Configuring Parameter Protection

https://campus.barracuda.com/doc/4259861/

To protect a service from attacks that employ the parameters of a URL query string or parameters of the form POST parameters, use **SECURITY POLICIES > Parameter Protection**. Parameter Protection defends web applications from parameter-based attacks when parameter profiles are not used.

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 Protection, and compared to requests. If a parameter matches, the corresponding request is not processed.

Configure Parameter Protection

1. Go to the **SECURITY POLICIES > Parameter Protection** page.
2. Select the policy whose Parameter Protection settings you want to configure from the **Policy Name** drop-down list.
3. In the **Parameter Protection** section, configure the following fields:
   - **Enable Parameter Protection** – Select *Yes* to enforce Parameter Protection when **Parameter Profiles** are not used for validating the incoming requests.
     - **Values**: Yes, No
     - **Recommended**: Yes
   - **Denied Metacharacters** – Specify disallowed metacharacters in parameters. Non-printable characters such as "backspace" and UI-reserved characters like "?" should be URL encoded. Denied metacharacters help prevent SQL Injection and cross-site scripting attacks. Some specified metacharacters may be valid for some parameters, resulting in valid requests being blocked. The metacharacter list should be appropriately tuned for specific parameters to avoid this problem. To add metacharacters, click the **Edit** icon enter disallowed values.
   - **Maximum Parameter Value Length** – Specify the maximum allowed length of any parameter value, including no-name parameters.
     - **Range**: 0 to 1073741824. Leave blank for "unlimited"
     - **Recommended**: 1000
     - **Units**: Bytes
   - **Maximum Instances** – Enter the maximum number of times a parameter is allowed in a request. By default, the value is set to 1. Restricting this value to one (1) avoids a large class of HTTP Parameter pollution attacks and is recommended.
   - **Base64 Decode Parameter Value**– Set to *Yes* to apply base64 decoding to the parameter values. After the decoding is successful, other parameter checks are enforced according to the policy settings.

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

- **Allowed File Upload Type** – Select **Extensions** to allow the files uploaded with extensions specified in **File Upload Extensions**.
  Select **Mime Types** to identify the content in the files before allowing to be uploaded with the mime types specified in **File Upload Mime Types**.
- **File Upload Extensions** – Specify the extensions of files that may be uploaded. '.' is a special extension allowing files with no extension, and '*' allows any extension.
- **File Upload Mime Types**– Specify the mime types that are to be allowed as uploaded files. Use a "." to indicate a file with unknown mime type, and use a * to indicate any kind of mime type.
- **Max Upload File Size** – Specify the maximum allowed size of individual files being uploaded.
  - **Range**: 0 to 51200. Leave blank for "unlimited"
  - **Recommended**: 1024
  - **Units**: Kilobytes
- **Blocked Attack Types** – Select the Attack Types that need to be matched in the requests. Attack Types specify malicious patterns. Parameter values that match one of the specified Attack Types indicate an intrusion and are logged on the **BASIC > Web Firewall Logs** page.
  Attack Types are defined by groups of regular expression patterns. Attack Types for SQL Injection, cross-site scripting and System Command Injection attacks are provided by default, one or more of which can be enabled for comparison to request parameters.

  Each security policy is configured with a default set of attack types that are applied to the matching requests. For more comprehensive validation, select other attack type patterns.
- **Custom Blocked Attack Types** – Select the custom attack types that need to be matched in the requests. For information on how to create custom blocked attack types, see Configuring User Defined Patterns.
- **Exception Patterns** – Enter patterns that should be allowed despite matching a malicious pattern group. Configure the exact "Pattern Name" displayed on the **ADVANCED > View Internal Patterns** page, or configured creating a "New Group" on the **ADVANCED > Libraries** page. The pattern name is also displayed in the Web Firewall Log when it is incorrectly denied (a false positive). For example, if the parameter value matched "sql-comments" regex pattern under "sql-injection medium"on the **ADVANCED > View Internal Patterns** page, then add "sql-comments" to the list to allow "sql-comments" in future.
- **Ignore Parameters** – Specify parameters exempt from all validations. Use this to skip validations for especially large parameters that are automatically generated by servers, such as __VIEWSTATE. Since these parameters are auto-generated, they are less likely to be attacks, and therefore can safely be exempted from validation checks. Note: Ignore Parameter is an exact match; wildcard is not supported. So a value with "*" does not work like a wildcard. Examples: __VIEWSTATE, POSTBODY

4. Click **Save**.