn0=\n:' -X POST -H Content-Type:application/json -d '{"name":"new_policy"}'

Response:

{"id":"new_policy","token":"eyJldCI6IjEzODAxNDg5OTgiLCJwYXNzd29yZCI6IjJjMGRhZTk4ZGQ1ZWJjMzQ0ZTA4ZWY3NGNk\nNDczNmZkIiwidXNlciI6ImFkbWluIn0=\n"}

To Retrieve Security Policies

URL: /v1/security_policies /v1/security_policies/{policy_id}			
Method: GET			
Description: Lists all security policies if “policy_id” is not specified.			
Parameter Name	**Data Type**	**Mandatory**	**Description**

Input Parameters:			
parameters	Alphanumeric	Optional	Any specific parameter name that needs to be retrieved. See *Example 2*.

Example 1:

Request:

curl http://192.168.0.1:8000/restapi/v1/security_policies/new_policy -u 'eyJldCI6IjEzODAxNDg0NjgiLCJwYXNzd29yZCI6ImFjNGEzNDJmNjAzNTBhNWE3MTgxNjQ4Nzll\nOGJhMGY3IiwidXNlciI6ImFkbWluIn0=\n:' -X GET

Response:

{"default_character_set":"UTF-8","cloaking":{"suppress_return_code":"1","headers_to_filter":["Server","X-Powered-By","X-AspNet-Version"],"return_codes_to_exempt":[],"filter_response_header":"1"},"apply_double_decoding":"No","data_theft_protection":["credit-cards","ssn","directory-indexing"],"url_protection_status":"1","allowed_acls":2,"request_limits":{"max_number_of_headers":"20","enable":"1","max_header_name_length":"32","max_cookie_name_length":"64","max_query_length":"4096","max_cookie_value_length":"4096","max_request_length":"32768","max_header_value_length":"512","max_url_length":"4096","max_request_line_length":"4096","max_number_of_cookies":"40"},"parameter_protection":{"enable":"1","denied_metacharacters":"%00 %04%1b%08%7f","file_upload_extensions":["JPG","GIF","PDF"],"maximum_upload_file_size":"1024","blocked_attack_types":null,"ignore_parameters":["__VIEWSTATE"],"custom_blocked_attack_types":[],"allowed_file_upload_type":"extensions","maximum_parameter_value_length":"1000","maximum_instances":null,"file_upload_mime_types":["image/jpeg","image/gif","application/pdf"],"exception_patterns":[]},"id":"new_policy","token":"eyJldCI6IjEzODAxNDkxNzMiLCJwYXNzd29yZCI6IjgxMTg5ZGYzODU3YjcwOTA0ODEwYTJkODZl\nZTU3MzRjIiwidXNlciI6ImFkbWluIn0=\n","url_protection":{"enable":"1","maximum_parameter_name_length":"64","max_content_length":"32768","max_parameters":"40","allowed_content_types":["application/x-www-form-urlencoded","multipart/form-data","text/xml"],"maximum_upload_files":"5","blocked_attack_types":null,"custom_blocked_attack_types":[],"csrf_prevention":"none","allowed_methods":["GET","POST","HEAD"],"exception_patterns":[]},"cookie_security":{"secure_cookie":"0","cookies_exempted":["__utma","__utmc","__utmz","__utmb","AuthSuccessURL","CTSESSION","SMSESSION","SMCHALLENGE"],"cookie_max_age":"1440","cookie_replay_protection_type":"IP","http_only":"0","days_allowed":"7","tamper_proof_mode":"signed","custom_headers":[],"allow_unrecognized_cookies":"custom"},"url_normalization":{"parameter_separators":"ampersand","default_charset":"UTF-8","double_decoding":"No","detect_response_charset":"0"},"cookie_protection":"signed","limit_checks":"1","name":"new_policy","parameter_protection_status":"1","disallowed_acls":7}

Example 2:

Request:

curl http://192.168.0.1:8000/restapi/v1/security_policies/new_policy -u 'eyJldCI6IjE1MDE5MDUxMzkiLCJwYXNzd29yZCI6IjUwN2I1ZDRhMTc3Mzc4Zjc5NGY2ZmM3NTNh\nYTczM2IxIiwidXNlciI6ImFkbWluIn0=\n:' -X GET -G -d parameters=cookie_security,cloaking

Response:

{"cookie_security":{"secure_cookie":"yes","cookies_exempted":["__utma","__utmc","__utmz","__utmb","AuthSuccessURL","CTSESSION","SMSESSION","SMCHALLENGIE"],"cookie_max_age":"50000","cookie_replay_protection_type":"none","http_only":"yes","days_allowed":"Never","tamper_proof_mode":"encrypted","custom_headers":["host","Cookie","User-Agent"],"allow_unrecognized_cookies":"never"},"cloaking":{"suppress_return_code":"yes","headers_to_filter":["Server","date"],"return_codes_to_exempt":["403"],"filter_response_header":"yes"},"id":"new_policy","token":"eyJldCI6IjE1MDQzMTY0MzciLCJwYXNzd29yZCI6IjMwZDhlZTU1MjRmMzY4MDEyMDlmODI2Yzc1\nZmU3Y2ZlIiwidXNlciI6ImFkbWluIn0=\n"}

To Update a Security Policy

In this REST API call, the parameters can be passed in a Simple JSON request or a Nested JSON request based on the parameters that needs to be modified. For information on JSON requests, see **Request Syntax**.

URL: /v1/security_policies/{policy_id}			
Method: PUT			
Description: Updates a security policy with the given values.			
Parameter Name	**Data Type**	**Mandatory**	**Description**
Input Parameters:			
request_limits. enable	String	Optional	Enforce size limit checks on request headers or not. The values include: • yes • no
request_limits. max_request_length	Numeric	Optional	The maximum allowable request length. This includes the Request Line and all HTTP request headers (for example, User Agent, Cookies, Referer etc.).
request_limits.max_request_line_length	Numeric	Optional	The maximum allowable length for the request line. The request line consists of the method, the URL (including any query strings) and the HTTP version.

request_limits.max_url_length	Numeric	Optional	The maximum allowable URL length including the query string portion of the URL.
request_limits.max_query_length	Numeric	Optional	The maximum allowable length for the query string portion of the URL.
request_limits.max_number_of_cookies	Numeric	Optional	The maximum number of cookies to be allowed.
request_limits.max_cookie_name_length	Numeric	Optional	The maximum allowable length for a cookie name.
request_limits.max_cookie_value_length	Numeric	Optional	The maximum allowable length for a cookie value.
request_limits.max_number_of_headers	Numeric	Optional	The maximum number of headers to be allowed in a request.
request_limits.max_header_name_length	Numeric	Optional	The maximum allowable length for a header name.
request_limits.max_header_value_length	Numeric	Optional	The maximum allowable length for header value in a request.
cookie_security.tamper_proof_mode	Enumeration	Optional	The tamper proof mode for cookies. The enumerated values include: • signed • encrypted • none
cookie_security.cookie_max_age	Numeric	Optional	The maximum age for session cookies.
cookie_security.cookie_replay_protection_type	Enumeration	Optional	The type of protection to be used to prevent the cookie replay attacks. The enumerated values include: • none • IP • IP_and_custom_headers • custom_headers
cookie_security.custom_headers	Alphanumeric	Optional	The custom headers to be used in the cookie if the parameter "Cookie Replay Protection Type" is set to "Custom Headers" or "IP and Custom Headers".
cookie_security.secure_cookie	String	Optional	Determines whether to allow or not the cookies if the client makes secure HTTPS connection. The values include: • yes • no
cookie_security.http_only	String	Optional	Determines whether or not the cookie security feature will be enabled for HTTP cookies. The values include; • yes • no
cookie_security.allow_unrecognized_cookies	Enumeration	Optional	Determines whether unrecognized cookies should be allowed. The enumerated values include: • custom • always • never

cookie_security.days_allowed	Numeric	Optional	The number of days the Barracuda Web Application Firewall to not reject unrecognized cookies.
cookie_security.cookies_exempted	Alphanumeric	Optional	The names of the cookies that needs to be exempted from the cookie security policy.
url_protection.enable	String	Optional	Determines whether to enforce URL protection or not. The values include: • yes • no
url_protection.allowed_methods	Alphanumeric	Optional	The list of allowable methods in a request.
url_protection.allowed_content_types	String	Optional	The list of content types to be allowed in the POST body of a request.
url_protection.max_content_length	Numeric	Optional	The maximum content length to be allowed for POST request body.
url_protection.max_parameters	Numeric	Optional	The maximum number of parameters to be allowed in a request.
url_protection.maximum_upload_files	Numeric	Optional	The maximum number of files that can be of file-upload type in a request.
url_protection.csrf_prevention	Enumeration	Optional	The Cross-Site Request Forgery (CSRF) prevention for the forms and URLs. The enumerated values include: • forms_and_urls • none • forms
url_protection.maximum_parameter_name_length	Numeric	Optional	The maximum length of a parameter name in a request.
url_protection.blocked_attack_types	Enumeration	Optional	The Attack Types to be matched in a request. The enumerated values include: • cross_site_scripting • remote_file_inclusion • sql_injection_strict • sql_injection • os_command_injection • remote_file_inclusion_strict • os_command_injection_strict • cross_site_scripting_strict
url_protection.custom_blocked_attack_types	Enumeration	Optional	The custom attack types defined on the **ADVANCED > Libraries** page (if any).
url_protection.exception_patterns	String	Optional	The patterns to be allowed despite matching a malicious pattern group. **Note**: Configure the exact "Pattern Name" displayed on the **ADVANCED > View Internal Patterns** page, or as defined when creating a "New Group" on the **ADVANCED > Libraries** page.

parameter_protection.enable	String	Optional	Determines whether to enforce parameter protection or not. The values include: • yes • no
parameter_protection.denied_metacharacters	String	Optional	The meta-characters to be denied in the parameter value. Meta-characters must be URL encoded. Non-printable characters such as "backspace" and web interface reserved characters like "?" should be URL encoded.
parameter_protection.maximum_parameter_value_length	Numeric	Optional	The maximum allowed length of any parameter value, including no-name parameters.
parameter_protection.maximum_instances	Numeric	Optional	The maximum number of times a parameter needs to be allowed in a request.
parameter_protection.base64_decode_parameter_value	String	Optional	Determines whether to apply base64 decoding to the parameter values or not. The values include: • yes • no **Note**: If the parameter value adheres to the Data URI Scheme, the base64 decoding is applied on the parameter value irrespective of **base64_decode_parameter_value** is set to *yes* or *no*. If not, the base64 decoding is applied to the parameter value only when **base64_decode_parameter_value** is set to *yes*.
parameter_protection.allowed_file_upload_type	Enumeration	Optional	The allowed file upload types. The enumerated values include: • extensions • mime_types
parameter_protection.file_upload_extensions	Alphanumeric	Optional	The extensions to be allowed as uploaded files.
parameter_protection.file_upload_mime_types	Alphanumeric	Optional	The Mime types to be allowed as uploaded files.
parameter_protection.maximum_upload_file_size	Numeric	Optional	The maximum size (in KB) for an individual file that can be uploaded in a request.
parameter_protection.blocked_attack_types	Enumeration	Optional	The Attack Types to be matched in a request. The enumerated values include: • directory_traversal • directory_traversal_strict • cross_site_scripting • remote_file_inclusion • sql_injection_strict • sql_injection • os_command_injection • remote_file_inclusion_strict • os_command_injection_strict • cross_site_scripting_strict

parameter_protection.custom_blocked_attack_types	Enumeration	Optional	The custom attack types defined on the **ADVANCED > Libraries** page (if any).
parameter_protection.exception_patterns	String	Optional	The patterns to be allowed despite matching a malicious pattern group. **Note**: Configure the exact "Pattern Name" displayed on the **ADVANCED > View Internal Patterns** page, or as defined when creating a "New Group" on the **ADVANCED > Libraries** page.
parameter_protection.ignore_parameters	Alphanumeric	Optional	The parameters to be exempted from *all* validations.
cloaking.suppress_return_code	String	Optional	Suppress an HTTP Status code in the response header and insert a default or custom response page in case of any error responses from the server. The value includes: • yes • no
cloaking.return_codes_to_exempt	String	Optional	The HTTP response codes that needs to be exempted from cloaking.
cloaking.filter_response_header	String	Optional	Remove the HTTP headers in responses. The values include: • yes • no
cloaking.headers_to_filter	String	Optional	The list of headers that are to be removed from a response before serving it to a client.
url_normalization.default_charset	Enumeration	Optional	The character set decoding type to be used for incoming requests. The enumerated values include: • GBK • ASCII • Shift-JIS • ISO-8859-1 • JOHAB • EUC-KR • EUC-JP • ISO-2022-KR • ISO-2022-CN • UTF-8 • HZ • BIG5 • GB2312 • EUC-TW • ISO-2022-JP
url_normalization.detect_response_charset	String	Optional	Determines whether or not the Barracuda Web Application Firewall will detect the character set decoding from the response. The values include: • yes • no

url_normalization.parameter_separators	Enumeration	Optional	The url-decoded parameter separator to be used. The enumerated values include: • ampersand • ampersand_and_semicolon • semicolon
url_normalization.apply_double_decoding	String	Optional	Determines whether or not to apply double-decoding of the character set. The values include: • yes • no

Example 1:

Request:

curl http://192.168.0.1:8000/restapi/v1/security_policies/new_policy -u 'eyJldCI6IjEzODAxNDg0NjgiLCJwYXNzd29yZCI6ImFjNGEzNDJmNjAzNTBhNWE3MTgxNjQ4Nzll\nOGJhMGY3IiwidXNlciI6ImFkbWluIn0=\n:' -X PUT -H Content-Type:application/json -d '{"cookie_security":{"cookie_replay_protection_type":"none","allow_unrecognized_cookies":"never","tamper_proof_mode":"encrypted"}}'

Response:

{"msg":"Configuration Updated","token":"eyJldCI6IjEzODAxNDkzMjAiLCJwYXNzd29yZCI6ImMwNTEyNzA3ZTM1NmI3ZmMyNTBkYjFhOGI4\nM2ZhOTg0IiwidXNlciI6ImFkbWluIn0=\n"}

Example 2:

Request:

curl http://192.168.0.1:8000/restapi/v1/security_policies/new_policy -u 'eyJldCI6IjE1MDE4NDAxMTciLCJwYXNzd29yZCI6IjdhNDQyN2I1ODAxMGM2MTBiYWM5NGRiNGVj\nNTY3ZDFlIiwidXNlciI6ImFkbWluIn0=\n:' -X PUT -H Content-Type:application/json -d '{"cookie_security":{"cookie_replay_protection_type":"none","allow_unrecognized_cookies":"never","tamper_proof_mode":"encrypted"},"url_protection":{"enable":"no","max_content_length":"0","max_parameters":"0","maximum_upload_files":"100","maximum_parameter_name_length":"100","allowed_methods":["GET","POST"]},"parameter_protection":{"enable":"yes","denied_metacharacters":"%00%04%1b%08%7f%23%50","exception_patterns":["sql-quote","unsafe-tag"],"file_upload_mime_types":["text/html","image/jpeg","image/gif"]},"cloaking":{"return_codes_to_exempt":["403"],"filter_response_header":"yes","headers_to_filter":["Server","date"]}}'

Response:

{"msg":"Configuration Updated","token":"eyJldCI6IjE1MDQyNDc2OTciLCJwYXNzd29yZCI6IjA2MjVhMDViMzJjNTg1NDRlMDBlZDUxNTFh\nZGI1MGQ0IiwidXNlciI6ImFkbWluIn0=\n"}

To Delete a Security Policy

URL: /v1/security_policies/{policy_id}
Method: DELETE
Description: Deletes the given security policy.

Example:

Request:

curl http://192.168.0.1:8000/restapi/v1/security_policies/new_policy -u 'eyJldCI6IjEzODAxNDg0NjgiLCJwYXNzd29yZCI6ImFjNGEzNDJmNjAzNTBhNWE3MTgxNjQ4Nzll\nOGJhMGY3IiwidXNlciI6ImFkbWluIn0=\n:' -X DELETE

Response:

{"msg":"Successfully deleted","token":"eyJldCI6IjEzODAxNDk0MTAiLCJwYXNzd29yZCI6Ijg0MDQ5MTkyYzhhZjMwY2YxMzM5M2M5NTdi\nMGVmNDJmIiwidXNlciI6ImFkbWluIn0=\n"}