firewalld

A service daemon with D-Bus interface

firewall-offline-cmd

Name

firewall-offline-cmd — firewalld offline command line client

Synopsis

`firewall-offline-cmd` [OPTIONS...]

Description

firewall-offline-cmd is an offline command line client of the firewalld daemon. It should be used only if the firewalld service is not running. For example to migrate from system-config-firewall/lokkit or in the install environment to configure firewall settings with kickstart.

Some lokkit options can not be automatically converted for firewalld, they will result in an error or warning message. This tool tries to convert as much as possible, but there are limitations for example with custom rules, modules and masquerading.

Check the firewall configuration after using this tool.

Options

If no options are given, configuration from **/etc/sysconfig/system-config-firewall** will be migrated.

Sequence options are the options that can be specified multiple times, the exit code is 0 if there is at least one item that succeeded. The `ALREADY_ENABLED` (11), `NOT_ENABLED` (12) and also `ZONE_ALREADY_SET` (16) errors are treated as succeeded. If there are issues while parsing the items, then these are treated as warnings and will not change the result as long as there is a succeeded one. Without any succeeded item, the exit code will depend on the error codes. If there is exactly one error code, then this is used. If there are more than one then `UNKNOWN_ERROR` (254) will be used.

The following options are supported:

General Options

`-h`, `--help`

Prints a short help text and exits.

`-V`, `--version`
Prints the version string of firewalld and exits.

`-q`, `--quiet`
Do not print status messages.

`--default-config`
Path to firewalld default configuration. This usually defaults to `/usr/lib/firewalld`.

`--system-config`
Path to firewalld system (user) configuration. This usually defaults to `/etc/firewalld`.

Status Options

`--enabled`
Enable the firewall. This option is a default option and will activate the firewall if not already enabled as long as the option `--disabled` is not given.

`--disabled`
Disable the firewall by disabling the firewalld service.

`--check-config`
Run checks on the permanent (default and system) configuration. This includes XML validity and semantics.

This is may be used with `--system-config` to check the validity of handwritten configuration files before copying them to the standard location.

Lokkit Compatibility Options

These options are nearly identical to the options of **lokkit**.

`--migrate-system-config-firewall=`*`file`*
Migrate system-config-firewall configuration from the given file. No further

`--addmodule`=*`module`*
This option will result in a warning message and will be ignored.

Handling of netfilter helpers has been merged into services completely. Adding or removing netfilter helpers outside of services is therefore not needed anymore. For more information on handling netfilter helpers in services, please have a look at firewalld.zone(5).

`--removemodule`
This option will result in a warning message and will be ignored.

Handling of netfilter helpers has been merged into services completely. Adding or removing netfilter helpers outside of services is therefore not needed anymore. For more information on handling netfilter helpers in services, please have a look at firewalld.zone(5).

`--remove-service`=*service*

Remove a service from the default zone. This option can be specified multiple times.

The service is one of the firewalld provided services. To get a list of the supported services, use **firewall-cmd --get-services**.

`-s` *service*, `--service`=*service*

Add a service to the default zone. This option can be specified multiple times.

The service is one of the firewalld provided services. To get a list of the supported services, use **firewall-cmd --get-services**.

`-p` *portid*[-*portid*]:*protocol*, `--port`=*portid*[-*portid*]:*protocol*

Add the port to the default zone. This option can be specified multiple times.

The port can either be a single port number or a port range *portid*-*portid*. The protocol can either be `tcp`, `udp`, `sctp` or `dccp`.

`-t` *interface*, `--trust`=*interface*

This option will result in a warning message.

Mark an interface as trusted. This option can be specified multiple times. The interface will be bound to the trusted zone.

If the interface is used in a NetworkManager managed connection or if there is an ifcfg file for this interface, the zone will be changed to the zone defined in the configuration as soon as it gets activated. To change the zone of a connection use **nm-connection-editor** and set the zone to trusted, for an ifcfg file, use an editor and add "ZONE=trusted". If the zone is not defined in the ifcfg file, the firewalld default zone will be used.

`-m` *interface*, `--masq`=*interface*

This option will result in a warning message.

Masquerading will be enabled in the default zone. The interface argument will be ignored. This is for *IPv4* only.

`--custom-rules`=[*type*:][*table*:]*filename*

This option will result in a warning message and will be ignored.

Custom rule files are not supported by firewalld.

`--forward-port`=if=*interface*:port=*port*:proto=*protocol*[:toport=*destination port*:][:toaddr=*destination address*]

This option will result in a warning message.

Add the *IPv4* forward port in the default zone. This option can be specified multiple times.

The port can either be a single port number `portid` or a port range `portid`-`portid`. The protocol can either be `tcp`, `udp`, `sctp` or `dccp`. The destination address is an IP address.

`--block-icmp`=`icmptype`

This option will result in a warning message.

Add an ICMP block for `icmptype` in the default zone. This option can be specified multiple times.

The `icmptype` is the one of the icmp types firewalld supports. To get a listing of supported icmp types: **firewall-cmd --get-icmptypes**

Log Denied Options

`--get-log-denied`

Print the log denied setting.

`--set-log-denied`=`value`

Add logging rules right before reject and drop rules in the INPUT, FORWARD and OUTPUT chains for the default rules and also final reject and drop rules in zones for the configured link-layer packet type. The possible values are: `all`, `unicast`, `broadcast`, `multicast` and `off`. The default setting is `off`, which disables the logging.

This is a runtime and permanent change and will also reload the firewall to be able to add the logging rules.

Zone Options

`--get-default-zone`

Print default zone for connections and interfaces.

`--set-default-zone`=`zone`

Set default zone for connections and interfaces where no zone has been selected. Setting the default zone changes the zone for the connections or interfaces, that are using the default zone.

`--get-zones`

Print predefined zones as a space separated list.

`--get-services`

Print predefined services as a space separated list.

`--get-icmptypes`

Print predefined icmptypes as a space separated list.

`--get-zone-of-interface`=*`interface`*

Print the name of the zone the *`interface`* is bound to or *no zone*.

`--get-zone-of-source`=*`source`*[/*`mask`*]|*`MAC`*|ipset:*`ipset`*

Print the name of the zone the source is bound to or *no zone*.

`--info-zone=`*`zone`*

Print information about the zone *`zone`*. The output format is:

```
zone
  interfaces: interface1 ..
  sources: source1 ..
  services: service1 ..
  ports: port1 ..
  protocols: protocol1 ..
  forward-ports:
        forward-port1
        ..
  source-ports: source-port1 ..
  icmp-blocks: icmp-type1 ..
  rich rules:
        rich-rule1
        ..
```

`--list-all-zones`

List everything added for or enabled in all zones. The output format is:

```
zone1
  interfaces: interface1 ..
  sources: source1 ..
  services: service1 ..
  ports: port1 ..
  protocols: protocol1 ..
  forward-ports:
        forward-port1
        ..
  source-ports: source-port1 ..
  icmp-blocks: icmp-type1 ..
  rich rules:
        rich-rule1
        ..
..
```

`--new-zone`=*`zone`*

Add a new permanent zone.

Zone names must be alphanumeric and may additionally include characters: '_' and '-'.

`--new-zone-from-file`=*`filename`* [`--name`=*`zone`*]
Add a new permanent zone from a prepared zone file with an optional name override.

`--path-zone=`*`zone`*
Print path of the zone configuration file.

`--delete-zone`=*`zone`*
Delete an existing permanent zone.

Policy Options

`--get-policies`
Print predefined policies as a space separated list.

`--info-policy`=*`policy`*
Print information about the policy *`policy`*.

`--list-all-policies`
List everything added for or enabled in all policies.

`--new-policy`=*`policy`*
Add a new permanent policy.

Policy names must be alphanumeric and may additionally include characters: '_' and '-'.

`--new-policy-from-file`=*`filename`* [`--name`=*`policy`*]
Add a new permanent policy from a prepared policy file with an optional name override.

`--path-policy`=*`policy`*
Print path of the policy configuration file.

`--delete-policy`=*`policy`*
Delete an existing permanent policy.

`--load-policy-defaults`=*`policy`*
Load the shipped defaults for a policy. Only applies to policies shipped with firewalld. Does not apply to user defined policies.

Options to Adapt and Query Zones and Policies

Options in this section affect only one particular zone or policy. If used with `--zone`=*`zone`* or `--policy`=*`policy`* option, they affect the specified zone or policy. If both options are omitted, they affect default zone (see `--get-default-zone`).

[`--zone`=*zone*] [`--policy`=*policy*] `--list-all`
List everything added or enabled.

[`--zone`=*zone*] [`--policy`=*policy*] `--get-target`
Get the target.

[`--zone`=*zone*] [`--policy`=*policy*] `--set-target`=*target*
Set the target.

For zones *target* is one of: `default`, `ACCEPT`, `DROP`, `REJECT`

For policies *target* is one of: `CONTINUE`, `ACCEPT`, `DROP`, `REJECT`

`default` is similar to `REJECT`, but it implicitly allows ICMP packets.

[`--zone`=*zone*] [`--policy`=*policy*] `--set-description`=*description*
Set description.

[`--zone`=*zone*] [`--policy`=*policy*] `--get-description`
Print description.

[`--zone`=*zone*] [`--policy`=*policy*] `--set-short`=*description*
Set short description.

[`--zone`=*zone*] [`--policy`=*policy*] `--get-short`
Print short description.

[`--zone`=*zone*] [`--policy`=*policy*] `--list-services`
List services added as a space separated list.

[`--zone`=*zone*] [`--policy`=*policy*] `--add-service`=*service*
Add a service. This option can be specified multiple times.

The service is one of the firewalld provided services. To get a list of the supported services, use **firewall-cmd --get-services**.

Note: Some services define connection tracking helpers. Helpers that may operate in client mode (e.g. tftp) must be added to an outbound policy instead of a zone to take effect for clients. Otherwise the helper will not be applied to the outbound traffic. The related traffic, as defined by the connection tracking helper, on the return path (ingress) will be allowed by the stateful firewall rules.

An example of an outbound policy for connection tracking helpers:

```
# firewall-cmd --new-policy clientConntrack
# firewall-cmd --policy clientConntrack --add-ingress-zone HOST
```

```
# firewall-cmd --policy clientConntrack --add-egress-zone ANY
# firewall-cmd --policy clientConntrack --add-service tftp
```

[`--zone` = *`zone`*] `--remove-service-from-zone` = *`service`*
Remove a service from *`zone`*. This option can be specified multiple times. If zone is omitted, default zone will be used.

[`--policy` = *`policy`*] `--remove-service-from-policy` = *`service`*
Remove a service from *`policy`*. This option can be specified multiple times.

[`--zone` = *`zone`*] [`--policy` = *`policy`*] `--query-service` = *`service`*
Return whether *`service`* has been added. Returns 0 if true, 1 otherwise.

[`--zone` = *`zone`*] [`--policy` = *`policy`*] `--list-ports`
List ports added as a space separated list. A port is of the form *`portid`* [- *`portid`*]/ *`protocol`*, it can be either a port and protocol pair or a port range with a protocol.

[`--zone` = *`zone`*] [`--policy` = *`policy`*] `--add-port` = *`portid`* [- *`portid`*]/ *`protocol`*
Add the port. This option can be specified multiple times.

The port can either be a single port number or a port range *`portid`* - *`portid`*. The protocol can either be `tcp`, `udp`, `sctp` or `dccp`.

[`--zone` = *`zone`*] [`--policy` = *`policy`*] `--remove-port` = *`portid`* [- *`portid`*]/ *`protocol`*
Remove the port. This option can be specified multiple times.

[`--zone` = *`zone`*] [`--policy` = *`policy`*] `--query-port` = *`portid`* [- *`portid`*]/ *`protocol`*
Return whether the port has been added. Returns 0 if true, 1 otherwise.

[`--zone` = *`zone`*] [`--policy` = *`policy`*] `--list-protocols`
List protocols added as a space separated list.

[`--zone` = *`zone`*] [`--policy` = *`policy`*] `--add-protocol` = *`protocol`*
Add the protocol. This option can be specified multiple times. *`timeval`* is either a number (of seconds) or number followed by one of characters `s` (seconds), `m` (minutes), `h` (hours), for example `20m` or `1h`.

The protocol can be any protocol supported by the system. Please have a look at `/etc/protocols` for supported protocols.

[`--zone` = *`zone`*] [`--policy` = *`policy`*] `--remove-protocol` = *`protocol`*
Remove the protocol. This option can be specified multiple times.

[`--zone` = *`zone`*] [`--policy` = *`policy`*] `--query-protocol` = *`protocol`*
Return whether the protocol has been added. Returns 0 if true, 1 otherwise.

[`--zone` = *`zone`*] [`--policy` = *`policy`*] `--list-icmp-blocks`

List Internet Control Message Protocol (ICMP) type blocks added as a space separated list.

[`--zone`=*`zone`*] [`--policy`=*`policy`*] `--add-icmp-block`=*`icmptype`*
Add an ICMP block for *`icmptype`*. This option can be specified multiple times.

The *`icmptype`* is the one of the icmp types firewalld supports. To get a listing of supported icmp types: **firewall-cmd --get-icmptypes**

[`--zone`=*`zone`*] [`--policy`=*`policy`*] `--remove-icmp-block`=*`icmptype`*
Remove the ICMP block for *`icmptype`*. This option can be specified multiple times.

[`--zone`=*`zone`*] [`--policy`=*`policy`*] `--query-icmp-block`=*`icmptype`*
Return whether an ICMP block for *`icmptype`* has been added. Returns 0 if true, 1 otherwise.

[`--zone`=*`zone`*] [`--policy`=*`policy`*] `--list-forward-ports`
List *IPv4* forward ports added as a space separated list.

For *IPv6* forward ports, please use the rich language.

[`--zone`=*`zone`*] [`--policy`=*`policy`*] `--add-forward-port`=port=*`portid`*[-*`portid`*]:proto=*`protocol`*[:toport=*`portid`*[-*`portid`*]][:toaddr=*`address`*[/*`mask`*]]
Add the *IPv4* forward port. This option can be specified multiple times.

The port can either be a single port number *`portid`* or a port range *`portid`*-*`portid`*. The protocol can either be `tcp`, `udp`, `sctp` or `dccp`. The destination address is a simple IP address.

For *IPv6* forward ports, please use the rich language.

Note: IP forwarding will be implicitly enabled if `toaddr` is specified.

[`--zone`=*`zone`*] [`--policy`=*`policy`*] `--remove-forward-port`=port=*`portid`*[-*`portid`*]:proto=*`protocol`*[:toport=*`portid`*[-*`portid`*]][:toaddr=*`address`*[/*`mask`*]]
Remove the *IPv4* forward port. This option can be specified multiple times.

For *IPv6* forward ports, please use the rich language.

[`--zone`=*`zone`*] [`--policy`=*`policy`*] `--query-forward-port`=port=*`portid`*[-*`portid`*]:proto=*`protocol`*[:toport=*`portid`*[-*`portid`*]][:toaddr=*`address`*[/*`mask`*]]
Return whether the *IPv4* forward port has been added. Returns 0 if true, 1 otherwise.

For *IPv6* forward ports, please use the rich language.

[`--zone`=*`zone`*] [`--policy`=*`policy`*] `--list-source-ports`
List source ports added as a space separated list. A port is of the form *`portid`*[-*`portid`*]/*`protocol`*.

[`--zone`=*zone*] [`--policy`=*policy*] `--add-source-port`=*portid*[-*portid*]/*protocol*
Add the source port. This option can be specified multiple times.

The port can either be a single port number or a port range *portid*-*portid*. The protocol can either be `tcp`, `udp`, `sctp` or `dccp`.

[`--zone`=*zone*] [`--policy`=*policy*] `--remove-source-port`=*portid*[-*portid*]/*protocol*
Remove the source port. This option can be specified multiple times.

[`--zone`=*zone*] [`--policy`=*policy*] `--query-source-port`=*portid*[-*portid*]/*protocol*
Return whether the source port has been added. Returns 0 if true, 1 otherwise.

[`--zone`=*zone*] [`--policy`=*policy*] `--add-masquerade`
Enable *IPv4* masquerade. Masquerading is useful if the machine is a router and machines connected over an interface in another zone should be able to use the first connection.

For *IPv6* masquerading, please use the rich language.

Note: IP forwarding will be implicitly enabled.

[`--zone`=*zone*] [`--policy`=*policy*] `--remove-masquerade`
Disable *IPv4* masquerade.

For *IPv6* masquerading, please use the rich language.

[`--zone`=*zone*] [`--policy`=*policy*] `--query-masquerade`
Return whether *IPv4* masquerading has been enabled. Returns 0 if true, 1 otherwise.

For *IPv6* masquerading, please use the rich language.

[`--zone`=*zone*] [`--policy`=*policy*] `--list-rich-rules`
List rich language rules added as a newline separated list.

[`--zone`=*zone*] [`--policy`=*policy*] `--add-rich-rule`='*rule*'
Add rich language rule '*rule*'. This option can be specified multiple times.

For the rich language rule syntax, please have a look at firewalld.richlanguage(5).

[`--zone`=*zone*] [`--policy`=*policy*] `--remove-rich-rule`='*rule*'
Remove rich language rule '*rule*'. This option can be specified multiple times.

For the rich language rule syntax, please have a look at firewalld.richlanguage(5).

[`--zone`=*zone*] [`--policy`=*policy*] `--query-rich-rule`='*rule*'
Return whether a rich language rule '*rule*' has been added. Returns 0 if true, 1 otherwise.

For the rich language rule syntax, please have a look at firewalld.richlanguage(5).

Options to Adapt and Query Zones

Options in this section affect only one particular zone. If used with `--zone`=*zone* option, they affect the specified zone. If the option is omitted, they affect the default zone (see `--get-default-zone`).

[`--zone`=*zone*] `--add-icmp-block-inversion`
Enable ICMP block inversion.

[`--zone`=*zone*] `--remove-icmp-block-inversion`
Disable ICMP block inversion.

[`--zone`=*zone*] `--query-icmp-block-inversion`
Return whether ICMP block inversion is enabled. Returns 0 if true, 1 otherwise.

[`--zone`=*zone*] `--add-forward`
Enable intra zone forwarding.

[`--zone`=*zone*] `--remove-forward`
Disable intra zone forwarding.

[`--zone`=*zone*] `--query-forward`
Return whether intra zone forwarding is enabled. Returns 0 if true, 1 otherwise.

Options to Adapt and Query Policies

Options in this section affect only one particular policy. It's required to specify `--policy`=*policy* with these options.

`--policy`=*policy* `--get-priority`
Get the priority.

`--policy`=*policy* `--set-priority` *priority*
Set the priority. The priority determines the relative ordering of policies. This is an integer value between -32768 and 32767 where -1 is the default value for new policies and 0 is reserved for internal use.

If a priority is < 0, then the policy's rules will execute before all rules in all zones.

If a priority is > 0, then the policy's rules will execute after all rules in all zones.

`--policy`=*policy* `--list-ingress-zones`
List ingress zones added as a space separated list.

`--policy`=*policy* `--add-ingress-zone`=*zone*
Add an ingress zone. This option can be specified multiple times.

The ingress zone is one of the firewalld provided zones or one of the pseudo-zones: HOST, ANY.

HOST is used for traffic originating from the host machine, i.e. the host running firewalld.

ANY is used for traffic originating from any zone. This can be thought of as a wild card for zones. However it does not include traffic originating from the host machine - use HOST for that.

`--policy`=*policy* `--remove-ingress-zone`=*zone*
Remove an ingress zone. This option can be specified multiple times.

`--policy`=*policy* `--query-ingress-zone`=*zone*
Return whether *zone* has been added. Returns 0 if true, 1 otherwise.

`--policy`=*policy* `--list-egress-zones`
List egress zones added as a space separated list.

`--policy`=*policy* `--add-egress-zone`=*zone*
Add an egress zone. This option can be specified multiple times.

The egress zone is one of the firewalld provided zones or one of the pseudo-zones: HOST, ANY.

For clarification on HOST and ANY see option `--add-ingress-zone`.

`--policy`=*policy* `--remove-egress-zone`=*zone*
Remove an egress zone. This option can be specified multiple times.

`--policy`=*policy* `--query-egress-zone`=*zone*
Return whether *zone* has been added. Returns 0 if true, 1 otherwise.

Options to Handle Bindings of Interfaces

Binding an interface to a zone means that this zone settings are used to restrict traffic via the interface.

Options in this section affect only one particular zone. If used with `--zone`=*zone* option, they affect the zone *zone*. If the option is omitted, they affect default zone (see `--get-default-zone`).

For a list of predefined zones use **firewall-cmd --get-zones**.

An interface name is a string up to 16 characters long, that may not contain `' '`, `'/'`, `'!'` and `'*'`.

[`--zone`=*zone*] `--list-interfaces`
List interfaces that are bound to zone *zone* as a space separated list. If zone is omitted, default zone will be used.

[`--zone`=*zone*] `--add-interface`=*interface*
Bind interface *interface* to zone *zone*. If zone is omitted, default zone will be used.

[`--zone`=*zone*] `--change-interface`=*interface*
Change zone the interface *interface* is bound to to zone *zone*. If zone is omitted, default zone will be used. If old and new zone are the same, the call will be ignored without an error. If the interface has not been bound to a zone before, it will behave like `--add-interface`.

[`--zone`=*zone*] `--query-interface`=*interface*
Query whether interface *interface* is bound to zone *zone*. Returns 0 if true, 1 otherwise.

[`--zone`=*zone*] `--remove-interface`=*interface*
Remove binding of interface *interface* from zone *zone*. If zone is omitted, default zone will be used.

Options to Handle Bindings of Sources

Binding a source to a zone means that this zone settings will be used to restrict traffic from this source.

A source address or address range is either an IP address or a network IP address with a mask for IPv4 or IPv6 or a MAC address or an ipset with the ipset: prefix. For IPv4, the mask can be a network mask or a plain number. For IPv6 the mask is a plain number. The use of host names is not supported.

Options in this section affect only one particular zone. If used with `--zone`=*zone* option, they affect the zone *zone*. If the option is omitted, they affect default zone (see `--get-default-zone`).

For a list of predefined zones use **firewall-cmd --get-zones**.

[`--zone`=*zone*] `--list-sources`
List sources that are bound to zone *zone* as a space separated list. If zone is omitted, default zone will be used.

[`--zone`=*zone*] `--add-source`=*source*[/*mask*]|*MAC*|ipset:*ipset*
Bind the source to zone *zone*. If zone is omitted, default zone will be used.

[`--zone`=*zone*] `--change-source`=*source*[/*mask*]|*MAC*|ipset:*ipset*
Change zone the source is bound to to zone *zone*. If zone is omitted, default zone will be used. If old and new zone are the same, the call will be ignored without an error. If the source has not been bound to a zone before, it will behave like `--add-source`.

[`--zone`=*zone*] `--query-source`=*source*[/*mask*]|*MAC*|ipset:*ipset*
Query whether the source is bound to the zone *zone*. Returns 0 if true, 1 otherwise.

[`--zone`=*zone*] `--remove-source`=*source*[/*mask*]|*MAC*|ipset:*ipset*
Remove binding of the source from zone *zone*. If zone is omitted, default zone will be used.

IPSet Options

`--new-ipset`=*ipset* `--type`=*ipset type* [`--option`=*ipset option*[=*value*]]
Add a new permanent ipset with specifying the type and optional options.

ipset names must be alphanumeric and may additionally include characters: '_' and '-'.

`--new-ipset-from-file`=*filename* [`--name`=*ipset*]
Add a new permanent ipset from a prepared ipset file with an optional name override.

`--delete-ipset`=*ipset*
Delete an existing permanent ipset.

`--info-ipset`=*ipset*
Print information about the ipset *ipset*. The output format is:

```
ipset
  type: type
  options: option1[=value1] ..
  entries: entry1 ..
```

`--get-ipsets`
Print predefined ipsets as a space separated list.

`--ipset`=*ipset* `--add-entry`=*entry*
Add a new entry to the ipset.

`--ipset`=*ipset* `--remove-entry`=*entry*
Remove an entry from the ipset.

`--ipset`=*ipset* `--query-entry`=*entry*
Return whether the entry has been added to an ipset. Returns 0 if true, 1 otherwise.

`--ipset`=*ipset* `--get-entries`
List all entries of the ipset.

`--ipset`=*ipset* `--add-entries-from-file`=*filename*
Add a new entries to the ipset from the file. For all entries that are listed in the file but already in the ipset, a warning will be printed.

The file should contain an entry per line. Lines starting with an hash or semicolon are ignored. Also empty lines.

`--ipset`=*ipset* `--remove-entries-from-file`=*filename*
Remove existing entries from the ipset from the file. For all entries that are listed in the file but not in the ipset, a warning will be printed.

The file should contain an entry per line. Lines starting with an hash or semicolon are ignored. Also empty lines.

`--ipset`=*`ipset`* `--set-description`=*`description`*

Set new description to ipset

`--ipset`=*`ipset`* `--get-description`

Print description for ipset

`--ipset`=*`ipset`* `--set-short`=*`description`*

Set new short description to ipset

`--ipset`=*`ipset`* `--get-short`

Print short description for ipset

`--path-ipset=`*`ipset`*

Print path of the ipset configuration file.

Service Options

`--info-service=`*`service`*

Print information about the service *`service`*. The output format is:

```
service
  ports: port1 ..
  protocols: protocol1 ..
  source-ports: source-port1 ..
  helpers: helper1 ..
  destination: ipv1:address1 ..
```

`--new-service`=*`service`*

Add a new permanent service.

Service names must be alphanumeric and may additionally include characters: '_' and '-'.

`--new-service-from-file`=*`filename`* [`--name`=*`service`*]

Add a new permanent service from a prepared service file with an optional name override.

`--delete-service`=*`service`*

Delete an existing permanent service.

`--path-service=`*`service`*

Print path of the service configuration file.

`--service`=*`service`* `--set-description`=*`description`*

Set new description to service

`--service=service` `--get-description`

Print description for service

`--service=service` `--set-short=description`

Set short description to service

`--service=service` `--get-short`

Print short description for service

`--service=service` `--add-port=portid[-portid]/protocol`

Add a new port to the permanent service.

`--service=service` `--remove-port=portid[-portid]/protocol`

Remove a port from the permanent service.

`--service=service` `--query-port=portid[-portid]/protocol`

Return whether the port has been added to the permanent service.

`--service=service` `--get-ports`

List ports added to the permanent service.

`--service=service` `--add-protocol=protocol`

Add a new protocol to the permanent service.

`--service=service` `--remove-protocol=protocol`

Remove a protocol from the permanent service.

`--service=service` `--query-protocol=protocol`

Return whether the protocol has been added to the permanent service.

`--service=service` `--get-protocols`

List protocols added to the permanent service.

`--service=service` `--add-source-port=portid[-portid]/protocol`

Add a new source port to the permanent service.

`--service=service` `--remove-source-port=portid[-portid]/protocol`

Remove a source port from the permanent service.

`--service=service` `--query-source-port=portid[-portid]/protocol`

Return whether the source port has been added to the permanent service.

`--service=service` `--get-source-ports`

List source ports added to the permanent service.

`--service=service` `--add-helper=helper`

Add a new helper to the permanent service.

`--service=service` `--remove-helper=helper`
Remove a helper from the permanent service.

`--service=service` `--query-helper=helper`
Return whether the helper has been added to the permanent service.

`--service=service` `--get-service-helpers`
List helpers added to the permanent service.

`--service=service` `--set-destination=ipv:address[/mask]`
Set destination for ipv to address[/mask] in the permanent service.

`--service=service` `--remove-destination=ipv`
Remove the destination for ipv from the permanent service.

`--service=service` `--query-destination=ipv:address[/mask]`
Return whether the destination ipv to address[/mask] has been set in the permanent service.

`--service=service` `--get-destinations`
List destinations added to the permanent service.

`--service=service` `--add-include=service`
Add a new include to the permanent service.

`--service=service` `--remove-include=service`
Remove a include from the permanent service.

`--service=service` `--query-include=service`
Return whether the include has been added to the permanent service.

`--service=service` `--get-includes`
List includes added to the permanent service.

Helper Options

Options in this section affect only one particular helper.

`--info-helper=helper`
Print information about the helper `helper`. The output format is:

```
helper
  family: family
  module: module
  ports: port1 ..
```

The following options are only usable in the permanent configuration.

`--new-helper`=*`helper`* `--module`=*`nf_conntrack_module`* [`--family`=`ipv4` | `ipv6`]
Add a new permanent helper with module and optionally family defined.

Helper names must be alphanumeric and may additionally include characters: '-'.

`--new-helper-from-file`=*`filename`* [`--name`=*`helper`*]
Add a new permanent helper from a prepared helper file with an optional name override.

`--delete-helper`=*`helper`*
Delete an existing permanent helper.

`--load-helper-defaults`=*`helper`*
Load helper default settings or report NO_DEFAULTS error.

`--path-helper=`*`helper`*
Print path of the helper configuration file.

`--get-helpers`
Print predefined helpers as a space separated list.

`--helper`=*`helper`* `--set-description`=*`description`*
Set new description to helper

`--helper`=*`helper`* `--get-description`
Print description for helper

`--helper`=*`helper`* `--set-short`=*`description`*
Set short description to helper

`--helper`=*`helper`* `--get-short`
Print short description for helper

`--helper`=*`helper`* `--add-port`=*`portid`*[-*`portid`*]/*`protocol`*
Add a new port to the permanent helper.

`--helper`=*`helper`* `--remove-port`=*`portid`*[-*`portid`*]/*`protocol`*
Remove a port from the permanent helper.

`--helper`=*`helper`* `--query-port`=*`portid`*[-*`portid`*]/*`protocol`*
Return whether the port has been added to the permanent helper.

`--helper`=*`helper`* `--get-ports`
List ports added to the permanent helper.

`--helper`=*`helper`* `--set-module`=*`description`*
Set module description for helper

`--helper`=*`helper`* `--get-module`
Print module description for helper

`--helper`=*`helper`* `--set-family`=*`description`*
Set family description for helper

`--helper`=*`helper`* `--get-family`
Print family description of helper

Internet Control Message Protocol (ICMP) type Options

`--info-icmptype=`*`icmptype`*
Print information about the icmptype *`icmptype`*. The output format is:

```
icmptype
  destination: ipv1 ..
```

`--new-icmptype`=*`icmptype`*
Add a new permanent icmptype.

ICMP type names must be alphanumeric and may additionally include characters: '_' and '-'.

`--new-icmptype-from-file`=*`filename`* [`--name`=*`icmptype`*]
Add a new permanent icmptype from a prepared icmptype file with an optional name override.

`--delete-icmptype`=*`icmptype`*
Delete an existing permanent icmptype.

`--icmptype`=*`icmptype`* `--set-description`=*`description`*
Set new description to icmptype

`--icmptype`=*`icmptype`* `--get-description`
Print description for icmptype

`--icmptype`=*`icmptype`* `--set-short`=*`description`*
Set short description to icmptype

`--icmptype`=*`icmptype`* `--get-short`
Print short description for icmptype

`--icmptype`=*`icmptype`* `--add-destination`=*`ipv`*
Enable destination for ipv in permanent icmptype. ipv is one of `ipv4` or `ipv6`.

`--icmptype`=*`icmptype`* `--remove-destination`=*`ipv`*
Disable destination for ipv in permanent icmptype. ipv is one of `ipv4` or `ipv6`.

`--icmptype`=*`icmptype`* `--query-destination`=*`ipv`*
Return whether destination for ipv is enabled in permanent icmptype. ipv is one of `ipv4` or `ipv6`.

`--icmptype`=*`icmptype`* `--get-destinations`
List destinations in permanent icmptype.

`--path-icmptype=`*`icmptype`*
Print path of the icmptype configuration file.

Direct Options

DEPRECATED

The direct interface has been deprecated. It will be removed in a future release. It is superseded by policies, see firewalld.policies(5).

The direct options give a more direct access to the firewall. These options require user to know basic iptables concepts, i.e. `table` (filter/mangle/nat/...), `chain` (INPUT/OUTPUT/FORWARD/...), `commands` (-A/-D/-I/...), `parameters` (-p/-s/-d/-j/...) and `targets` (ACCEPT/DROP/REJECT/...).

Direct options should be used only as a last resort when it's not possible to use for example `--add-service`=*`service`* or `--add-rich-rule`='*`rule`'.

Warning: Direct rules behavior is different depending on the value of `FirewallBackend`. See `CAVEATS` in firewalld.direct(5).

The first argument of each option has to be `ipv4` or `ipv6` or `eb`. With `ipv4` it will be for IPv4 (iptables(8)), with `ipv6` for IPv6 (ip6tables(8)) and with `eb` for ethernet bridges (ebtables(8)).

`--direct` `--get-all-chains`
Get all chains added to all tables.

This option concerns only chains previously added with `--direct --add-chain`.

`--direct` `--get-chains` { `ipv4` | `ipv6` | `eb` } *`table`*
Get all chains added to table *`table`* as a space separated list.

This option concerns only chains previously added with `--direct --add-chain`.

`--direct` `--add-chain` { `ipv4` | `ipv6` | `eb` } *`table`* *`chain`*
Add a new chain with name *`chain`* to table *`table`*.

There already exist basic chains to use with direct options, for example `INPUT_direct` chain (see `iptables-save | grep direct` output for all of them). These chains are jumped into

before chains for zones, i.e. every rule put into `INPUT_direct` will be checked before rules in zones.

`--direct` `--remove-chain` { `ipv4` | `ipv6` | `eb` } *`table`* *`chain`*
Remove the chain with name *`chain`* from table *`table`*.

`--direct` `--query-chain` { `ipv4` | `ipv6` | `eb` } *`table`* *`chain`*
Return whether a chain with name *`chain`* exists in table *`table`*. Returns 0 if true, 1 otherwise.

This option concerns only chains previously added with `--direct --add-chain`.

`--direct` `--get-all-rules`
Get all rules added to all chains in all tables as a newline separated list of the priority and arguments.

`--direct` `--get-rules` { `ipv4` | `ipv6` | `eb` } *`table`* *`chain`*
Get all rules added to chain *`chain`* in table *`table`* as a newline separated list of the priority and arguments.

`--direct` `--add-rule` { `ipv4` | `ipv6` | `eb` } *`table`* *`chain`* *`priority`* *`args`*
Add a rule with the arguments *`args`* to chain *`chain`* in table *`table`* with priority *`priority`*.

The *`priority`* is used to order rules. Priority 0 means add rule on top of the chain, with a higher priority the rule will be added further down. Rules with the same priority are on the same level and the order of these rules is not fixed and may change. If you want to make sure that a rule will be added after another one, use a low priority for the first and a higher for the following.

`--direct` `--remove-rule` { `ipv4` | `ipv6` | `eb` } *`table`* *`chain`* *`priority`* *`args`*
Remove a rule with *`priority`* and the arguments *`args`* from chain *`chain`* in table *`table`*.

`--direct` `--remove-rules` { `ipv4` | `ipv6` | `eb` } *`table`* *`chain`*
Remove all rules in the chain with name *`chain`* exists in table *`table`*.

This option concerns only rules previously added with `--direct --add-rule` in this chain.

`--direct` `--query-rule` { `ipv4` | `ipv6` | `eb` } *`table`* *`chain`* *`priority`* *`args`*
Return whether a rule with *`priority`* and the arguments *`args`* exists in chain *`chain`* in table *`table`*. Returns 0 if true, 1 otherwise.

`--direct` `--get-all-passthroughs`
Get all permanent passthrough as a newline separated list of the ipv value and arguments.

`--direct` `--get-passthroughs` { `ipv4` | `ipv6` | `eb` }
Get all permanent passthrough rules for the ipv value as a newline separated list of the priority and arguments.

`--direct` `--add-passthrough` { `ipv4` | `ipv6` | `eb` } *`args`*

Add a permanent passthrough rule with the arguments *args* for the ipv value.

`--direct` `--remove-passthrough` { `ipv4` | `ipv6` | `eb` } *args*
Remove a permanent passthrough rule with the arguments *args* for the ipv value.

`--direct` `--query-passthrough` { `ipv4` | `ipv6` | `eb` } *args*
Return whether a permanent passthrough rule with the arguments *args* exists for the ipv value. Returns 0 if true, 1 otherwise.

Lockdown Options

Local applications or services are able to change the firewall configuration if they are running as root (example: libvirt) or are authenticated using PolicyKit. With this feature administrators can lock the firewall configuration so that only applications on lockdown whitelist are able to request firewall changes.

The lockdown access check limits D-Bus methods that are changing firewall rules. Query, list and get methods are not limited.

The lockdown feature is a very light version of user and application policies for firewalld and is turned off by default.

`--lockdown-on`
Enable lockdown. Be careful - if firewall-cmd is not on lockdown whitelist when you enable lockdown you won't be able to disable it again with firewall-cmd, you would need to edit firewalld.conf.

`--lockdown-off`
Disable lockdown.

`--query-lockdown`
Query whether lockdown is enabled. Returns 0 if lockdown is enabled, 1 otherwise.

Lockdown Whitelist Options

The lockdown whitelist can contain *commands*, *contexts*, *users* and *user ids*.

If a command entry on the whitelist ends with an asterisk '*', then all command lines starting with the command will match. If the '*' is not there the absolute command inclusive arguments must match.

Commands for user root and others is not always the same. Example: As root **/bin/firewall-cmd** is used, as a normal user **/usr/bin/firewall-cmd** is be used on Fedora.

The context is the security (SELinux) context of a running application or service. To get the context of a running application use **ps -e --context**.

Warning: If the context is unconfined, then this will open access for more than the desired application.

The lockdown whitelist entries are checked in the following order:

1. *context*
2. *uid*
3. *user*
4. *command*

`--list-lockdown-whitelist-commands`
List all command lines that are on the whitelist.

`--add-lockdown-whitelist-command`=*command*
Add the *command* to the whitelist.

`--remove-lockdown-whitelist-command`=*command*
Remove the *command* from the whitelist.

`--query-lockdown-whitelist-command`=*command*
Query whether the *command* is on the whitelist. Returns 0 if true, 1 otherwise.

`--list-lockdown-whitelist-contexts`
List all contexts that are on the whitelist.

`--add-lockdown-whitelist-context`=*context*
Add the context *context* to the whitelist.

`--remove-lockdown-whitelist-context`=*context*
Remove the *context* from the whitelist.

`--query-lockdown-whitelist-context`=*context*
Query whether the *context* is on the whitelist. Returns 0 if true, 1 otherwise.

`--list-lockdown-whitelist-uids`
List all user ids that are on the whitelist.

`--add-lockdown-whitelist-uid`=*uid*
Add the user id *uid* to the whitelist.

`--remove-lockdown-whitelist-uid`=*uid*
Remove the user id *uid* from the whitelist.

`--query-lockdown-whitelist-uid`=*uid*
Query whether the user id *uid* is on the whitelist. Returns 0 if true, 1 otherwise.

`--list-lockdown-whitelist-users`
List all user names that are on the whitelist.

`--add-lockdown-whitelist-user`=*`user`*
Add the user name *`user`* to the whitelist.

`--remove-lockdown-whitelist-user`=*`user`*
Remove the user name *`user`* from the whitelist.

`--query-lockdown-whitelist-user`=*`user`*
Query whether the user name *`user`* is on the whitelist. Returns 0 if true, 1 otherwise.

Policy Options

`--policy-server`
Change Polkit actions to 'server' (more restricted)

`--policy-desktop`
Change Polkit actions to 'desktop' (less restricted)

See Also

firewall-applet(1), firewalld(1), firewall-cmd(1), firewall-config(1), firewalld.conf(5), firewalld.direct(5), firewalld.dbus(5), firewalld.icmptype(5), firewalld.lockdown-whitelist(5), firewall-offline-cmd(1), firewalld.richlanguage(5), firewalld.service(5), firewalld.zone(5), firewalld.zones(5), firewalld.policy(5), firewalld.policies(5), firewalld.ipset(5), firewalld.helper(5)

Notes

firewalld home page:
http://firewalld.org

More documentation with examples:
http://fedoraproject.org/wiki/FirewallD