Parameter Profiles

https://campus.barracuda.com/doc/98216222/

To protect from attacks that employ the parameters of a URL query string or parameters of the form POST parameters, use Parameter Profiles. They defend web applications from parameter-based attacks.

> Global Parameters can still be found under the Parameters Protection component. Parameters described here are URL specific.

Parameters that contain special characters may have SQL or HTML tagging expressions embedded in them. Embedded SQL keywords like "OR", "SELECT", or "UNION" in a parameter, or system commands such as "xp_cmdshell" can exploit web application vulnerabilities. These attack patterns can be configured in Parameter Profiles, and compared to requests. If a parameter matches, the corresponding request is not processed.

Configure a Parameter Profile

1. Go to App Profiles.
2. Click on the URL segment you would like to protect with a Parameter Profile. You can always click **Add URL** if it's not already there.
3. In the right panel, click **Add New**.
4. From the choices available, select **Parameter.**
    - You must first add **URL Profile** security policy before adding a Parameter Profile.
5. Configure the following fields:
    - **Status** – Set to **Enabled** to validate the requests using this parameter profile.
    - **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.
    - **Type** – Select the type of parameter to be validated in requests/responses. The types of parameters are:
        - **Input** – The parameter other than File Upload, Global Choice, Read Only, and Session Invariant is treated as Input.
        - **Read Only** – All hidden parameters in the form and query parameters in the URL are learned as Read Only. If an exception occurs while learning, then the type is updated to Input. This type makes the parameter session specific.
        - **Global Choice** – Input parameters like check boxes, radio buttons, and menu parameters in a form are treated as Global Choice.
        - **Session Invariant** – Select this if the parameter value is the same across multiple requests from the same session. For example: *session-id*. This type of parameter is

not learned automatically.
    - **File Upload** – The parameter of the type file upload in forms is treated as File Upload.
- **Protect this parameter against comment spam** – Check the box if you would like this parameter protected against spam commonly added to the comment section of forums, blogs, wikis, or online guestbooks. Note: you cannot protect against comment spam and also perform other validations on this parameter.
- **Parameter Class** – Select a parameter class to be compared to the parameters sent in the requests/responses.
- **Max Value Length** – Set the maximum allowable length for the value of the parameter. For example: If the parameter "param" was set to 0, it would be allowed in this case: `p1=v1&param=&p2=v2`, but not this one `p1=v1&param=v&p2=v2` (v is a single character, which is greater than 0).
    - **Unlimited** – Enable this if you do not want to put a **Max Value Length** on the parameter.
- **Required** – Set to **Enabled** if the parameter must always be present in the request.
- **Ignore** – Set to **Enabled** if the parameter must be ignored completely, that is, never validate the value of the parameter at all.
- **Validate Parameter Name** – Enable this to validate the parameter names in a request against attacks.
- **Maximum Instances of Same Parameter** – Specify the maximum number of times the parameter should be allowed in the request/response.
    - **Unlimited** – Enable this if you do not want to put a maximum number of instances that the parameter can occur in a request.
- **Base64 Decode Parameter Value** - Enable this to apply base64 decoding to the parameter values. If the parameter value adheres to the data URI scheme, the base64 decoding is applied on the parameter value regardless of whether Base64 Decode Parameter Value is Enabled or Disabled. If not, the base64 decoding is applied to the parameter value only when Base64 Decode Parameter Value is Enabled. Once the decoding is successful, other parameter checks are enforced as per the policy settings.

  The parameter value length check is always applied on the encoded/original value.

6. Click **Add**.