Reverse Proxy Setup Guide

It is possible to run PowerFolder Server behind a third party web server. There are several reasons, why you might want to use such a setup:

- Privileged Ports on Linux - Most Linux systems doesn't allow normal users to run services, which bind to a port below the port number 1024. To have your PowerFolder Server web be reachable on the standard web ports 80 or 443 you need a web server with proxy support.
- Simple Proxying - You have an existing website and want to integrate PowerFolder Server in your virtual host (e.g. http://www.example.com/powerfolder)
- SSL-encrypted HTTP sessions - Sessions to the web interface are by default not encrypted. PowerFolder Server supports SSL-encrypted web access internally, however you might want to get this done by a third party web server like Apache or Nginx

We provide several guides here to integrate PowerFolder Server with third party web servers:

Apache Proxy and PowerFolder Server for SSL Encryption

In this article we are showing a configuration example for running using PowerFolder Server with an Apache Proxy for a SSL-encrypted web interface sessions.

Requirements

The requirements below are necessary for the setup:

- Apache 2.2 and higher with mod_proxy,mod_rewrite and mod_ssl enabled.
- A valid, officially signed SSL certificate. ⚠️ PowerFolder Clients will NOT work with invalid or self-signed certificates.

Notes for Windows users

Users installing Apache on Windows, might want to download the Apache Binaries from Apache Lounge. The installation is easy:

1. Place the `Apache24` directory, extracted from the `.zip` file, at `C:\Apache24`.
2. To install it as a service, go to `C:\Apache24\bin` and execute the following command: `httpd.exe -k install`
3. Uncomment (remove the # in front) the following lines in the `C:\Apache24\conf\httpd.conf` file:

```
LoadModule proxy_module modules/mod_proxy.so
LoadModule rewrite_module modules/mod_rewrite.so
LoadModule ssl_module modules/mod_ssl.so
Include conf/extra/httpd-vhosts.conf
```

4. Modify the `C:\Apache24\conf\extra\httpd-vhosts.conf` file as described later in this article. Make sure you modify the right paths to your certificate files, hostnames and the IP address of your local network interface (where the actual PowerFolder Server web interface is listening).
5. Restart the Apache Windows service.

For troubleshooting you can also use the following command to start the Apache Windows service manually: `C:\Apache24\bin\httpd.exe -k start`

In the below configuration example for Apache, we use several placeholders which need to be changed to match your installation:

- Web interface URL: https://powerfolder.example.com
- PowerFolder Server HTTP Port: 8080
- Apache HTTP Port: 80
- Apache HTTPS Port: 443
- Server IP: 10.0.0.1

PowerFolder Server Web Configuration

For this scenario we need to change settings in the server preferences:

- Set the ***Web Base URL*** under **Preferences > Network > Server URLs**:

```
https://powerfolder.example.com
```

- Set the ***Web Tunnel URL*** under **Preferences > Network > Server URLs**:

```
http://powerfolder.example.com/rpc
```

⚠️ Please note that the URL must use HTTP not HTTPS, since the traffic posted against that URL will be encrypted by the PowerFolder internal protocol.

- Set the ***HTTPS/SSL port*** under **Preferences > Network > Hostname and Ports**:

```
"-1"
```

✅ After changing those settings, please restart PowerFolder Server.

Apache Configuration

IMPORTANT: On some systems the configuration entry `ProxyRequests` is set to `On` by default. Please check the Apache configuration file `/etc/apache2/mods-available/proxy.conf` and change `ProxyRequests On` to `ProxyRequests Off`. Otherwise the Apache server can be used as open proxy by others.

More information: http://httpd.apache.org/docs/2.2/mod/mod_proxy.html#proxyrequests

Configure a virtual host within Apache, which responds to requests to http://powerfolder.example.com and https://powerfolder.example.com and forwards the requests to the web port of PowerFolder Server:

```
<VirtualHost *:80>
       ServerAdmin hostmaster@example.com
       ServerName powerfolder.example.com

       RewriteEngine on
       RewriteCond             %{SERVER_PORT}  !=443
       RewriteCond             %{REQUEST_URI}  !^/rpc
       RewriteRule             ^.*$            https://%{SERVER_NAME}%
{REQUEST_URI} [NC,R=301,L]



       ProxyPass               /rpc                    http://10.0.0.1:
8080/rpc     nocanon
       ProxyPassReverse        /rpc                    http://10.0.0.1:
8080/rpc

</VirtualHost>


Listen 10.0.0.1:443
<VirtualHost 10.0.0.1:443>
       ServerAdmin hostmaster@example.com
       ServerName powerfolder.example.com

       SSLEngine  On
       SSLCACertificateFile   /etc/apache2/ssl/powerfolder.example.com.
ca-bundle.crt
       SSLCertificateFile     /etc/apache2/ssl/powerfolder.example.com.
crt
       SSLCertificateKeyFile  /etc/apache2/ssl/powerfolder.example.com.
key
       SSLCipherSuite ALL:-ADH:+HIGH:+MEDIUM:-LOW:-SSLv2:-EXP


       ProxyPass               /rpc                    http://10.0.0.1:
8080/rpc     nocanon
       ProxyPassReverse        /rpc                    http://10.0.0.1:
8080/rpc


       ProxyPass               /rpc                    !


       ProxyPass               /                       http://10.0.0.1:
8080/        nocanon
       ProxyPassReverse        /                       http://10.0.0.1:
8080/
</VirtualHost>
```

ℹ️ We exclude the /rpc URL part from the SSL-encryption, because this URL is used for the PowerFolder Clients to tunnel their traffic by using the HTTP POST method, in case they are behind a firewall and can't establish a direct connection to PowerFolder Server. Since the PowerFolder data traffic is encrypted anyway by a PowerFolder internal protocol, we don't need encryption here. It would just slow down the connection.

⚠️ Please note: When it is required by the SSL certificate authority to use an intermediate certificate, it has to be loaded with the `SSLCACertificateFile` configuration entry. If such an intermediate certificate is NOT required, you can simply drop that line.

Setting up Nginx as SSL proxy

In this article we are showing a configuration example for running using PowerFolder Server with a Nginx Proxy for a SSL-encrypted web interface sessions.

Requirements

The requirements below are necessary for the setup:

- Nginx 1.1 and higher.
- A valid, officially signed SSL certificate. ⚠️ PowerFolder Clients will NOT work with invalid or self-signed certificates.

In the below configuration example for Nginx, we use several placeholders which need to be changed to match your installation:

- Web interface URL: https://powerfolder.example.com
- PowerFolder Server HTTP Port: 8080
- Nginx HTTP Port: 80
- Nginx HTTPS Port: 443
- Server IP: 10.0.0.1

PowerFolder Server Web Configuration

For this scenario we need to change settings in the server preferences:

- Set the ***Web Base URL*** under **Preferences > Network > Server URLs**:

```
https://powerfolder.example.com
```

- Set the ***Web Tunnel URL*** under **Preferences > Network > Server URLs**:

```
http://powerfolder.example.com/rpc
```

⚠️ Please note that the URL should not use HTTP not HTTPS, since the traffic posted against that URL will be encrypted by the PowerFolder internal protocol.

- Set the ***HTTPS/SSL port*** under **Preferences > Network > Hostname and Ports**:

```
' -1 '
```

✅ After changing those settings, please restart PowerFolder Server.

Nginx Configuration

Configure a virtual host within Nginx, which responds to requests to http://powerfolder.example.com and https://powerfolder.example.com and forwards the requests to the web port of PowerFolder Server:

```
        server {
                listen 10.0.0.1;
                server_name powerfolder.example.com;
                location / {
                        rewrite        ^
https://$server_name$request_uri? permanent;
                }
                location /rpc {
                        proxy_pass http://10.0.0.1:8080/rpc;
                }
        }

        server {
                listen 10.0.0.1:443;
                server_name powerfolder.example.com;
                ssl on;
                ssl_certificate /etc/nginx/ssl/powerfolder.example.com.
chained.crt;
                ssl_certificate_key /etc/nginx/ssl/powerfolder.example.
com.key;

                location / {
                        proxy_pass http://10.0.0.1:8080;
                }
        }
```

ℹ️ We exclude the /rpc URL part from the SSL-encryption, because this URL is used for the PowerFolder Clients to tunnel their traffic by using the HTTP POST method, in case they are behind a firewall and can't establish a direct connection to PowerFolder Server. Since the PowerFolder data traffic is encrypted anyway by a PowerFolder internal protocol, we don't need encryption here. It would just slow down the connection.

⚠️ Please note: When it is required by the SSL certificate authority to use an intermediate certificate, a chained certificate has to be created. Simply create a new text file, copy & paste the intermediate certificate into it and right after it the actual certificate for your domain. In our example we called the file `powerfolder.example.com.chained.crt`.

Using nginx with cache (experimental)

It is possible to activate cache in nginx to reduce load of your PowerFolder Server. Static content will get cached by nginx and get delivered directly to the browser. The caching directory can get freely chosen. Since this might contain many data it should have sufficient disk space!

Please ensure to clean your caching directory after every server update to ensure, that no old cached content get delivered to your users.

https://www.nginx.com/blog/nginx-caching-guide/

```
                proxy_cache_path </etc/nginx/cache> levels=1:2
keys_zone=pf_cache:10m max_size=10g inactive=10m use_temp_path=off;
                server {
                listen 10.0.0.1;
                server_name powerfolder.example.com;
                location / {
                        rewrite        ^
https://$server_name$request_uri? permanent;
                }
                location /rpc {
                        proxy_pass http://10.0.0.1:8080/rpc;
                }
        }

        server {
                listen 10.0.0.1:443;
                server_name powerfolder.example.com;
                ssl on;
                ssl_certificate /etc/nginx/ssl/powerfolder.example.com.
chained.crt;
                ssl_certificate_key /etc/nginx/ssl/powerfolder.example.
com.key;

                location / {
                        proxy_pass http://10.0.0.1:8080;
                                                proxy_buffering on;
                                                proxy_cache pf_cache;
                                                proxy_cache_valid 200
1d;
                                                proxy_cache_key
$proxy_host$request_uri$cookie_JSESSIONID;
                                }
        }
```

Overview: