Layer 7 HTTP(S) Services

https://campus.barracuda.com/doc/3539006/

In this article:

Introduction

HTTP or HTTPS traffic can be handled to a varying degree by the Barracuda Load Balancer before it is directed to a web server. The handling differs based on the type of the Service that receives the traffic.

- Choose a Layer 4 - TCP Service type if you want the traffic simply redirected to the web servers and using only client IP based persistence. This requires a two-armed deployment.
- If you only need client IP based persistence but want to use a one-armed deployment, choose a TCP Proxy Service type.
- To take advantage of Layer 7 handling such as directing requests based on content rules, inspecting and modifying HTTP headers, SSL offloading, or persistence based on cookies, choose either Layer 7 - HTTP (for HTTP traffic) or Layer 7 - HTTPS (for HTTPS traffic).

The rest of this section describes the Layer 7 processing options.

Direct HTTP Requests Based on Content Rules

Content rules are used to direct HTTP requests to specific Real Servers associated with a Layer 7 - HTTP(S) Service. This functionality is also known as content switching or URL switching. A content rule includes:

- One or more expressions that specify a pattern in the host, URL or header fields of the request
- The Real Server or Servers that handle the matching request
- The load balancing algorithm used to direct requests to the Real Servers
- Persistence: none, HTTP cookie, HTTP header, URL parameter or client IP address

Use these rules to partition requests to Real Servers that deliver different types of data, such as:

- Content optimized for a mobile device
- Content in a particular language
- Images or video

- Data that is maintained on different servers but you want to make it appear to have come from one source.

Create a content rule by clicking **Rule** next to a Layer 7 - HTTP(S) Service on the **BASIC > Services** page. This option only appears next to a Service that has at least one Real Server associated with it.

Click on the **Edit** icon next to the rule name on the **BASIC > Services** to edit an existing content rule.

You can edit one or more Real Servers from the **BASIC > Services** page to accept only HTTP requests that match a content rule. Requests that fail to match any rule are directed to the Real Servers for the Service that are not configured to exclusively handle requests that match a content rule. For example, a Real Server which only delivers images can be configured to accept only HTTP requests that match a content rule.

Content Rule Execution

There are up to three types of patterns in each content rule: host match, URL match, and extended match. Extended matches are compared to values in the HTTP header.

If there are multiple rules for a Service, the most specific host and URL match will be executed. For example, if a Service has these two rules:

- Rule A - `host www.example.com, URL /images/*`
- Rule B - `host www.example.com, URL /images/*.png`

and if the incoming request is for `www.example.com/images/x.png` then the most specific matching rule, which is Rule B, is executed.

If a rule has the most specific host and URL for a request, any extended match expressions for that rule are evaluated in the order established by the Extended Match Order field. If the request does not match any extended match expression for the rule then the request is considered to have failed to match any rule.

The possible values for the content rules can be found in the online help. A detailed description of the extended match syntax can be found in How to Use Extended Match and Condition Expressions.

Content Rule Caching and Compression

You can enable caching and compression on the data that matches a content rule using the **WEBSITES > HTTP Caching** and the **WEBSITES > HTTP Compression** pages.

Setting Up an HTTP Redirect

HTTP redirect causes all HTTP traffic on the specified port on a virtual IP address to be redirected to another port (usually port 443) on the same virtual IP address, where SSL requests are served. Implementing HTTP redirect requires configuring two Services, an HTTP redirect Service and an HTTPS Service. HTTP requests that are addressed to **http://VIP:HTTP_redirect_Service_port/** are redirected to **https://VIP:HTTPS_Service_port/**.

This is useful when a site supports only HTTPS access. A client may initially access the site using HTTP, which is the default for most browsers if the URI scheme is not entered. The redirect option allows the client to be transparently moved over to the secure site.

To enable the redirect, create an HTTP redirect Service with Service type of **Layer 7 - HTTP**. Edit the Service and enable HTTP redirect. Because this Service only redirects HTTP requests to an HTTPS Service (the one at the redirect port), you cannot add Real Servers. In fact, few configuration options on the **Service Detail** page are relevant, and all of the other options are hidden (and the settings, if any, ignored).

Finally, you need to create a Layer 7 - HTTPS Service for the same VIP address on the redirect port to receive the redirected requests.

Modify HTTP Requests and Responses

You can set up rules to modify HTTP requests and responses that pass through the Barracuda Load Balancer. These rules, which are associated with a Layer 7 - HTTP(S) Service, are listed on the **WEBSITES > URL Rewrites** page.

One HTTP request rewrite rule is created automatically. It sets the X-Forwarded-For header to the IP address of the client. The Real Server can examine the X-Forwarded-For header to discover the true identity of the requestor, rather than using the sending IP address, which is the IP address of the Barracuda Load Balancer.

You can create response rewrite rules to remove server banners or other header or body information which you do not want the clients to see.

The actions which can be performed by the request rewrite rules are:

- Insert Header - Inserts a header in the request.
- Remove Header - Removes the header from the request.

- Rewrite Header - Rewrites the value of the header in the request
- Rewrite URL - Rewrites the request URL to the URL specified in the rule.
- Redirect the URL - Redirects the request to the URL specified in the rule and sends that redirect back to the client.

Only the first three actions are valid for response header rewrite rules. Response body rules allow any text string (content-type must begin with text/) in an outbound HTTP response body to be rewritten.

The online help for the **WEBSITES > URL Rewrites** page lists the syntax for the rules. In addition, a detailed description of the condition expressions, which specify when the rewrite should occur, is found in Extended Match and Condition Expressions.

Rule Execution Order

Content rules are evaluated first on incoming HTTP traffic. The rules on the **WEBSITES > URL Rewrites** page are evaluated second.

Configure Caching

Caching is a process of storing commonly used information in local memory for quick retrieval rather than sending repeated requests to the web server for the same information. This can improve performance (sometimes dramatically) and reliability. It also reduces the resource utilization on the web servers. Caching can store web pages and commonly used objects such as graphics files. Caching provides the following benefits:

- Reduced latency when retrieving web content.
- An overall reduction in bandwidth and server load.
- Automatic identification and replication of site content.

By default, caching is disabled, but you can enable caching on any Layer 7 - HTTP(S) Service or content rule on the **WEBSITES > HTTP Caching** page. For each Service or content rule you can specify a set of parameters that determine what is cached.

Configure Compression

Compression improves the response time for clients accessing the service through dial-up or other slow methods. Enabling this feature compresses web pages that use HTML, JavaScript, Java and other

text-based languages, resulting in a reduction in download time.

By default, compression is disabled, but you can enable compression on any Layer 7 - HTTP(S) Service or content rule on the **WEBSITES > HTTP Compression** page. For each Service or content rule you can specify the content types and minimum response size to be compressed. Barracuda Networks recommends enabling compression for text based content-types like text/plain, text/html, etc.

Host Multiple Domains with one Service

Hosting multiple SSL-enabled sites on a single server usually requires a unique IP address for each domain, but the Barracuda Load Balancer supports three alternative ways to host multiple domains on one Service. This is particularly useful in a virtual hosting scenario, where you may have several domains hosted on a single Real Server, using the same IP address. These methods are:

- Server Name Indication (SNI)
- Wildcard certificates
- Subject Alternative Name (SAN) certificates

Server Name Indication (SNI)

SNI extends the SSL/TLS protocol to solve the issue of hosting multiple domains on the same IP address. If each domain has a distinct SSL certificate, there needs to be a way for the Real Server to select the proper certificate for a particular domain. The virtual domain information is sent as part of the SSL/TLS negotiation between the client and server. Clients supporting this extension send the domain name when initializing a secure SSL session. The server side component will look at the domain name and send the corresponding certificate to the client.

For SNI to work properly, both the client browser and the web servers must support the SNI extension. SNI is already supported on most major browser platforms, and on both Apache and IIS.

With SNI, you can use the Barracuda Load Balancer to assign any number and any type of certificates (single, wildcard or SAN) to a single Barracuda Load Balancer Service. SNI support applies only to Services with type Layer 7 - HTTPS. To enable SNI, edit the Service and change the setting on the **Service Detail** page. On the same page, you can enter multiple domain names and associate a certificate with each one. Client requests for domains that are not associated with any certificate will get the default certificate. You can add as many certificates to the Service as needed.

Wildcard Certificates

Another alternative is to use wildcard certificates. This allows you to use a single certificate for sub-domains within a domain. If you use a wildcard certificate, you only have to set up a single Service on the Barracuda Load Balancer to serve multiple sub-domains. For example, you can configure a single

Layer 7 - HTTPS Service using a wildcard certificate, such as *.example.com, for **https://sales.example.com** or **https://support.example.com**.

On the negative side, wildcard certificates:

- Are more expensive (typically 3-5x more expensive than single domain certificates).
- Cannot support multi-domains that are distinct from each other, such as **www.mysite1.com** and **www.mysite2.com**. Multi-domain support is especially critical for web hosting providers or Managed Service Providers (MSP) who may have multiple virtual web servers representing numerous domains on a single physical server using a single IP address.
- Cannot secure host names on different base domains, such as **www.mysite1.com** and **www.mysite1.net**.

Subject Alternative Name (SAN) Certificates

SAN certificates fall between a wildcard certificate and a single domain certificate, as each certificate allows you to specify a list of domain names to be protected. A SAN certificate for **www.example.com** could have the domains **www.examples.net** and **www.ex.com** listed as alternative names for the same Service. On the negative side, SAN certificates are more expensive than single domain certificates and are often limited to 3-5 domains. More importantly, not all Certificate Authorities sell SAN enabled certificates.

Related Articles

- Services
- Route-Path Deployment
- Two-Armed Route-Path Deployment
- One-Armed Route-Path Deployment