pveproxy(8)

Proxmox Server Solutions GmbH
<support@proxmox.com>
version 8.4.0, Wed Apr 9 08:00:00 CEST 2025

NAME

pveproxy - PVE API Proxy Daemon

SYNOPSIS

pveproxy `<COMMAND> [ARGS] [OPTIONS]`

pveproxy help `[OPTIONS]`

Get help about specified command.

`--extra-args <array>`
    Shows help for a specific command

`--verbose <boolean>`
    Verbose output format.

pveproxy restart

Restart the daemon (or start if not running).

pveproxy start `[OPTIONS]`

Start the daemon.

`--debug <boolean>` (*default* = `0`)
    Debug mode - stay in foreground

pveproxy status

Get daemon status.

pveproxy stop

Stop the daemon.

DESCRIPTION

This daemon exposes the whole Proxmox VE API on TCP port 8006 using HTTPS. It runs as user `www-data` and has very limited permissions. Operation requiring more permissions are forwarded to the local `pvedaemon`.

Requests targeted for other nodes are automatically forwarded to those nodes. This means that you can manage your whole cluster by connecting to a single Proxmox VE node.

Host based Access Control

It is possible to configure “apache2”-like access control lists. Values are read from file `/etc/default/pveproxy`. For example:

```
ALLOW_FROM="10.0.0.1-10.0.0.5,192.168.0.0/22"
DENY_FROM="all"
POLICY="allow"
```

IP addresses can be specified using any syntax understood by Net::IP. The name all is an alias for 0/0 and ::/0 (meaning all IPv4 and IPv6 addresses).

The default policy is allow.

Match	POLICY=deny	POLICY=allow
Match Allow only	allow	allow
Match Deny only	deny	deny
No match	deny	allow
Match Both Allow & Deny	deny	allow

Listening IP Address

By default the pveproxy and spiceproxy daemons listen on the wildcard address and accept connections from both IPv4 and IPv6 clients.

By setting LISTEN_IP in /etc/default/pveproxy you can control to which IP address the pveproxy and spiceproxy daemons bind. The IP-address needs to be configured on the system.

Setting the sysctl net.ipv6.bindv6only to the non-default 1 will cause the daemons to only accept connection from IPv6 clients, while usually also causing lots of other issues. If you set this configuration we recommend to either remove the sysctl setting, or set the LISTEN_IP to 0.0.0.0 (which will only allow IPv4 clients).

LISTEN_IP can be used to only to restricting the socket to an internal interface and thus have less exposure to the public internet, for example:

```
LISTEN_IP="192.0.2.1"
```

Similarly, you can also set an IPv6 address:

```
LISTEN_IP="2001:db8:85a3::1"
```

Note that if you want to specify a link-local IPv6 address, you need to provide the interface name itself. For example:

```
LISTEN_IP="fe80::c463:8cff:feb9:6a4e%vmbr0"
```

The nodes in a cluster need access to pveproxy for communication, possibly on different sub-nets. It is **not recommended** to set LISTEN_IP on clustered systems.

To apply the change you need to either reboot your node or fully restart the pveproxy and spiceproxy service:

```
systemctl restart pveproxy.service spiceproxy.service
```

Unlike reload, a restart of the pveproxy service can interrupt some long-running worker processes, for example a running console or shell from a virtual guest. So, please use a maintenance window to bring this change in effect.

SSL Cipher Suite

You can define the cipher list in /etc/default/pveproxy via the CIPHERS (TLS ⇐ 1.2) and CIPHERSUITES (TLS >= 1.3) keys. For example

```
CIPHERS="ECDHE-ECDSA-AES256-GCM-SHA384:ECDHE-RSA-AES256-GCM-SHA384:ECDHE-ECDSA-
CHACHA20-POLY1305:ECDHE-RSA-CHACHA20-POLY1305:ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-
RSA-AES128-GCM-SHA256:ECDHE-ECDSA-AES256-SHA384:ECDHE-RSA-AES256-SHA384:ECDHE-
ECDSA-AES128-SHA256:ECDHE-RSA-AES128-SHA256"
CIPHERSUITES="TLS_AES_256_GCM_SHA384:TLS_CHACHA20_POLY1305_SHA256:TLS_AES_128_GCM_SHA256
```

Above is the default. See the ciphers(1) man page from the openssl package for a list of all available options.

Additionally, you can set the client to choose the cipher used in /etc/default/pveproxy (default is the first cipher in the list available to both client and pveproxy):

```
HONOR_CIPHER_ORDER=0
```

Supported TLS versions

The insecure SSL versions 2 and 3 are unconditionally disabled for pveproxy. TLS versions below 1.1 are disabled by default on recent OpenSSL versions, which is honored by pveproxy (see /etc/ssl/openssl.cnf).

To disable TLS version 1.2 or 1.3, set the following in /etc/default/pveproxy:

```
DISABLE_TLS_1_2=1
```

or, respectively:

```
DISABLE_TLS_1_3=1
```

> Unless there is a specific reason to do so, it is not recommended to manually adjust the supported TLS versions.

Diffie-Hellman Parameters

You can define the used Diffie-Hellman parameters in /etc/default/pveproxy by setting DHPARAMS to the path of a file containing DH parameters in PEM format, for example

```
DHPARAMS="/path/to/dhparams.pem"
```

If this option is not set, the built-in skip2048 parameters will be used.

> DH parameters are only used if a cipher suite utilizing the DH key exchange algorithm is negotiated.

Alternative HTTPS certificate

You can change the certificate used to an external one or to one obtained via ACME.

pveproxy uses /etc/pve/local/pveproxy-ssl.pem and /etc/pve/local/pveproxy-ssl.key, if present, and falls back to /etc/pve/local/pve-ssl.pem and /etc/pve/local/pve-ssl.key. The private key may not use a passphrase.

It is possible to override the location of the certificate private key /etc/pve/local/pveproxy-ssl.key by setting TLS_KEY_FILE in /etc/default/pveproxy, for example:

```
TLS_KEY_FILE="/secrets/pveproxy.key"
```

> The included ACME integration does not honor this setting.

See the Host System Administration chapter of the documentation for details.

Response Compression

By default `pveproxy` uses gzip HTTP-level compression for compressible content, if the client supports it. This can disabled in `/etc/default/pveproxy`

```
COMPRESSION=0
```

Real Client IP Logging

By default, `pveproxy` logs the IP address of the client that sent the request. In cases where a proxy server is in front of `pveproxy`, it may be desirable to log the IP of the client making the request instead of the proxy IP.

To enable processing of a HTTP header set by the proxy for logging purposes, set `PROXY_REAL_IP_HEADER` to the name of the header to retrieve the client IP from. For example:

```
PROXY_REAL_IP_HEADER="X-Forwarded-For"
```

Any invalid values passed in this header will be ignored.

The default behavior is log the value in this header on all incoming requests. To define a list of proxy servers that should be trusted to set the above HTTP header, set `PROXY_REAL_IP_ALLOW_FROM`, for example:

```
PROXY_REAL_IP_ALLOW_FROM="192.168.0.2"
```

The `PROXY_REAL_IP_ALLOW_FROM` setting also supports values similar to the `ALLOW_FROM` and `DENY_FROM` settings.

IP addresses can be specified using any syntax understood by `Net::IP`. The name `all` is an alias for `0/0` and `::/0` (meaning all IPv4 and IPv6 addresses).

Copyright and Disclaimer

Version 8.4.0
Last updated Wed Apr 9 08:00:00 CEST 2025