How to Configure Neighbor Proxies

https://campus.barracuda.com/doc/96026350/

For the HTTP Proxy service, you can configure the proxy server to treat adjacent proxies as parents or siblings. For the neighbor proxies, you can configure authentication and caching.

Configure a Neighbor Proxy

If the proxy server will be surrounded by multiple adjacent neighbor caches, see Multiple Adjacent Proxies before you configure the neighbors.

1. Go to **CONFIGURATION > Configuration Tree > Box > Assigned Services > HTTP Proxy > HTTP Proxy Settings**.
2. In the left menu, select **IP Configuration**.
3. Click **Lock**.
4. In the **Neighbor Settings** table, click **+** to add an entry for the neighboring proxy.
   1. Enter a **Name** for the proxy and click **OK**.

      Because you can use this name in the ACL for the proxy server, do not use it when naming an entry in the **ACL Entries** table.
   2. In the **Neighbor Settings** window, configure the settings for the neighbor. For more details about these settings, see Neighbor Settings.
   3. Click **OK**.
5. Click **Send Changes** and **Activate**.

Multiple Adjacent Proxies

If the proxy server will be surrounded by multiple adjacent neighbor caches, it is important that you correctly configure the caching settings for the neighbors. In particular, the **Cache Priority** setting directly affects the execution of the **Cache Peer Access** and **Domain Restrictions** settings (for more details about these settings, see **Neighbor Settings**). For example, consider the following scenario:

- ProxySrv1 is surrounded by three neighbor caches.
- ProxySrv2, ProxySrv3, and ProxySrv4 are configured as the parents of ProxySrv1.

The aim is to direct all requests with the source IP address of 10.0.8.20 to ProxySrv2. All requests with the destination of *exampledomain.com* should be directed to ProxySrv3. All other requests are to be fetched from the cache of ProxySrv4.
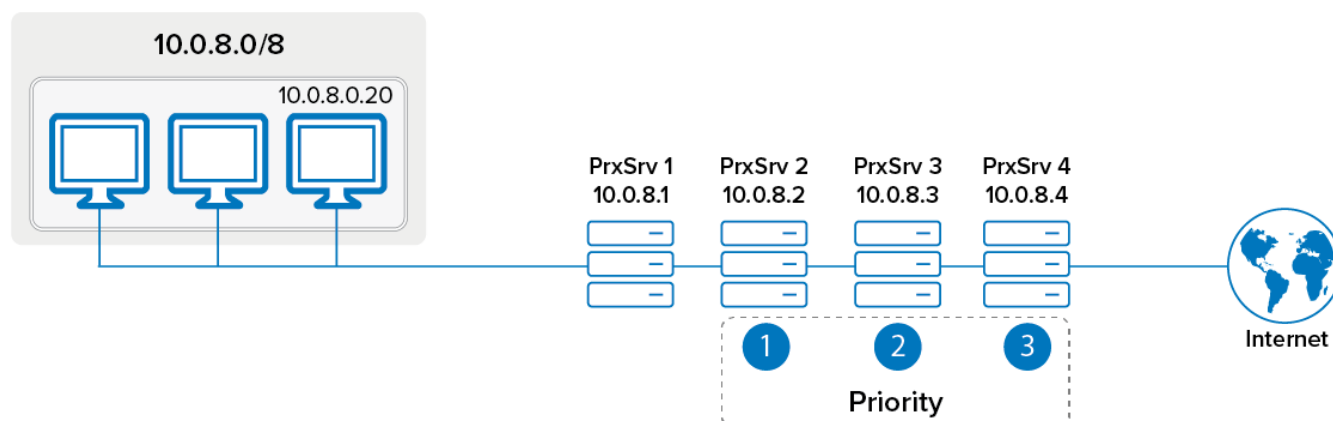
A **Cache Peer Access** filter must be set for ProxySrv2 and a **Domain Restrictions** filter must be set for ProxySrv3. ProxySrv4 is set up without any filters, which means that all requests that do not match the configured filters will be directed to it.

ProxySrv4 is vital for the example setup to work. If it is not present, requests that do not match the configured filters cannot be directed to any neighbor. ProxySrv1 cannot process the requests spontaneously without the appropriate directive.

The neighbor servers are configured with the following settings:

Server	Neighbor Settings
ProxySrv2	• **Name** – ProxySrv2 • **IP/Hostname** – 10.0.8.2 • **Neighbor Type** – **parent** • **Exclusive Parent** – **no** • **Cache Priority** – 1 • **Cache Peer Access** – 10.0.8.20 • **Cache IP Objects** – **no**
ProxySrv3	• **Name** – ProxySrv3 • **IP/Hostname** – 10.0.8.3 • **Neighbor Type** – **parent** • **Exclusive Parent** – **no** • **Cache Priority** – 2 • **Domain Restrictions** – *.exampledomain.com • **Cache Domain Objects** – **no**
ProxySrv4	• **Name** – ProxySrv4 • **IP/Hostname** – 10.0.8.4 • **Neighbor Type** – **parent** • **Exclusive Parent** – **no** • **Cache Priority** – 3

Neighbor Settings

This table provides more detailed descriptions of the settings that you can configure for neighbor proxies:

Setting	Description
Connection Type	The type of settings to use for the neighbor proxy. You can select one of the following types: • **Like-System-Settings** – To use your system settings for the neighbor proxy. • **Explicit** – To define explicit settings for the neighbor proxy.
IP/Hostname	The hostname (FQDN) or IP address (IPv4 or IPv6) of the neighboring proxy server.
Neighbor Type	The relationship to the neighboring proxy server. You can select: • **parent** • **sibling** In a sibling relationship, a peer may only request objects that are already held in the cache. A sibling cannot forward cache misses on behalf of the peer.
Exclusive Parent	For a neighboring parent proxy server, select *yes* if it should handle all forwarding requests. This setting is recommended if the parent proxy is a virus scanning proxy server.
Proxy Port	The port on which the neighbor server listens for incoming HTTP requests. By default, port 3128 is used.
ICP Port	The port on which the neighbor server listens for incoming ICP connections. By default, port 3130 is used. To configure a neighbor cache that does not use ICP, enable the UDP echo port on it and enter 7 for the ICP port. To disable this port for neighbors that do not support ICP queries, enter 0 in the **ICP Port** field and then enter *no-query* in the **Options** table.
Cache Priority	The cache priority for the server. This setting is mandatory. Numbers with a lower value grant higher priority to the server. If only one neighbor cache exists, you can enter any value for the cache priority; the priority is ignored. You cannot enter *0* for the **Cache Priority**.
Authentication	The authentication mechanism from the proxy to its neighbor. You can select: • **PASS (=log in)** – For authentication against an upstream proxy (parent). To combine this with proxy_auth, both proxies must share the same user database because HTTP only allows one proxy login. The *PASS (=log in)* setting exposes your user's proxy password to the parent. • **noPASS** – Use with a personal or workgroup proxy when the parent requires proxy authentication. In the following **User** and **Password** fields, you must also specify the login credentials.

Options	Additional options for the specified parent proxy. For example, you can enter options such as *proxy-only*, *weight=n*, *ttl=n*, *no-query*, *default*, *round-robin*, *multicast-responder*, *closest-only*, *no-digest*, *no-netdb-exchange*, *connect-timeout=nn*, *digest-url=url*, and *allow-miss*. For more information, see the Squid documentation.
URL Fetching	If a page should be fetched directly from its origin server, add the complete URL or a list of words from the URL of the page. Before communicating with any of the cache peers, Squid tries fetching the requested URL directly from the server. If Squid cannot find the page, it tries to establish a connection to the configured parent caches. If you do not specify which protocol should be used to fetch the URL (for example, *www.barracuda.com* or **barracuda**), Squid tries to fetch the page via HTTP and FTP. If virus scanning and FTP scanning are activated for URLs that are fetched via FTP, you must specify the FTP protocol (for example, *ftp://www.barracuda.com* and *ftp://*barracuda**). Otherwise, the data stream is forwarded without virus scanning. It is recommended that you include dynamic pages in this tag (such as jsp, asp, and php). These **URL Fetching** settings are shared by all neighbor proxies, even if you do not enter them in other configuration sections.
Cache Direct Objects	To cache URLs that are directly fetched with the **URL Fetching** settings, select **yes**.
Domain Restrictions	In this table, add the domains of the neighbor caches to be queried. Use the following syntax: • `.domainame.tld` • `.subdomain.domainame.tld` • `*.domainname.tld` … If a domain should be not queried from the cache, add a quotation mark ("!") before its name. For example: • `!.domainname.com` … Cache hosts that are configured without domain restrictions will be queried for all domains.
Cache Domain Objects	To cache URLs that are fetched with the **Domain Restrictions** setting, select **yes**.
Cache Peer Access	In this table, add IP addresses and IP address ranges that must be directed to a specific neighbor cache. If restrictions are not configured, the cache will be queried for all requests.
Cache IP Objects	To cache requests originating from the IP addresses that are entered in the **Cache Peer Access** table, select **yes**.

Figures

1. ffw_neighbor_proxy.png