Tuning Security Rules Using Web Firewall Logs

Introduction

Barracuda Web Application Firewall enables administrators to configure security rules with varying degrees of granularity. A security policy, comprised of security settings, is shared by multiple applications.

A newly configured service originally uses the 'default security policy', so all URLs and Parameters are compared to the 'default security policy' settings. The Barracuda Web Application Firewall applies rules to traffic and generates a log of rule violations viewable in the **BASIC > Web Firewall Logs** page.

You can use the Web Firewall Logs to evaluate rule violations, and when warranted, create exceptions to the rule violated. Exceptions can apply globally if they modify the security policy, which affects all services using that policy. Or you can apply an exception locally that only applies to a specific website or URL. To create a fine grained exception, you use the **WEBSITES > Allow/Deny** OR **WEBSITES > Website Profiles** pages.

The default security policy associated with a Service might sometimes end up blocking genuine requests, which are called false positives. To reduce false positives you can enable **Exception Profiling** for desired websites on the **WEBSITES** tab. Exception profiling uses heuristics displayed on the **WEBSITES > Exception Heuristics** page to identify false positives. You can set the exception profiler to automatically refine security policy rules for the respective site section by setting **Request Violation Handling** to **Auto** on the **Exception Heuristics** page; alternatively, set **Request Violation Handling** to **Manual** if you want the profiler to generate policy recommendations under **Pending Recommendation** on **WEBSITES > Exception Profiling**. In this case, the administrator must review the violations, and manually apply desired fixes. See Using Exception Profiling to Generate Recommendations for Tuning.

Automatic settings are recommended for trusted hosts.

Creating Exceptions using the Web Firewall Logs

Once logged in to the unit, select **Web Firewall Logs** from the **BASIC** tab to search for a log entry believed to be a false positive. These log entries will be in red and have an action of DENY (active mode) or LOG (passive mode).

Time	Event Details		Client Details		Attack Details		Actions
DENIED	URL	/~index.html					
Time 22:44:38.690	Service IP:Port	99.99.224.2:80	Client IP	99.99.48.1	Attack Name	Tilde in URL Path	Fix Details
Date 2017-02-08	Service Name	service1	Country	US	Attack Detail		
ID 15a219ddea1-a032f19	Protocol	HTTP	Method	GET	Rule	security-policy	
DENIED	URL	/index.html					
Time 22:43:25.29	Service IP:Port	99.99.224.2:80	Client IP	99.99.48.1	Attack Name	SQL Injection in Parameter	Fix Details
Date 2017-02-08	Service Name	service1	Country	US	Attack Detail	type="sql-injection-medium" pat	
ID 15a219cbee5-a032f19	Protocol	HTTP	Method	GET	Rule	security-policy	
LOGGED	URL	/index.html					
Time 22:40:15.241	Service IP:Port	99.99.224.2:80	Client IP	99.99.48.1	Attack Name	SQL Injection in Parameter	Fix Details
Date 2017-02-08	Service Name	service1	Country	US	Attack Detail	type="sql-injection-medium" pat	
ID 15a2199d95c-a032f19	Protocol	HTTP	Method	GET	Rule	security-policy	
LOGGED	URL	/index.html					
Time 22:40:15.241	Service IP:Port	99.99.224.2:80	Client IP	99.99.48.1	Attack Name	SQL Injection in Parameter	Fix Details
Date 2017-02-08	Service Name	service1	Country	US	Attack Detail	type="sql-injection-medium" pat	
ID 15a2199d95c-a032f19	Protocol	HTTP	Method	GET	Rule	security-policy	

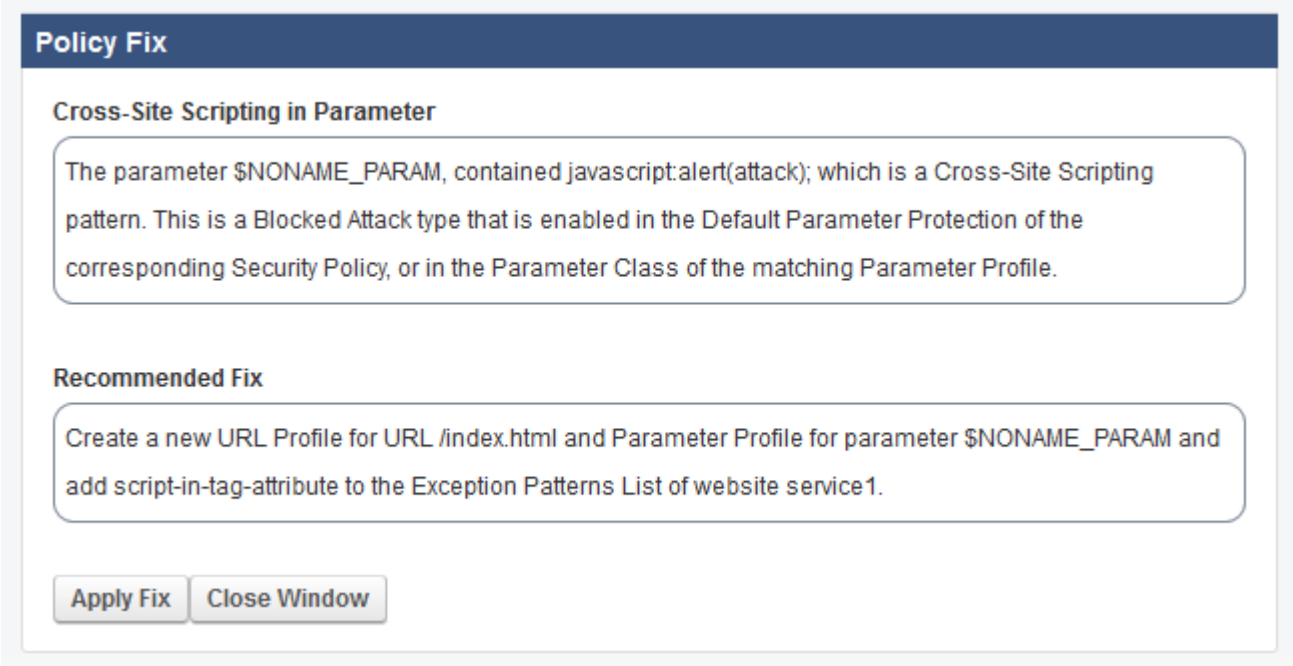
Scroll over to the right of the selected log and click **Fix**. A **Policy Fix** window will appear.

The fix recommended by the Barracuda Web Application Firewall may be localized or global, depending upon which rule was violated. Accepting a recommendation could have the following impact:

1. **Web site profile (localized) modification:** As the most fine-grained security, changes impact only a given URL or parameter.
2. **Security Policy (global) modification:** As a policy shared by multiple applications, changes impact all applications using the security policy.

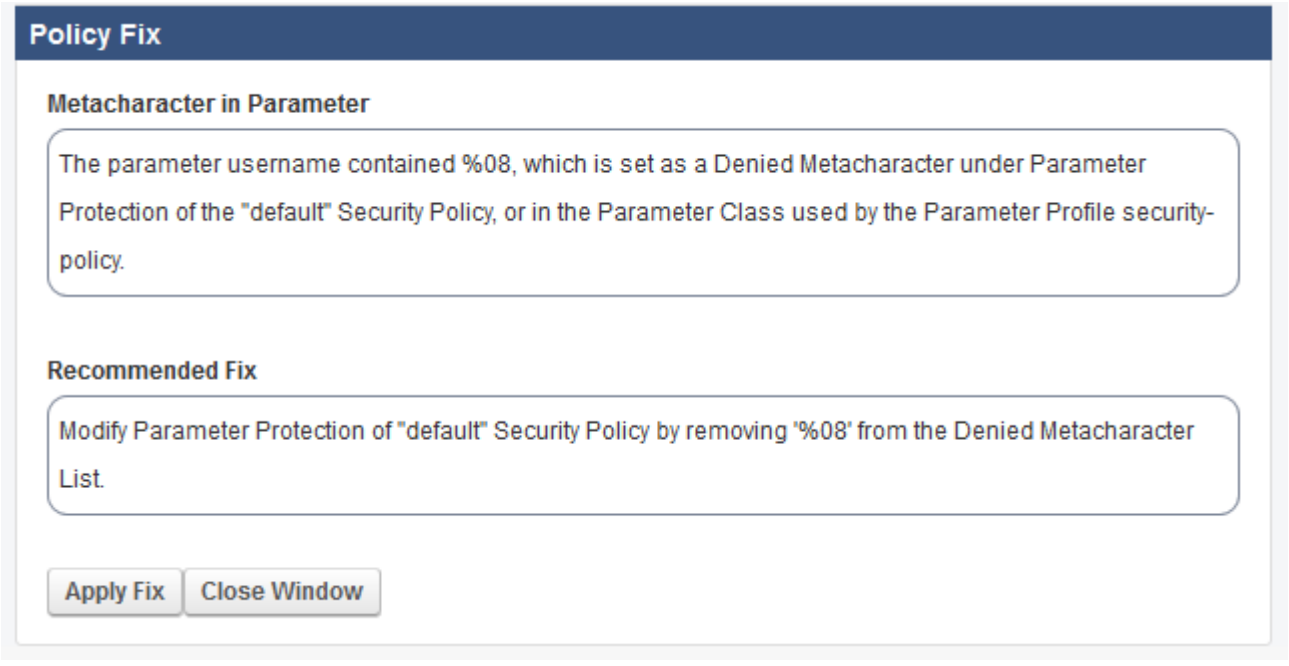
Examples of Fixes Suggested by the Barracuda Web Application Firewall

Example 1: Recommendation to configure a fine grained rule.



Following the recommendation to create a URL Profile for /modules.php creates an exception only for that particular page.

Example 2: Recommendation to change the configuration in Security Policy.



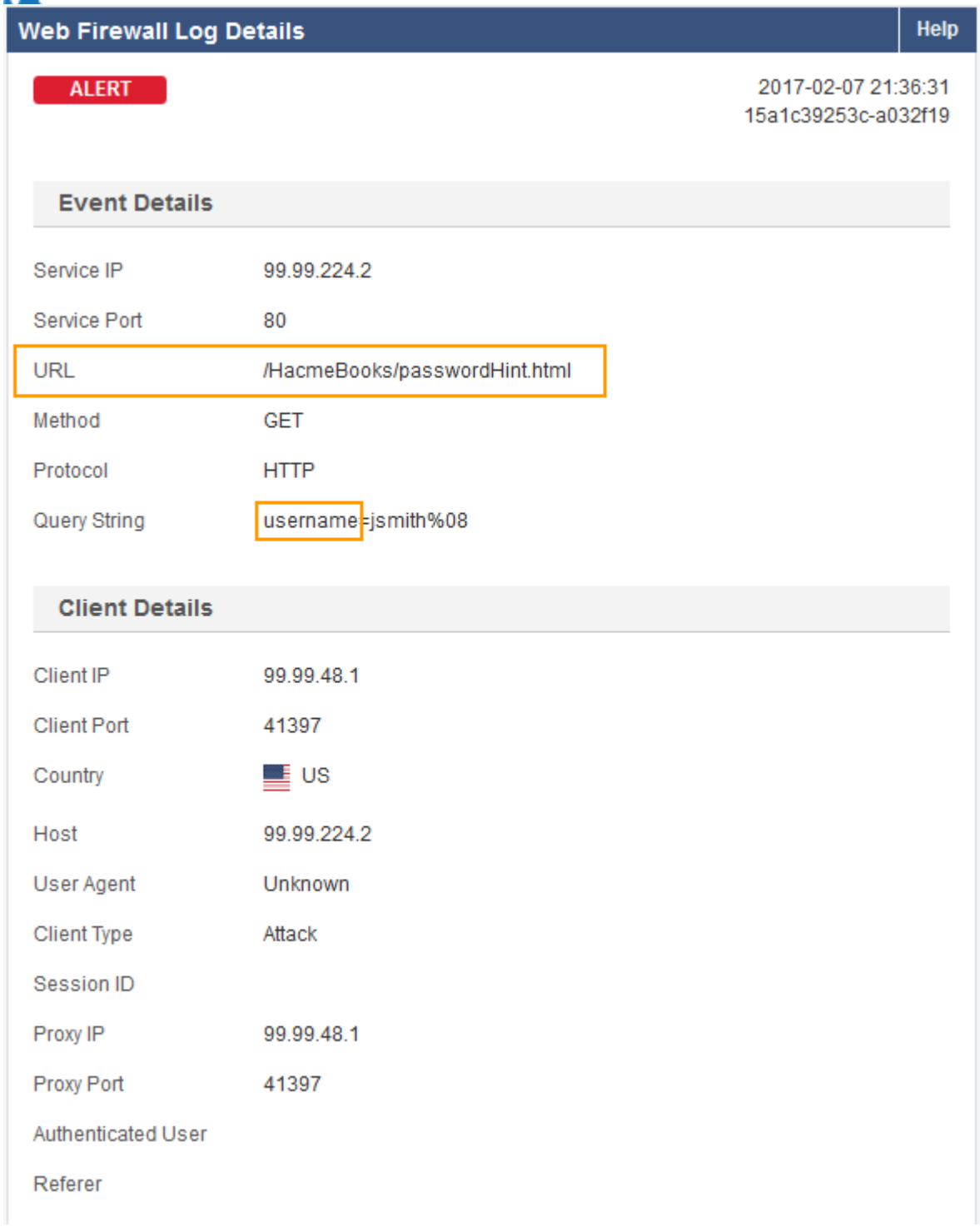
The Barracuda Web Application Firewall suggested change to the ‘Parameter Protection’ sub policy of the ‘default Security Policy’ would allow the Meta character (%08) in any parameter for any application using this security policy. To avoid an exception which applies globally, you can add an exception which only applies to the URL or parameter noted in the log.

Manually Configuring a Fine Grained Rule

When you want to apply a local exception instead of a recommended global fix you need to manually do a two step process.

Step 1: Figure out the exception specifics from Web Firewall Logs.

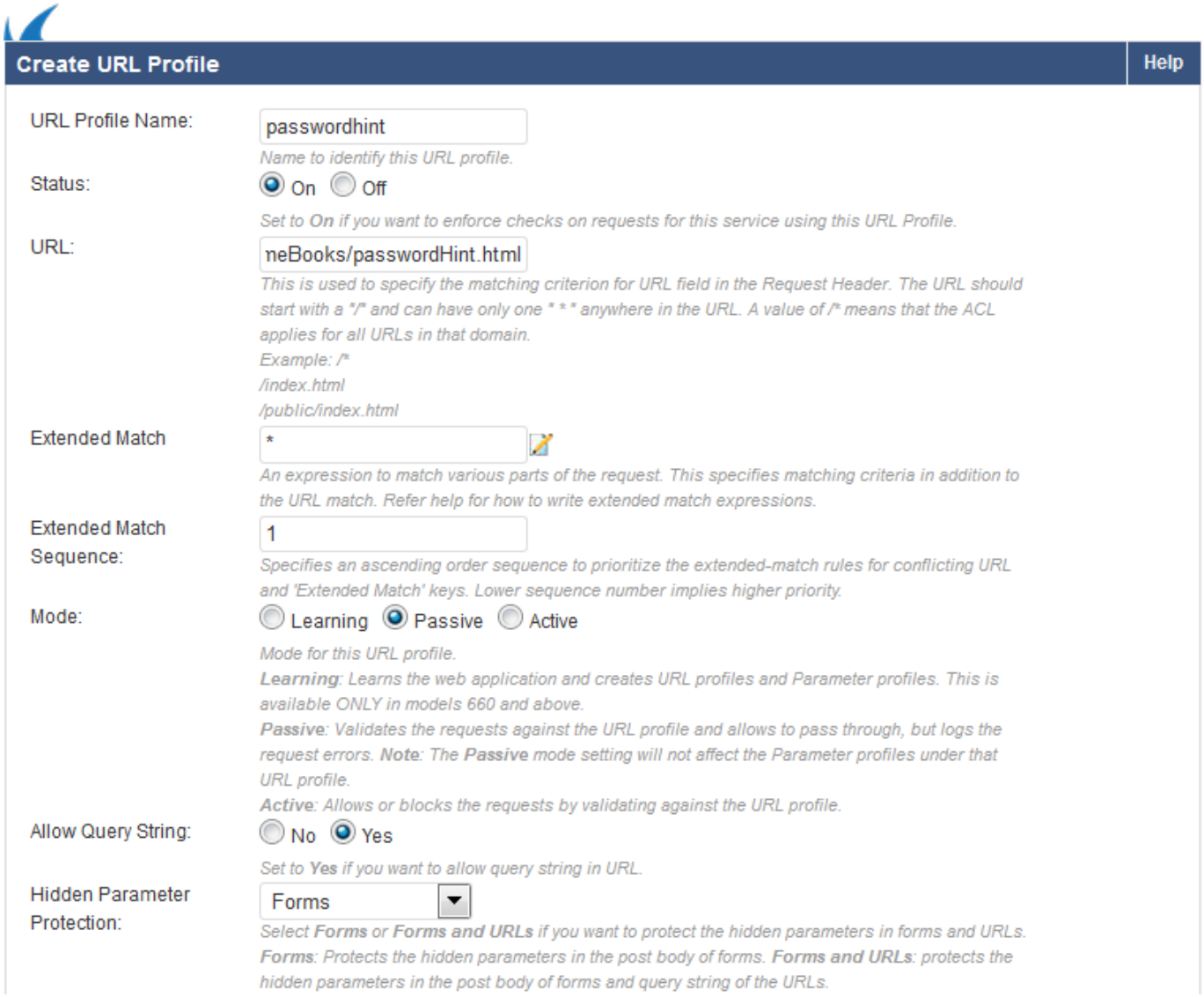
- Scroll over to the selected log in the Web Firewall Logs and click **Details**.
- Select the URL.
- Note the parameter name in the **Query String**.
- Close the **Web Firewall log Details** window.



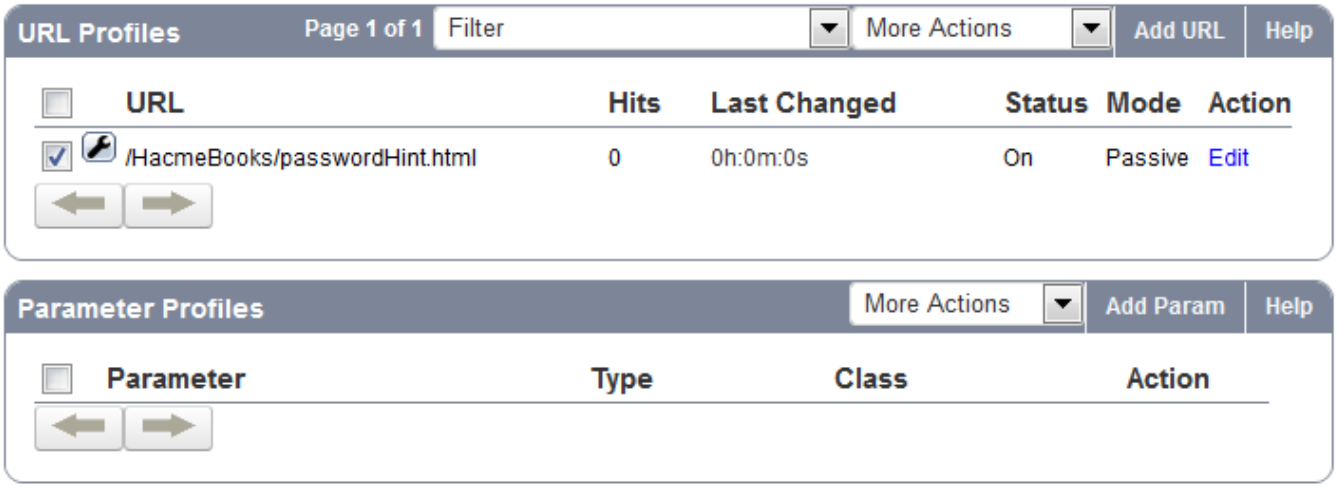
Example: For the log shown above – VIP is **192.168.9.96** – Port 80. The URL is /HacmeBooks/passwordHint.html and the URL has a parameter called 'username'.

Step 2: Configure the Exception.

- From the **WEBSITES** > **Website Profiles** page select the appropriate service from the **Website** drop down list (**192.168.9.96 : 80**).
- In the **URL Profiles** section, click **Add URL**.
- The **Create URL Profile** window appears.
- Enter a name in the **URL Profile Name** field.
- Paste the URL into the **URL** field (/HacmeBooks/passwordHint.html).
- Click **Add**.



You should now see the new URL profile in the URL Profile section. Click **Edit** to make the necessary security exceptions to the URL. Click **Save** when done.



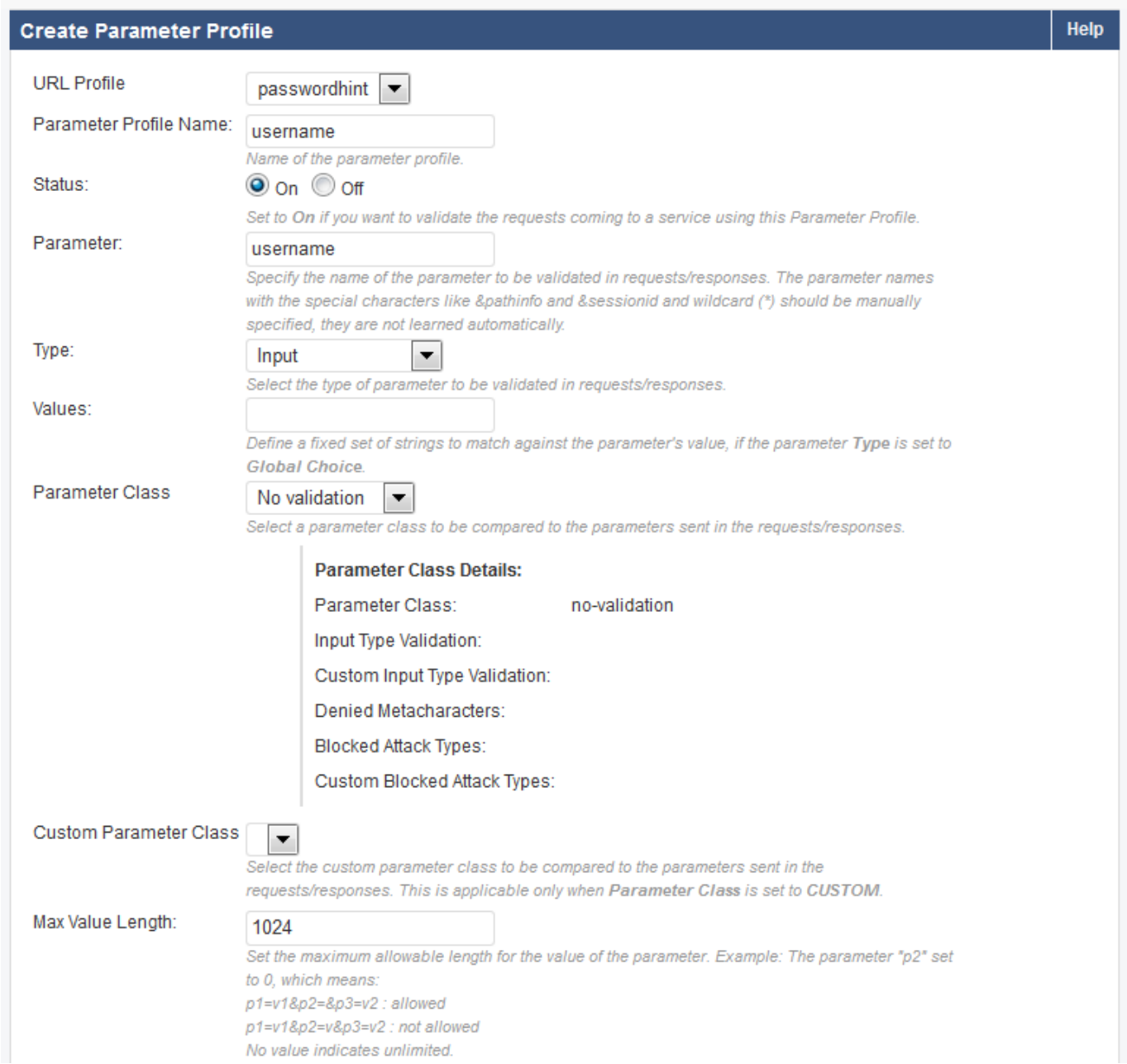
To specify parameter settings, you will need to configure Parameter profiles for the relevant URL Profile (Example: passwordHint).

- Click **Add Param** in the **Parameter Profiles** section.
- The **Create Parameter Profile** window appears.

- Select the appropriate URL Profile from the drop down list (Example: passwordHint).
- Enter a name in the **Parameter Profile Name** field.
- In the **Parameter** field, enter the parameter that you noted from the details of the Web Firewall Logs.
- Select the appropriate Parameter Class – typically 'No Validation' is selected if it isn't a specific class.
- Click **Add**.

You should now see the new parameter profile in the Parameter Profile section. Click **Edit** to make the necessary exceptions to the Parameter. Click **Save** when done.

The example below shows the created local exception to allow meta character %08 in parameter username for URL profile passwordhint.



URL Profiles Page 1 of 1 | Filter | More Actions | Add URL | Help

	URL	Hits	Last Changed	Status	Mode	Action
☑	/HacmeBooks/passwordHint.html	0	0h:0m:0s	On	Passive	Edit

Parameter Profiles Page 1 of 1 | More Actions | Add Param | Help

	Parameter	Type	Class	Action
☐	username	Input	No validat...	Edit

Edit Parameter Profile Help

Parameter Profile Name: username

Parameter: username

Status: (•) On ( ) Off

*Set to **On** if you want to validate the requests coming to a service using this Parameter Profile.*

Type: Input

Select the type of parameter to be validated in requests/responses.

Values: [ ] Add

*Define a fixed set of strings to match against the parameter's value, if the parameter **Type** is set to **Global Choice**.*

Parameter Class: No validation

Select a parameter class to be compared to the parameters sent in the requests/responses.

Parameter Class Details:

Parameter Class: no-validation

Input Type Validation:

Custom Input Type Validation:

Denied Metacharacters:

Blocked Attack Types:

Custom Blocked Attack Types:

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**.*

Max Value Length: 1024

Set the maximum allowable length for the value of the parameter. Example: The parameter "p2" set to 0, which means:
p1=v1&p2=&p3=v2 : allowed
p1=v1&p2=v&p3=v2 : not allowed
No value indicates unlimited.

Required: (•) No ( ) Yes

*Set to **Yes** if the parameter must always be present in the request.*

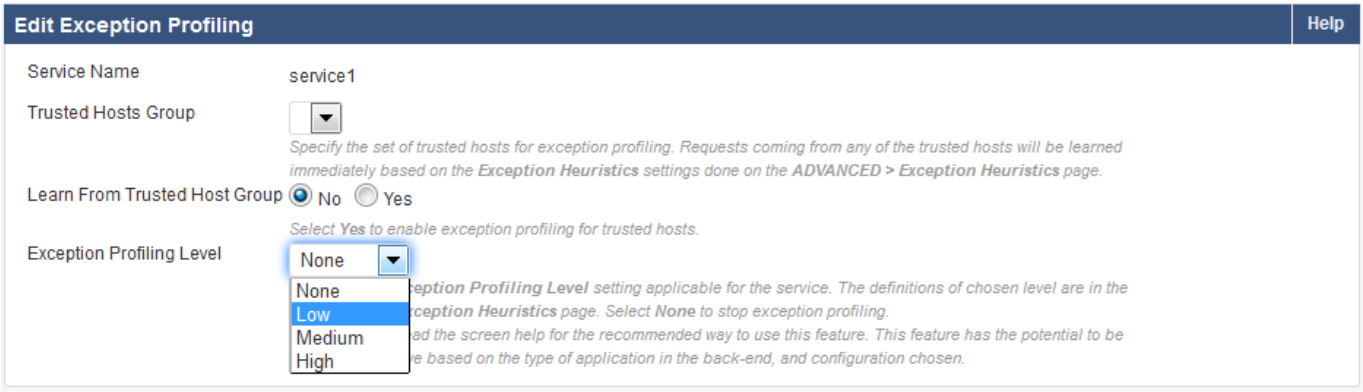
Ignore: (•) No ( ) Yes

Using Exception Profiling to Generate Recommendations for Tuning

To configure exception profiling for a Service:

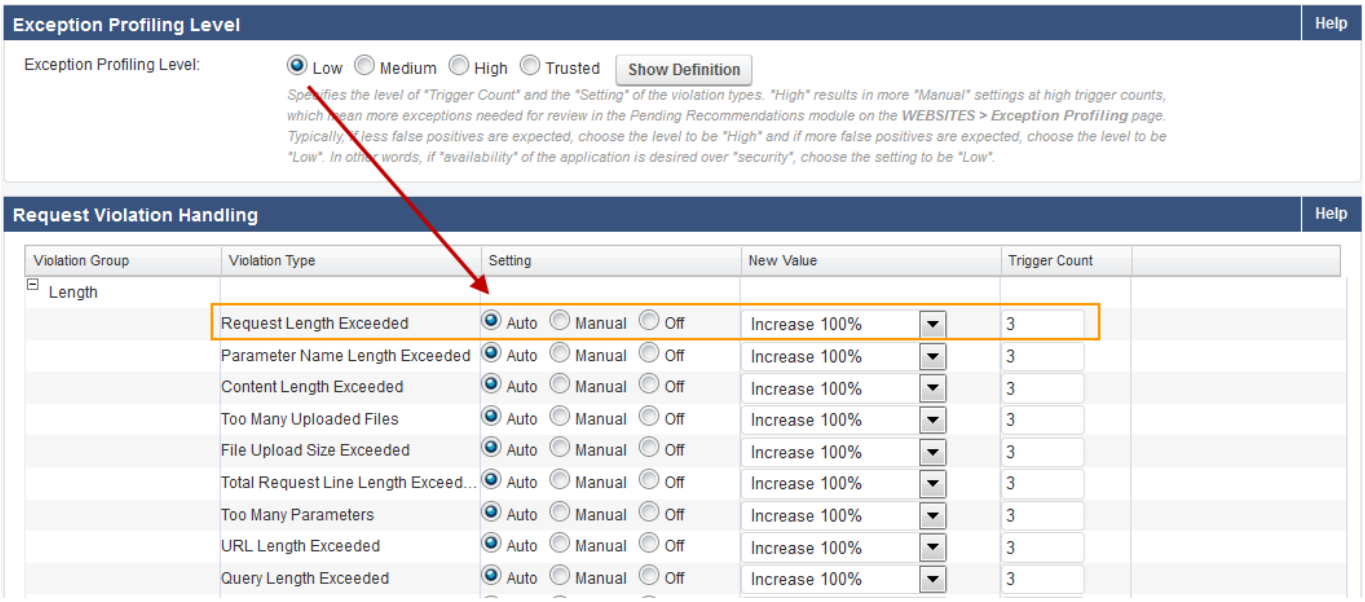
1. From the **WEBSITES > Exception Profiling** page identify the Service for which you want exception profiling enabled.
2. Click **Edit** next to that Service. The **Edit Exception Profiling** window appears.
3. To learn from a trusted hosts group, select the trusted host group from the **Trusted Hosts Group** drop-down list and set **Learn From Trusted Host Group** to *Yes*. For more information, see Fine Tuning Security Settings for a Trusted Hosts Group using Exception Profiling.
4. To learn from untrusted traffic, select the level of tolerance to violations (Low, Medium or High) from the **Exception Profiling Level** drop-down list**.** For more information on Exception Profiling, see How to Configure Exception Profiling**.**
5. Click **Save**.

The figure below shows the **Exception Profiling Level** set to *Low* for untrusted traffic.



Exception profiling provides default settings for each violation type. The settings indicate how exceptions update profiles (Automatically, Manually, or not at all), how the new setting in the profile is generated (for example, increasing the current value, or accepting the observed value) and how many times the logged error needs to be seen before generating an exception (Trigger Count). These default settings for an Exception Profiling Level can be edited and saved on the **WEBSITES > Exception Heuristics** page.

The figure below displays the default set of heuristics for the **Exception Profiling Level: Low**.



By default, all settings are set to *Auto* for the Exception profiling levels for untrusted traffic. The Trusted settings

are either *Auto* or *Off*. For untrusted traffic, the *Manual* setting requires you to verify the exception before applying it, or you can turn exception profiling *Off* for a particular violation. If the traffic originates from trusted hosts, the trusted policy heuristics apply. If the traffic originates from non-trusted hosts, the selected **Exception Profiling** policy applies.

Manually Fine Tuning the Security Policy Using Exception Profiling

By default, each violation type is set to *Auto* on the **WEBSITES > Exception Heuristics** page. Hence, whenever violations from unique sources are encountered the number of times indicated in **Trigger Count**, the profiles are automatically updated creating the respective profiles for the Service. This applies ONLY when exception profiling is enabled for the Service on **WEBSITES > Exception Profiling**; that is, the **Exception Profiling Level** for the service is not equal to **None**.

To view encountered violations and manually apply desired recommended fixes, do the following:

Step 1: The Setting for the desired Violation Type should be Manual.

1. From the **WEBSITES > Exception Heuristics** page select the **Exception Profiling Level** (Low, Medium or High) you want to modify. **Note**: Trusted does not support Manual exception creation.
2. In the **Request Violation Handling** section, identify the violation type(s) for which you wish to generate recommendations.
3. Change **Setting** to **Manual** next to the violation type(s). Also, change the settings in **New Value** and **Trigger Count** if required.
4. Click **Save**.

Learned false positives are displayed on the **WEBSITES > Exception Profiling > Pending Recommendation** page every 600 seconds (10 minutes).

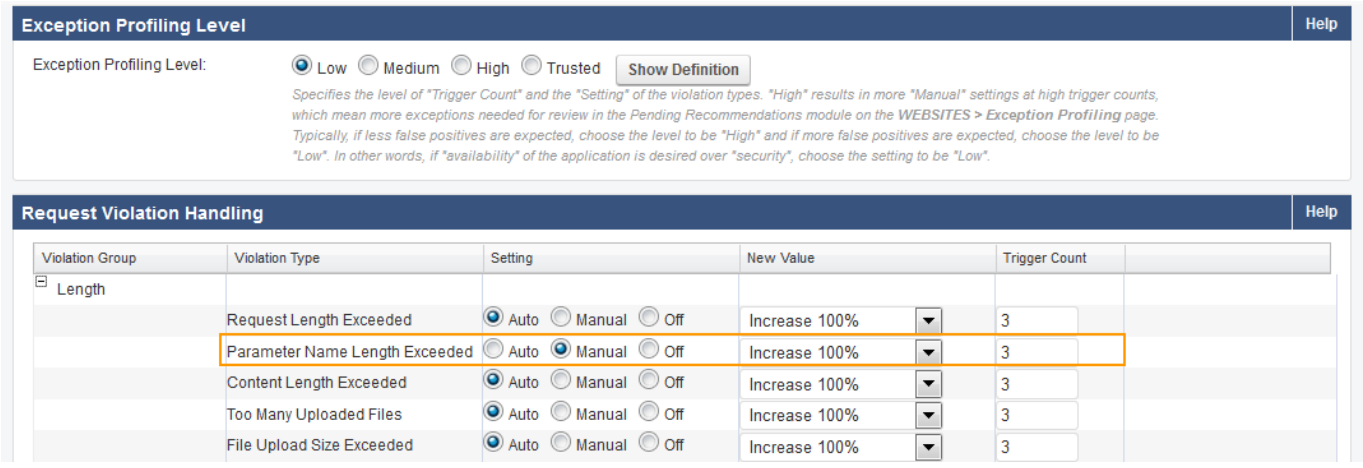
Step 2: Select the recommendation and apply fix.

1. Go to the **WEBSITES > Exception Profiling** page.
2. In the **Pending Recommendations** section, view the recommendations for relevant violation type(s).
3. Select the check box(es) next to the recommendations you want to fix, and click **Apply Fix**.

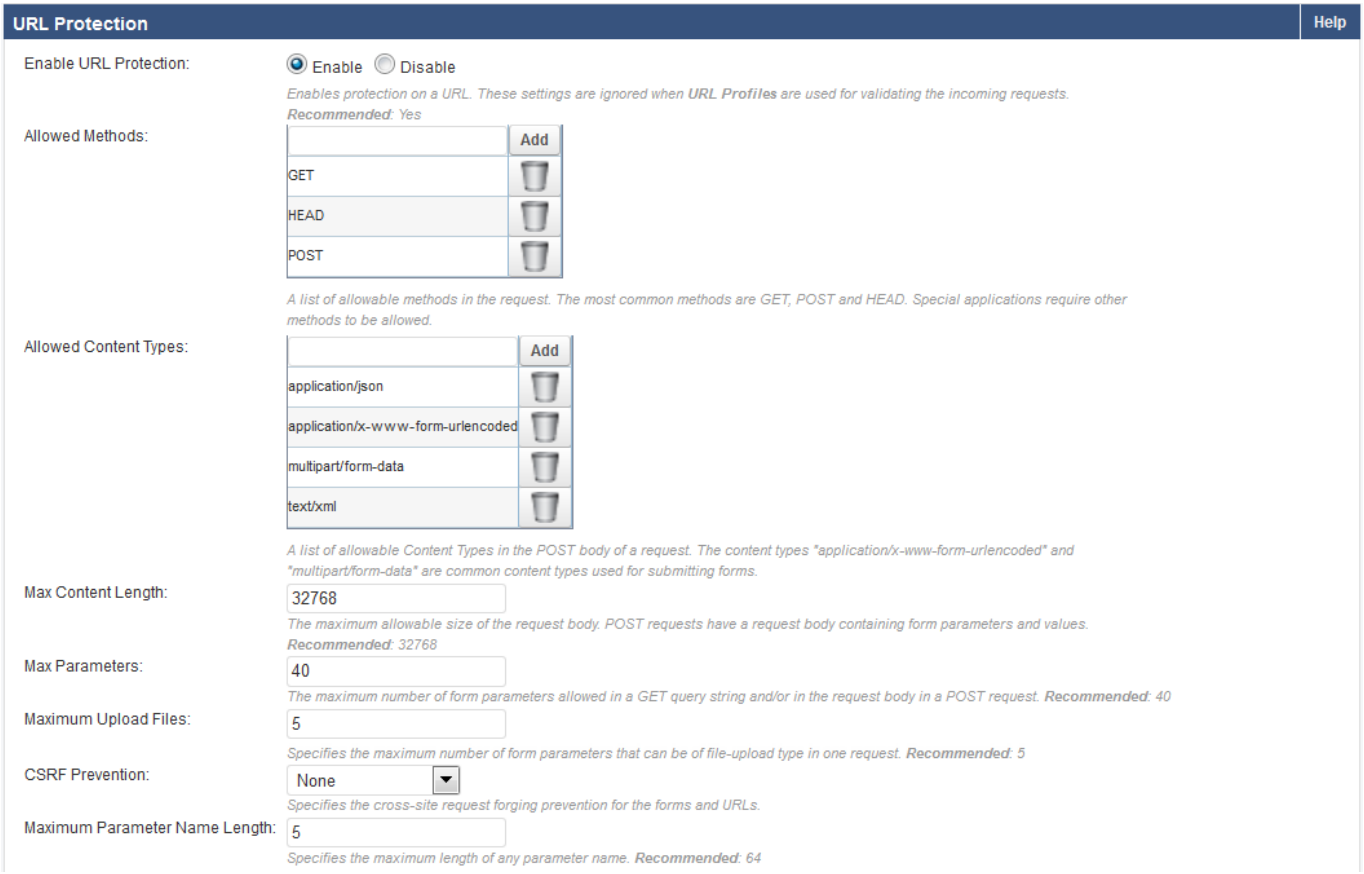
Examples for Tuning the Security Policy Using Exception Profiling

Example 1: Parameter Name length exceeded

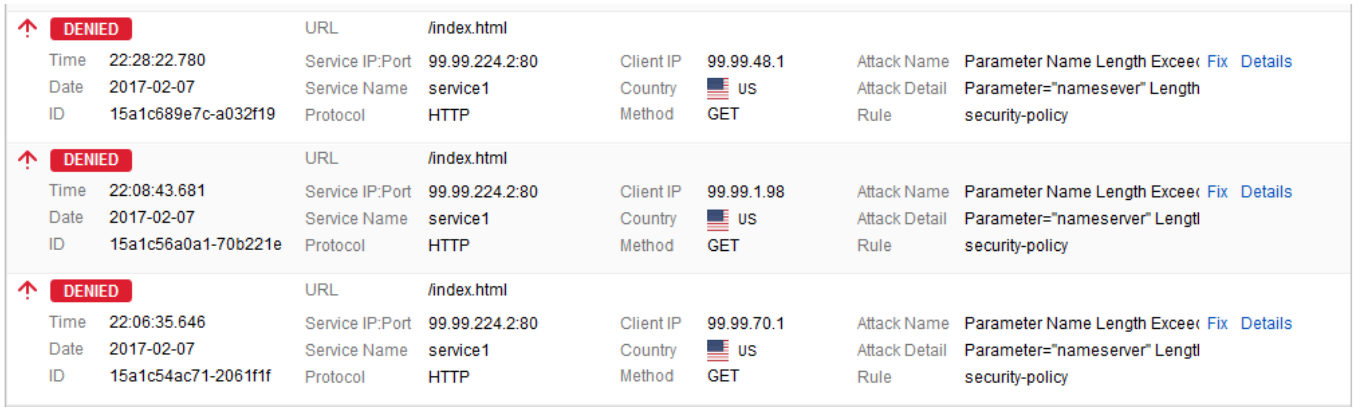
In this example, the violation type **Parameter Name Length exceeded** is set to *Manual*, **New Value** is set to *Increase 100%***,** and **Trigger Count** is set to ***3***.



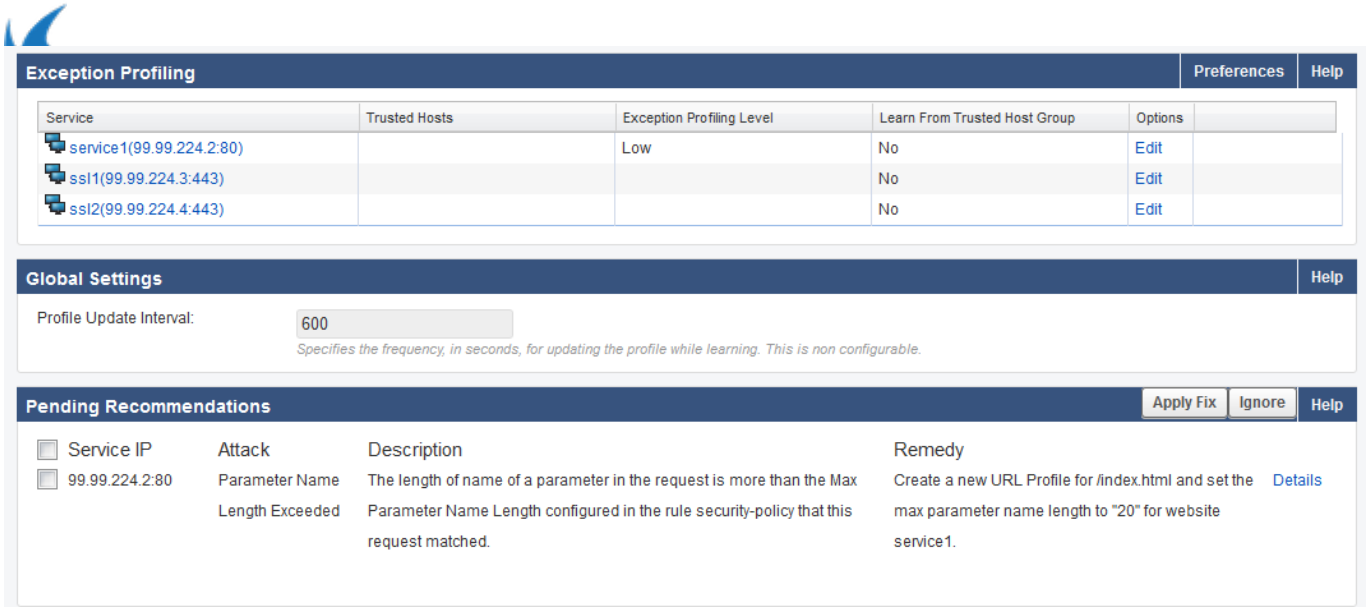
The **Max Parameter Name Length**, set on **SECURITY POLICIES > URL Protection**, is 5.



When three unique clients (based on the value set in **Trigger Count**) send requests with parameter name length 10, the violation is logged under **BASIC > Web Firewall Logs**.

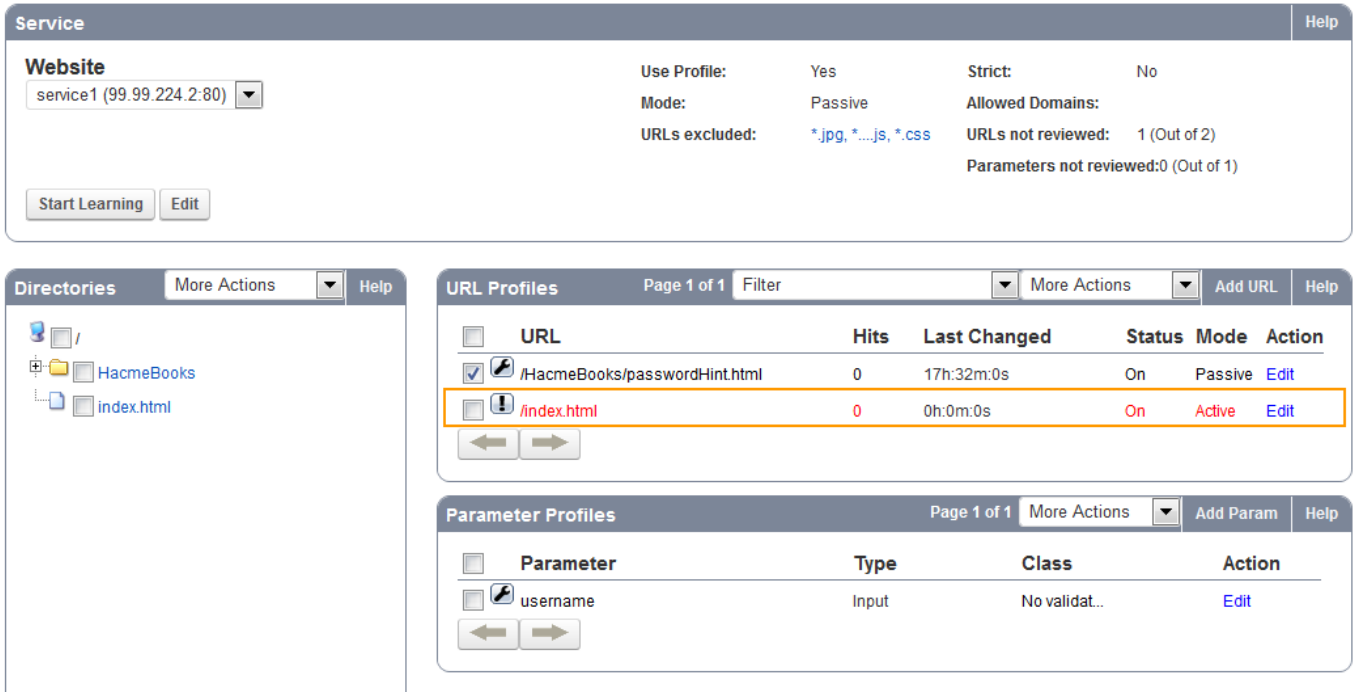


The recommendations are displayed after 600 seconds (10 minutes) on the **WEBSITES > Exception Profiling** page.



Click **Details** to see the log information. Select the check box(es) and click **Apply Fix** to apply the recommended fix.

Since **New Value** for **Parameter Name length exceeded** is set to *Increase 100%* on **WEBSITES > Exception Heuristics** and the parameter length in the request is 10, a new URL profile is created on the **WEBSITES > Website Profiles** page with the **Max Parameter Name Length** set to *20*.



Edit URL Profile

Help

URL Profile Name: reco_d1546d731a9f30cc80127d57142a482b

Created By: Exception

Status: (•) On ( ) Off

*Set to **On** if you want to enforce checks on requests for this service using this URL Profile.*

URL: /index.html

*This is used to specify the matching criterion for URL field in the Request Header. The URL should start with a "/" and can have only one " * " anywhere in the URL. A value of /* means that the ACL applies for all URLs in that domain.*
*Example: /**
/index.html
/public/index.html

Extended Match: *

An expression to match various parts of the request. This specifies matching criteria in addition to the URL match. Refer help for how to write extended match expressions.

Extended Match Sequence: 1

Specifies an ascending order sequence to prioritize the extended-match rules for conflicting URL and 'Extended Match' keys. Lower sequence number implies higher priority.

Mode: ( ) Learning ( ) Passive (•) Active

Mode for this URL profile.
***Learning**: Learns the web application and creates URL profiles and Parameter profiles. This is available ONLY in models 660 and above.*
***Passive**: Validates the requests against the URL profile and allows to pass through, but logs the request errors. **Note**: The **Passive** mode setting will not affect the Parameter profiles under that URL profile.*
***Active**: Allows or blocks the requests by validating against the URL profile.*

Allowed Methods:

	Add
GET	
HEAD	
POST	

Specifies the list of allowable methods in the request.

Allow Query String: ( ) No (•) Yes

*Set to **Yes** if you want to allow query string in URL.*

Allowed Content Types:

	Add
application/x-www-form-urlencoded	
multipart/form-data	

Specifies the list of allowable content-types in the POST body for a URL.

Hidden Parameter Protection: Forms

*Select **Forms** or **Forms and URLs** if you want to protect the hidden parameters in forms and URLs. **Forms**: Protects the hidden parameters in the post body of forms. **Forms and URLs**: protects the hidden parameters in the post body of forms and query string of the URLs.*

CSRF Prevention: None

*Select **Forms** or **Forms and URLs** if you want cross-site request forging prevention for the forms and URLs. This is not applicable when there is no parameter profile.*
***Note:** If this is set to "Forms" or "Forms and URLs", ensure that the service using this policy should have **Use Profile** set to 'Yes' on **WEBSITES > Website Profiles**.*

Max Content Length:

Specify the maximum allowable content length for POST request body.

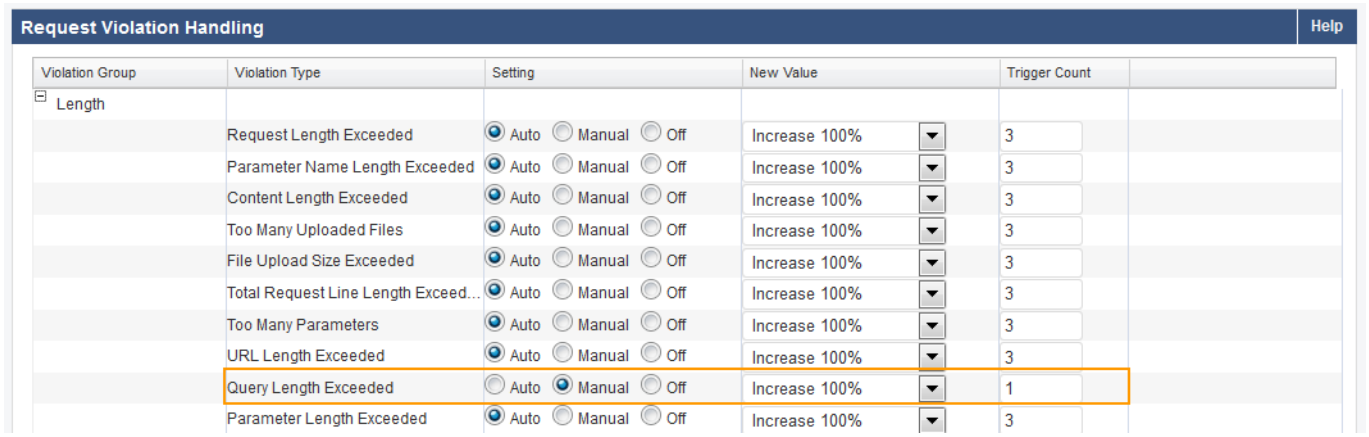
Maximum Parameter Name Length: 20

Specify the maximum length of the parameter name. No value (empty) implies unlimited.

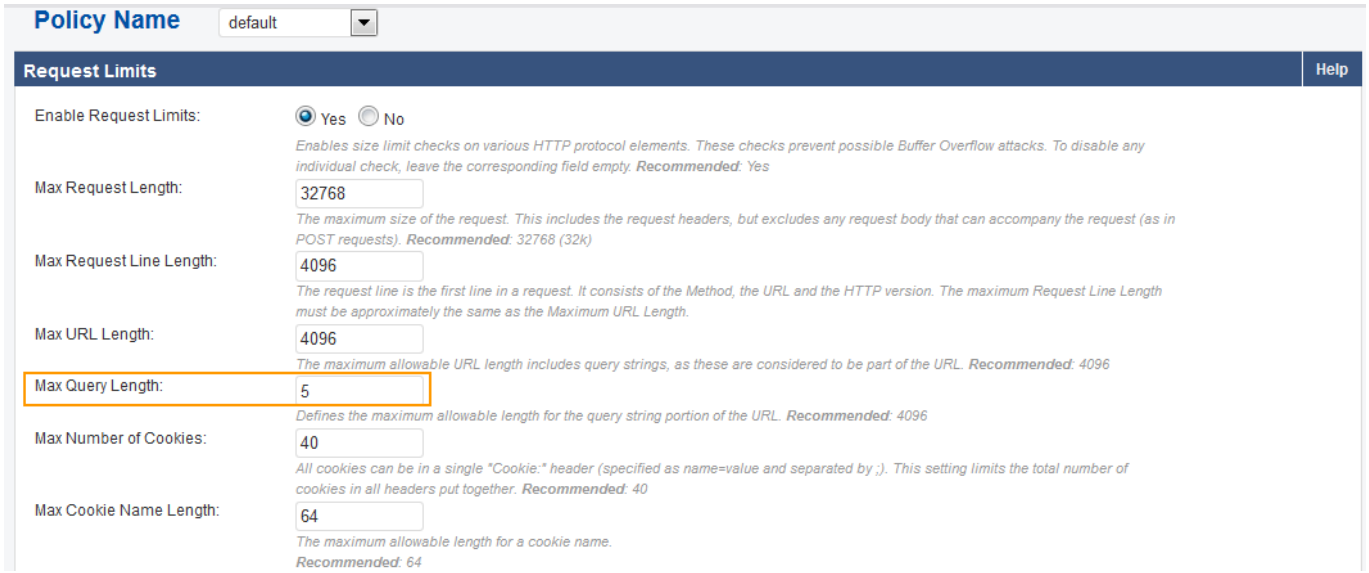
Referrers for the URL Profile: Add

Example 2: Query length exceeded.

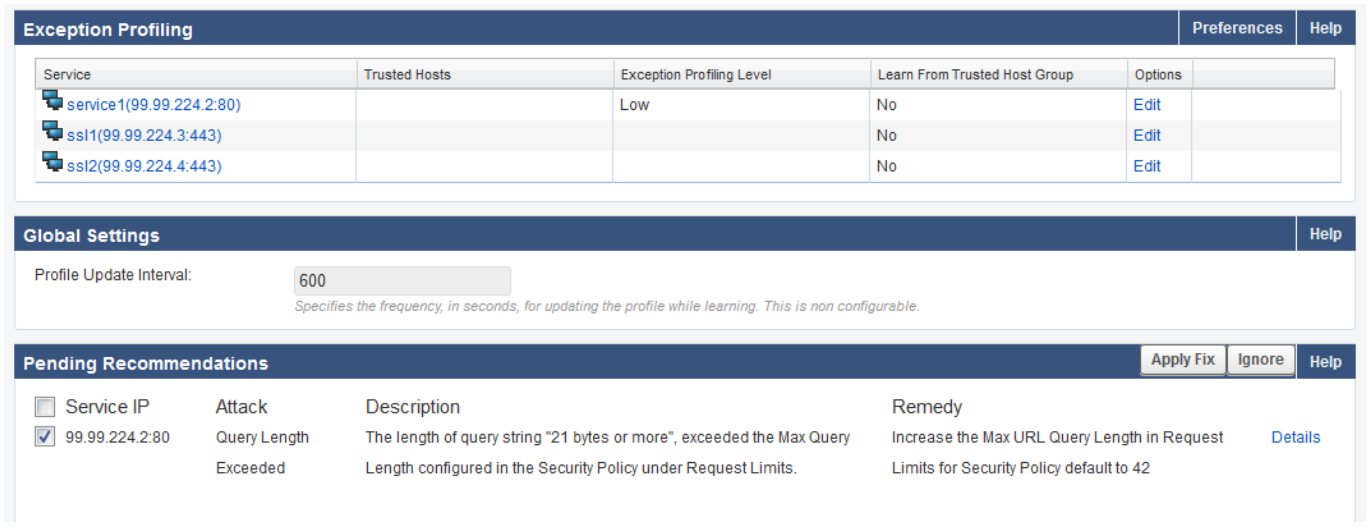
In this example, **Query length exceeded** is set to *Manual*, **New Value** is set to ***Increase 100%*** and **Trigger Count** is set to *3*.



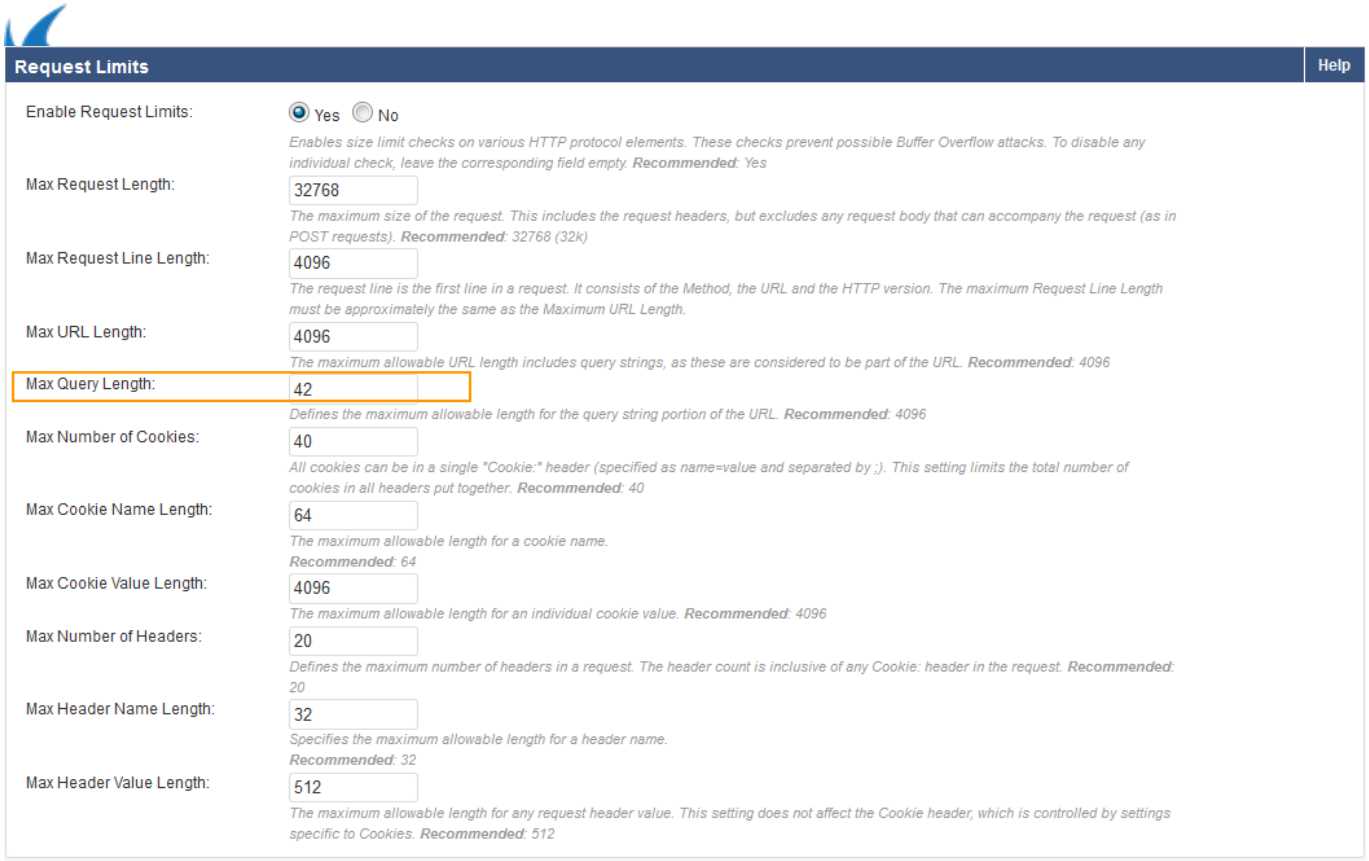
The **Max Query Length**, specified on the **SECURITY POLICIES > Request Limits**, is **5**.



Recommendation generated on **WEBSITES > Exception Profiling**:



Clicking **Apply Fix** increases the **Max Query length** to 24 on **SECURITY POLICIES > Request Limits**.


Request Limits
Help
Enable Request Limits:
Yes
No
Enables size limit checks on various HTTP protocol elements. These checks prevent possible Buffer Overflow attacks. To disable any individual check, leave the corresponding field empty. Recommended: Yes
Max Request Length:
32768
The maximum size of the request. This includes the request headers, but excludes any request body that can accompany the request (as in POST requests). Recommended: 32768 (32k)
Max Request Line Length:
4096
The request line is the first line in a request. It consists of the Method, the URL and the HTTP version. The maximum Request Line Length must be approximately the same as the Maximum URL Length.
Max URL Length:
4096
The maximum allowable URL length includes query strings, as these are considered to be part of the URL. Recommended: 4096
Max Query Length:
42
Defines the maximum allowable length for the query string portion of the URL. Recommended: 4096
Max Number of Cookies:
40
All cookies can be in a single "Cookie:" header (specified as name=value and separated by ;). This setting limits the total number of cookies in all headers put together. Recommended: 40
Max Cookie Name Length:
64
The maximum allowable length for a cookie name.
Recommended: 64
Max Cookie Value Length:
4096
The maximum allowable length for an individual cookie value. Recommended: 4096
Max Number of Headers:
20
Defines the maximum number of headers in a request. The header count is inclusive of any Cookie: header in the request. Recommended: 20
Max Header Name Length:
32
Specifies the maximum allowable length for a header name.
Recommended: 32
Max Header Value Length:
512
The maximum allowable length for any request header value. This setting does not affect the Cookie header, which is controlled by settings specific to Cookies. Recommended: 512

Figures