Configuring Website Profiles

https://campus.barracuda.com/doc/17629538/

Overview

The intricate structure of an application is called a profile of the website. Website profiles are made up of profiles for URLs and profiles for parameters of those URLs. A URL profile lists allowed fields like HTTP methods, names and types of each parameter, query strings, length based restrictions, etc. A Parameter profile defines the allowed format for each parameter using either a negative or positive security model and includes length restrictions.

Website Profiles allow you to create specific rules to fine tune the security settings of a Service. They do not modify the default security policy settings, but fine tune security settings specific to a Service. For a Service, a Website Profile is applied if **Use Profile** is set to *Yes*, meaning the request must be validated against configured URL and Parameter profiles of that Service. Initially no URL and Parameter Profiles exist for a Service. To use Website Profiles, the administrator must manually create URL and Parameter profiles for the Service.

When a Service is added on the **BASIC > Services** page, a website profile is created and Use Profile is set to "*Yes*" for the Service. To modify the default settings for a Service, perform the following steps:

1. Go to the **SECURITY > Website Profiles** page.
2. In the **Service** section, select the Service from the **Website** drop-down list whose settings you want to modify.
3. Click the **Edit** button. The **Edit Website Profile** window appears. Specify values for the following fields if required:
    1. **Use Profile –** Set to *Yes* to use URL profiles and parameter profiles for validating the requests coming for this Service.
    2. **Strict Profile –** Set to *Yes* to enforce strict profile checks thereby denying requests which do not match any profile. If set to "*No*", then the Service's default web firewall policy will be applied to those requests which do not have a profile.
    3. **Mode** – Set the mode for the service:
        1. **Passive** – Validates the requests against the URL Profiles and Parameter Profiles settings and logs request errors/violations on the **BASIC > Web Firewall Logs** page.
        2. **Active** – Validates the requests against the URL Profiles and Parameter Profiles settings, blocks request violations and logs the corresponding violations on the **BASIC > Web Firewall Logs** page.
    4. **Allowed Domains –** Enter the domain or IP address of the Service whose requests/responses should be validated against the URL and Parameter Profiles. If you wish to allow multiple sub domains under a main domain, then you can configure it as

domain=maindomain. For example, "**world.com**" might have pages at "**india.world.com**," "**america.world.com**," and "**japan.world.com**." By default, if a web page on "**india.world.com**" is configured under **Allowed Domains**, only pages on "**india.world.com**" are allowed. If the user wants all subdomains in the "**world.com**" domain to be allowed, then specify "**domain=world.com**".

5. **Exclude URL Patterns** – Enter the list of URL patterns to be excluded from the URL Profile validations. These URLs are exempted from learning even if the Learning is On. Examples: *.html,*.htm,*.jpg, *.gif,*.css,*.js
6. **Include URL Patterns** – Enter the list of URL patterns to be included in the URL profile validations in spite of being listed in **Exclude URL Patterns**.

4. Click **Save Changes** to save the settings.

URL Profiles

URL Profiles are validated against the requests for the Service based on the **Mode** setting of the URL profile.

How to Add a URL Profile

1. Go to the **SECURITY > Website Profiles** page.
2. In the **Service** section, select the Service from the **Website** drop-down list to which you want to add a URL profile.
3. In the **URL Profiles** section, click **Add URL**. The **Create URL Profile** window appears. Specify values for the following fields:
    1. **URL Profile Name** – Enter a name for the URL profile.
    2. **Status** – Set to *On* if you want to enforce checks on requests/responses for the Service using this profile.
    3. **URL** – Enter a URL to be compared to the URL in the request. The URL should start with a "/" and can have at most one " * " anywhere in the URL. The value of “/*” means all URLs in the Service are matched against the URL in the request.
    4. **Extended Match** – Specify an expression, a combination of HTTP headers and/or query string parameters, you want used to match the special attributes in the HTTP headers or query string parameters in the requests. Use '*' to denote "any request", that is, do not apply the Extended Match condition. For information on how to write extended match expression, see ***Extended Match Syntax***.
    5. **Extended Match Sequence** – Enter a number to indicate the order in which the extended match rule will be evaluated in for requests.
    6. **Mode** – Set the mode for this URL profile.
        1. **Passive –** Validates the requests comparing them to the URL profile and corresponding Parameter profile(s) settings and logging request errors/violations on the **BASIC > Web Firewall Logs** page.
        2. **Active –** Validates the requests comparing them to the URL profile and corresponding Parameter profile(s) settings, blocking request violations and logging

the corresponding violation on the **BASIC > Web Firewall Logs** page.

7. **Allow Query String** – Set to *Yes* to allow parameters and its values along with the URL.
8. **Hidden Parameter Protection** – Specify whether or not to protect hidden parameters in the forms and URLs.
    1. **Forms** – Protects the hidden parameters in the post body of forms.
    2. **Forms and URLs** – Protects the hidden parameters in the post body of forms and query string of the URLs.
    3. **None** – No protection to hidden parameters in forms and URLs.
9. **CSRF Prevention** – Specify whether or not to prevent cross-site request forgery attack on the forms and URLs.
10. **Max Content Length** – Enter the maximum content length to be allowed for POST request body.
11. **Maximum Parameter Name Length** – Enter the maximum length of the parameter name. The allowed length is 1 to 1024 bytes. No value (empty) implies unlimited.
12. **Maximum Upload Files** – Enter the maximum number of files that can be uploaded in one request. If the value is set to two (2), then the third (3) file upload is denied. The Passive mode logs every uploaded file that exceeds the max count.
13. **Blocked Attack Types** – By default, all attack types are selected. Attack Types are specifications of malicious patterns. If the value of a parameter matches one of the specified Attack Types, an intrusion is detected and logged on the **BASIC > Web Firewall Logs** page. Attack Types are defined with groups of Regular expression patterns. Attack Types for SQL Injection, Cross Site scripting and System Command Injection attacks are provided by default, and one or more of these can be enabled for matching against request parameters.
14. **Custom Blocked Attack Types** – By default, all custom attack types are selected. Clear the checkbox to allow any of the patterns.

4. Click **Save** to add the URL profile.
5. Click **Edit** next to the created URL profile to specify values for the following fields:
    1. Allowed Methods – Enter the methods to be allowed in the request. The Barracuda Load Balancer ADC uses this to decide whether to allow or disallow the methods.
    2. Allowed Content Types – Enter the content types to be allowed for this URL profile.
    3. Referrers for the URL Profile – Enter the address (URI) of the resource from which the Request URI was obtained. In case of adaptive profiling, the referrers are learned as the profile sources. This referrer is not same as the “Referrer” in CSRF protection. **Note**: This is used only for information purpose, and no security checks are enforced by the Barracuda Load Balancer.
    4. Exception Patterns – Enter the patterns to be allowed as exceptions even if part of a malicious pattern group. The configuration should be the exact "Pattern Name" as found on the **SECURITY > View Internal Patterns** page, or as defined during the creation of a "New Group" through the **SECURITY > Libraries** page. The pattern name can also be found in a Web firewall log when a false positive occurs due to a potential exception pattern. For example, if the parameter value matched "sql-comments" regex pattern under "sql-injection medium" attacks on the **SECURITY > View Internal Patterns** page, then adding "sql-comments" to this list will allow "sql-comments" in future.
6. Click **Save Changes** to save the above settings.

Parameter Profiles

Parameter profiles are compared to the requests for the Service based on the **Mode** setting of the corresponding URL profile.

How to Add a Parameter Profile

1. Go to the **SECURITY > Website Profiles** page.
2. In the **Service** section, select the Service from the **Website** drop-down list.
3. In the **URL Profiles** section, select the desired URL profile where you want to add the Parameter profile.
4. Click **Add Param** in the Parameter Profiles section. The **Create Parameter Profile** window appears. Specify values for the following fields:
    1. **Parameter Profile Name** – Enter a name for the parameter.
    2. **Status** – Set to *On* to validate the requests coming to the Service using this Parameter Profile.
    3. **Parameter** – Enter the name of the parameter to be validated in requests/responses. The parameter names with the special characters like &pathinfo and &sessionid and wildcard (*) should be manually specified, they are not learned automatically.
    4. **Type** – Select the type of parameter to be validated in requests/responses.

        If two or more parameters of different type have the same name, then parameters would be considered as **Input** type and be bound to one of standard parameter classes and the value of the parameter **Max Instances** would be updated. The types of parameters.

        1. *Input* – The parameter other than File Upload, Global Choice, Read Only, Session Choice, and Session Invariant type is treated as **Input** type.
        2. *Read Only* – All hidden parameters in the form and query parameters in the URL is learned as *Read Only* type. If an exception occurs while learning, then the type is updated to Input. This type makes the parameter session specific.
        3. *Session Choice* – The parameter from a response form and the drop-down list is different across different sessions or same session, then it is treated as *Session Choice*.
        4. *Global Choice* – The input type parameters like check boxes, radio buttons and menu parameters in a form is treated as *Global Choice* type.
        5. *Session Invariant* – Select this if the parameter value is same across multiple requests from the same session, then it can be set as *Session Invariant*, for example; session-id. This type of parameter is not learned automatically.
        6. *File Upload* – The parameter of the type file upload in forms is treated as *File Upload* type.
    5. **Values** – Define a fixed set of strings to match against the parameter's value, if the parameter **Type** is to *Global Choice*.
    6. **Parameter Class** – Select a parameter class to be compared to the parameters sent in

the requests/responses.

7. **Custom Parameter Class** – Select the custom parameter class to be compared to the parameters sent in the requests/responses. This is applicable only when **Parameter Class** is set to *CUSTOM*.
8. **Max Value Length** – Set the maximum allowable length for the value of the parameter. Example: The parameter "param" set to 0, which means:
p1=v1&param=&p2=v2 : allowed
p1=v1&param=v&p2=v2 : not allowed
9. **Required** – Set to *Yes* if the parameter must always be present in the request.
10. **Ignore** – Set to *Yes* if the parameter must be ignored completely, that is, never validate the value of the parameter at all.
11. **Maximum Instances** – Specify the maximum number of times the parameter should be allowed in the request/response.
12. **File Upload Extensions** – Define the extensions to be allowed in file upload. '.' is a special extension which indicates no extension, and * is a wildcard which indicates any extension is allowed.

5. Click **Add** to add the Parameter profile.
6. Click **Edit**next to the created parameter profile to specify values for the following fields:
    1. **Allowed Metacharacters** – Define the list of meta-characters to be allowed in spite of it being marked as denied in the parameter class. Click the **Edit** icon, select the meta-characters and click **Apply** to populate the selected meta-characters.
    2. **Exception Patterns** – Define a list of patterns to be allowed as exceptions in spite of them being part of a malicious pattern group. The configuration should be the exact "Pattern Name" as found on the **SECURITY > View Internal Patterns** page or as defined during the creation of a "New Group" through the **SECURITY > Libraries** page. The pattern name can also be found in a Web firewall log when a false positive occurs due to such a potentially "exception" pattern. For example, if the parameter value matched "sql-comments" regex pattern under "sql-injection medium" attacks on the **SECURITY > View Internal Patterns** page, then adding "sql-comments" to this list will allow "sql-comments" in future.
7. Click **Save Changes** to save the above settings.

© Barracuda Networks Inc., 2019 The information contained within this document is confidential and proprietary to Barracuda Networks Inc. No portion of this document may be copied, distributed, publicized or used for other than internal documentary purposes without the written consent of an official representative of Barracuda Networks Inc. All specifications are subject to change without notice. Barracuda Networks Inc. assumes no responsibility for any inaccuracies in this document. Barracuda Networks Inc. reserves the right to change, modify, transfer, or otherwise revise this publication without notice.