URL Profiles

https://campus.barracuda.com/doc/98211710/

The URL Protection and Parameter Protection components determine how Barracuda WAF-as-a-Service detects and blocks attacks in URLs and URL parameters, respectively, across your application. Sometimes, you will want to tune this behavior for a particular page or parameter. You can save these settings as URL Profiles.

URL Profiles are useful in the following two situations:

1. When a vulnerability scanner finds a vulnerability. In this case, you need to *tighten* security rules to block exploit of the vulnerability.
2. When you encounter a false positive; that is, when Barracuda WAF-as-a-Service detects an attack where no attack is actually happening. In this case, you need to *loosen* security rules to allow the legitimate request through.

URL Profiles are most useful when created automatically. For Case 1 above, rules can be created automatically using the Barracuda Vulnerability Remediation Service. For Case 2, rules are created automatically when you click **Mark as False Positive** in a Firewall Log entry.

You can also create URL Profiles manually.

To manually create an application profile:

1. Log into Barracuda WAF-as-a-Service, select **Applications**, and click the application you are profiling.
2. In the left panel, click **App Profiles**.
3. Select the **Root** folder if you would like the profile to apply to your entire application. Otherwise select or create a sub-URL by clicking **Add URL**. Note: if on a datapath prior to v12.0, simply select **Add URL**.
4. In the window pane at right, if **URL Profile** is not present, click **Add New** and select **URL Profile**.
5. In the **Add New URL Profile** window, specify the following information:
    - **URL** – The address of the application for which you are creating a profile. Note: this is not present in datapath versions 12.0 or later.
    - **Status** – Turn **ON** to enforce checks on the application using this profile.
    - **Block Attacks** – Set to **ON** to allow or block requests based on the profile. When set to OFF, the system validates requests against the URL profile and allows them to pass, but the system logs these errors.
    - **Allowed Methods** – Specify which HTTP methods to allow in requests. Disable methods your application does not use.

- **Allowed Content Types** – Specify one or more content types to allow in the POST body for a URL.
- **Allow Query String** – Set to **ON** to allow a query string (the part after the question mark) for this URL.
- **Hidden Parameter Protection** – Select **Forms** or **Forms and URLs** if you want to protect the hidden parameters in forms and, optionally, URLs. Otherwise, select **None**.
  - **Forms** protects 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** – This involves embedding or adding a random token to forms and URLs that make their location unpredictable. These tokens should be tied to single-user sessions, be time sensitive, and should be generated using random number cryptography. This forms a *challenge token* mechanism that is expected by the server with each subsequent client request. If the token is absent or incorrect, the server denies the request. Effectively, this renders the URLs very dynamic, so attackers cannot predict their locations, thus mitigating CSRF. CSRF protection can be turned on at an application level, at a URL level, or at a parameter level.

6. Click **Add**. If you set the **Status** to **ON**, your App Profile is now in effect.