Configuring URL Protection

https://campus.barracuda.com/doc/4259876/

URL requests and embedded parameters in them can contain malicious script. Attacks embedded in URL requests or their parameters are executed with the permissions of the executing component. Injection of operating system or database commands into the parameters of a URL request, cross site scripting, remote file inclusion attacks, and buffer overflow attacks can all be perpetrated through unchecked URL requests or their parameters.

Here is an example of malicious script within a URL Request:

```
http://www.example.com/sharepoint/default.aspx/%22
);}if(true){alert(%22qwertytis
```

Defense from these attacks is achieved by restricting the allowed methods in headers and content for invoked URL requests, restricting the number of request parameters and their lengths, limiting file uploads, and specifying attack types to explicitly detect and block. (Attack types are configured on **ADVANCED > Libraries** or **ADVANCED > View Internal Patterns**.) URL Protection uses a combination of these techniques to protect against various URL attack types. URL Protection defends the Service from URL request attacks when no URL Profile is configured to do it. For information on URL Profiles, see Configuring Website Profiles.

Steps To Configure URL Protection

1. Go to the **SECURITY POLICIES > URL Protection** page.
2. Select the policy from the **Policy Name** drop-down list for which you want to modify URL protection settings.
3. In the **URL Protection** section, specify values for the following fields:
    1. **Enable URL Protection** – Select *Enable* to enforce **URL Protection** when **URL Profiles** are not used for validating the incoming requests.
        - **Values**: Enable, Disable
        - **Recommended**: Enable
    2. **Allowed Methods** – Specify the list of methods to be allowed in the request. The Barracuda Web Application Firewall uses this list to determine whether to allow or deny methods in the requests. See Limiting Allowed Methods in HTTP Headers and Content.
    3. **Allowed Content Types** – Specify the list of content types to be allowed in the POST body for a URL. "`application/x-www-form-urlencoded`" and "`multipart/form-data`" are typical content types that are used in form submissions. These content types are recognized by the Barracuda Web Application Firewall and cause it to parse the content into parameters and values. Other content types are used by custom built applications in various special ways. For example, text/xml may be used by Web Services enabled applications. These content types, when encountered in the request, are not given any special consideration, and are passed through to the server if they are allowed by this setting.

4. **Max Content Length** – Enter the maximum content length to be allowed for POST request body. **Note**: Only requests with the Content-Length: headers are validated. Requests encoded using "Chunked Encoding" DO NOT have a Content-Length: header, and therefore are not subject to the Content Length check.
   - **Range**: 0 to 8388608 bytes. No value (empty) implies unlimited.
   - **Recommended**: 32768 bytes
5. **Max Parameters** – Enter the maximum number of parameters to be allowed in a request. Parameters can be supplied as part of the query string or as part of the request body or both. This limit applies to each method of supplying the parameter individually. For example, if the **Max Parameters** is 4, a maximum of 4 query string parameters are allowed and a maximum of 4 request body parameters are allowed. Thus, when both methods are combined, a maximum of 8 parameters will be allowed (4 each).
   - **Range**: 0 to 1024. No value (empty) implies unlimited.
   - **Recommended**: 40
6. **Max 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.
   - **Range**: 0 to 1024. No value (empty) implies unlimited.
   - **Recommended**: 5
7. **Max Parameter Name Length** – Enter the maximum length of the parameter name in the request.
   - **Range**: 0 to 1024. No value (empty) implies unlimited.
   - **Recommended**: 64
   - **Units**: Bytes
8. **Blocked Attack Types** – Select the attack types that needs to be matched in the requests/responses. 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.
   > Each security policy is configured with default set of attack types that are applied to the matching requests. For more comprehensive validation, select other attack type patterns.
9. **Custom Blocked Attack Types** – Select the custom attack types that needs to be matched in the requests/responses. For information on how to create custom blocked attack types, see Configuring User Defined Patterns.
10. **Exception Patterns** – Enter the 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 **ADVANCED > View Internal Patterns** page, or as defined during the creation of a "New Group" through the **ADVANCED > 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 **ADVANCED > View Internal Patterns** page, then adding "sql-comments" to this list will allow "sql-

comments" in future.

4. Click **Save**.