Full credit is given to the above companies including the OS that this PDF file was generated!

Red Hat Enterprise Linux Release 9.2 Manual Pages on 'firewalld.dbus.5' command

$ man firewalld.dbus.5

FIREWALLD.DBUS(5) firewalld.dbus FIREWALLD.DBUS(5)

NAME

firewalld.dbus - firewalld D-Bus interface description

OBJECT PATHS

This is the basic firewalld object path structure. The used interfaces

are explained below in the section called ?INTERFACES?.

/org/fedoraproject/FirewallD1

Interfaces

org.fedoraproject.FirewallD1

org.fedoraproject.FirewallD1.direct (deprecated)

org.fedoraproject.FirewallD1.ipset

org.fedoraproject.FirewallD1.policies

org.fedoraproject.FirewallD1.zone

org.freedesktop.DBus.Introspectable

org.freedesktop.DBus.Properties

/org/fedoraproject/FirewallD1/config

Interfaces

org.fedoraproject.FirewallD1.config

org.fedoraproject.FirewallD1.config.direct (deprecated)

org.fedoraproject.FirewallD1.config.policies

org.freedesktop.DBus.Introspectable

org.freedesktop.DBus.Properties

/org/fedoraproject/FirewallD1/config/zone/i

Interfaces

org.fedoraproject.FirewallD1.config.zone

org.freedesktop.DBus.Introspectable

org.freedesktop.DBus.Properties

/org/fedoraproject/FirewallD1/config/service/i

Interfaces:

org.fedoraproject.FirewallD1.config.service

org.freedesktop.DBus.Introspectable

org.freedesktop.DBus.Properties

/org/fedoraproject/FirewallD1/config/ipset/i

Interfaces

org.fedoraproject.FirewallD1.config.ipset

org.freedesktop.DBus.Introspectable

org.freedesktop.DBus.Properties

/org/fedoraproject/FirewallD1/config/icmptype/i

Interfaces

org.fedoraproject.FirewallD1.config.icmptype

org.freedesktop.DBus.Introspectable

org.freedesktop.DBus.Properties

INTERFACES

org.fedoraproject.FirewallD1

This interface contains general runtime operations, like: reloading, panic mode, default zone handling, getting services and icmp types and their settings.

Methods

authorizeAll() ? Nothing

Initiate authorization for the complete firewalld D-Bus interface. This method it mostly useful for configuration applications.

completeReload() ? Nothing

Reload firewall completely, even netfilter kernel modules. This will most likely terminate active connections, because state information is lost. This option should only be used in case of

severe firewall problems. For example if there are state information problems that no connection can be established with correct firewall rules.

disablePanicMode() ? Nothing

Disable panic mode. After disabling panic mode established connections might work again, if panic mode was enabled for a short period of time.

Possible errors: NOT_ENABLED, COMMAND_FAILED

enablePanicMode() ? Nothing

Enable panic mode. All incoming and outgoing packets are dropped, active connections will expire. Enable this only if there are serious problems with your network environment.

Possible errors: ALREADY_ENABLED, COMMAND_FAILED

getAutomaticHelpers() ? s

Deprecated. This always returns "no".

getDefaultZone() ? s

Return default zone.

getHelperSettings(s: helper) ? (sssssa(ss))

Return runtime settings of given helper. For getting permanent settings see org.fedoraproject.FirewallD1.config.helper.Methods.getSettings. Settings are in format: version, name, description, family, module and array of ports.

version (s): see version attribute of helper tag in firewalld.helper(5).

name (s): see short tag in firewalld.helper(5).

description (s): see description tag in firewalld.helper(5).

family (s): see family tag in firewalld.helper(5).

module (s): see module tag in firewalld.helper(5).

ports (a(ss)): array of port and protocol pairs. See port tag in firewalld.helper(5).

Possible errors: INVALID_HELPER

getHelpers() ? as

Return array of helper names (s) in runtime configuration. For permanent configuration see org.fedoraproject.FirewallD1.config.Methods.listHelpers.

getIcmpTypeSettings(s: icmptype) ? (sssas)

Return runtime settings of given icmptype. For getting permanent settings see org.fedoraproject.FirewallD1.config.icmptype.Methods.getSettings. Settings are in format: version, name, description, array of destinations.

version (s): see version attribute of icmptype tag in firewalld.icmptype(5).

name (s): see short tag in firewalld.icmptype(5).

description (s): see description tag in firewalld.icmptype(5).

destinations (as): array, either empty or containing strings 'ipv4' or 'ipv6', see destination tag in firewalld.icmptype(5).

Possible errors: INVALID_ICMPTYPE

getLogDenied() ? s

Returns the LogDenied value. If LogDenied is enabled, then logging rules are added 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. Possible values are: all, unicast, broadcast, multicast and off. The default value is off

getServiceSettings(s: service) ? (sssa(ss)asa{ss}asa(ss))

This function is deprecated, use org.fedoraproject.FirewallD1.Methods.getServiceSettings2 instead.

getServiceSettings2(s: service) ? s{sv}

Return runtime settings of given service. For getting permanent settings see org.fedoraproject.FirewallD1.config.service.Methods.getSettings2. Settings are a dictionary indexed by keywords. For the type of each value see below. If the value is empty it may be omitted.

version (s): see version attribute of service tag in firewalld.service(5).

name (s): see short tag in firewalld.service(5).

description (s): see description tag in firewalld.service(5).

ports (a(ss)): array of port and protocol pairs. See port tag in firewalld.service(5).

module names (as): array of kernel netfilter helpers, see module tag in firewalld.service(5).

destinations (a{ss}): dictionary of {IP family : IP address} where 'IP family' key can be either 'ipv4' or 'ipv6'. See destination tag in firewalld.service(5).

protocols (as): array of protocols, see protocol tag in firewalld.service(5).

source_ports (a(ss)): array of port and protocol pairs. See source-port tag in firewalld.service(5).

includes (as): array of service includes, see include tag in firewalld.service(5).

helpers (as): array of service helpers, see helper tag in firewalld.service(5).

Possible errors: INVALID_SERVICE

getZoneSettings(s: zone) ? (sssbsasa(ss)asba(ssss)asasasasa(ss)b)

This function is deprecated, use org.fedoraproject.FirewallD1.zone.Methods.getZoneSettings2 instead.

listIcmpTypes() ? as

Return array of names (s) of icmp types in runtime configuration. For permanent configuration see org.fedoraproject.FirewallD1.config.Methods.listIcmpTypes.

listServices() ? as

Return array of service names (s) in runtime configuration. For permanent configuration see org.fedoraproject.FirewallD1.config.Methods.listServices.

queryPanicMode() ? b

Return true if panic mode is enabled, false otherwise. In panic mode all incoming and outgoing packets are dropped.

reload() ? Nothing

Reload firewall rules and keep state information. Current permanent configuration will become new runtime configuration, i.e. all runtime only changes done until reload are lost with reload if they have not been also in permanent configuration.

runtimeToPermanent() ? Nothing

Make runtime settings permanent. Replaces permanent settings with runtime settings for zones, services, icmptypes, direct (deprecated) and policies (lockdown whitelist).

Possible errors: RT_TO_PERM_FAILED

checkPermanentConfig() ? Nothing

Run checks on the permanent configuration. This is most useful if changes were made manually to configuration files.

Possible errors: any

setDefaultZone(s: zone) ? Nothing

Set default zone for connections and interfaces where no zone has been selected to zone. Setting the default zone changes the zone for the connections or interfaces, that are using the default zone. This is a runtime and permanent change.

Possible errors: ZONE_ALREADY_SET, COMMAND_FAILED

setLogDenied(s: value) ? Nothing

Set LogDenied value to value. If LogDenied is enabled, then logging rules are added 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. Possible values are: all, unicast, broadcast, multicast and off. The default value is off This is a runtime and permanent change.

Possible errors: ALREADY_SET, INVALID_VALUE

Signals

DefaultZoneChanged(s: zone)

Emitted when default zone has been changed to zone.

LogDeniedChanged(s: value)

Emitted when LogDenied value has been changed.

PanicModeDisabled()

Emitted when panic mode has been deactivated.

PanicModeEnabled()

Emitted when panic mode has been activated.

Reloaded()

Emitted when firewalld has been reloaded. Also emitted for a complete reload.

Properties

BRIDGE - b - (ro)

Indicates whether the firewall has ethernet bridge support.

IPSet - b - (ro)

Indicates whether the firewall has IPSet support.

IPSetTypes - as - (ro)

The supported IPSet types by ipset and firewalld.

IPv4 - b - (ro)

Indicates whether the firewall has IPv4 support.

IPv4ICMPTypes - as - (ro)

The list of supported IPv4 ICMP types.

IPv6 - b - (ro)

Indicates whether the firewall has IPv6 support.

IPv6_rpfilter - b - (ro)

Indicates whether the reverse path filter test on a packet for IPv6 is enabled. If a reply to the packet would be sent via the same interface that the packet arrived on, the packet will match and be accepted, otherwise dropped.

IPv6ICMPTypes - as - (ro)

The list of supported IPv6 ICMP types.

nf_conntrach_helper_setting - b - (ro)

Deprecated. Always False.

nf_conntrack_helpers - a{sas} - (ro)

Deprecated. Always returns an empty dictionary.

nf_nat_helpers - a{sas} - (ro)

Deprecated. Always returns an empty dictionary.

interface_version - s - (ro)

firewalld D-Bus interface version string.

state - s - (ro)

firewalld state. This can be either INIT, FAILED, or RUNNING.

In INIT state, firewalld is starting up and initializing. In

FAILED state, firewalld completely started but experienced a

failure.

version - s - (ro)

firewalld version string.

org.fedoraproject.FirewallD1.ipset

Operations in this interface allows one to get, add, remove and query

runtime ipset settings. For permanent configuration see

org.fedoraproject.FirewallD1.config.ipset interface.

Methods

addEntry(s: ipset, s: entry) ? as

Add a new entry to ipset. The entry must match the type of the

ipset. If the ipset is using the timeout option, it is not

possible to see the entries, as they are timing out

automatically in the kernel. For permanent operation see

org.fedoraproject.FirewallD1.config.ipset.Methods.addEntry.

Possible errors: INVALID_IPSET, IPSET_WITH_TIMEOUT

getEntries(s: ipset) ? Nothing

Get all entries added to the ipset. If the ipset is using the

timeout option, it is not possible to see the entries, as they

are timing out automatically in the kernel. Return value is a

array of entry. For permanent operation see

org.fedoraproject.FirewallD1.config.ipset.Methods.getEntries.

Possible errors: INVALID_IPSET, IPSET_WITH_TIMEOUT

getIPSetSettings(s: ipset) ? (ssssa{ss}as)

Return runtime settings of given ipset. For getting permanent

settings see

org.fedoraproject.FirewallD1.config.ipset.Methods.getSettings.

Settings are in format: version, name, description, type,

dictionary of options and array of entries.

version (s): see version attribute of ipset tag in

firewalld.ipset(5).

name (s): see short tag in firewalld.ipset(5).

description (s): see description tag in firewalld.ipset(5).

type (s): see type attribute of ipset tag in

firewalld.ipset(5).

options (a{ss}): dictionary of {option : value} . See options

tag in firewalld.ipset(5).

entries (as): array of entries, see entry tag in

firewalld.ipset(5).

Possible errors: INVALID_IPSET

getIPSets() ? as

Return array of ipset names (s) in runtime configuration. For

permanent configuration see

org.fedoraproject.FirewallD1.config.Methods.listIPSets.

queryEntry(s: ipset, s: entry) ? b

Return whether entry has been added to ipset. For permanent

operation see

org.fedoraproject.FirewallD1.config.ipset.Methods.queryEntry.

Possible errors: INVALID_IPSET

queryIPSet(s: ipset) ? b

Return whether ipset is defined in runtime configuration.

removeEntry(s: ipset, s: entry) ? as

Removes an entry from ipset. For permanent operation see

org.fedoraproject.FirewallD1.config.ipset.Methods.removeEntry.

Possible errors: INVALID_IPSET, IPSET_WITH_TIMEOUT

setEntries(as: entries) ? Nothing

Permanently set list of entries to entries. For permanent

operation see

org.fedoraproject.FirewallD1.config.ipset.Methods.setEntries.

See entry tag in firewalld.ipset(5).

Signals

EntryAdded(s: ipset, s: entry)

Emitted when entry has been added to ipset.

EntryRemoved(s: ipset, s: entry)

Emitted when entry has been removed from ipset.

org.fedoraproject.FirewallD1.direct

DEPRECATED

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

This interface enables more direct access to the firewall. It enables runtime manipulation with chains and rules. For permanent configuration see org.fedoraproject.FirewallD1.config.direct interface.

Methods

addChain(s: ipv, s: table, s: chain) ? Nothing

Add a new chain to table for ipv being either ipv4 (iptables) or ipv6 (ip6tables) or eb (ebtables). Make sure there's no other chain with this name already. There already exist basic chains to use with direct methods, for example INPUT_direct chain. These chains are jumped into before chains for zones, i.e. every rule put into INPUT_direct will be checked before rules in zones. For permanent operation see org.fedoraproject.FirewallD1.config.direct.Methods.addChain. Possible errors: INVALID_IPV, INVALID_TABLE, ALREADY_ENABLED, COMMAND_FAILED

addPassthrough(s: ipv, as: args) ? Nothing

Add a tracked passthrough rule with the arguments args for ipv being either ipv4 (iptables) or ipv6 (ip6tables) or eb (ebtables). Valid commands in args are only -A/--append, -I/--insert and -N/--new-chain. This method is (unlike passthrough method) tracked, i.e. firewalld remembers it. It's useful with

org.fedoraproject.FirewallD1.Methods.runtimeToPermanent For
permanent operation see
org.fedoraproject.FirewallD1.config.direct.Methods.addPassthrough.
Possible errors: INVALID_IPV, ALREADY_ENABLED, COMMAND_FAILED

addRule(s: ipv, s: table, s: chain, i: priority, as: args) ?

Nothing

Add a rule with the arguments args to chain in table with
priority for ipv being either ipv4 (iptables) or ipv6
(ip6tables) or eb (ebtables). 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. For permanent
operation see
org.fedoraproject.FirewallD1.config.direct.Methods.addRule.
Possible errors: INVALID_IPV, INVALID_TABLE, ALREADY_ENABLED,
COMMAND_FAILED

getAllChains() ? a(sss)

Get all chains added to all tables in format: ipv, table,
chain. This concerns only chains previously added with
addChain. Return value is a array of (ipv, table, chain). For
permanent operation see
org.fedoraproject.FirewallD1.config.direct.Methods.getAllChains.
ipv (s): either ipv4 (iptables) or ipv6 (ip6tables) or eb
(ebtables).
table (s): one of filter, mangle, nat, raw, security
chain (s): name of a chain.

getAllPassthroughs() ? a(sas)

Get all tracked passthrough rules added in all ipv types in
format: ipv, rule. This concerns only rules previously added
with addPassthrough. Return value is a array of (ipv, array of

arguments). For permanent operation see
org.fedoraproject.FirewallD1.config.direct.Methods.getAllPassthroughs.
ipv (s): either ipv4 (iptables) or ipv6 (ip6tables) or eb
(ebtables).
arguments (as): array of commands, parameters and other
iptables/ip6tables/ebtables command line options.

getAllRules() ? a(sssias)

Get all rules added to all chains in all tables in format: ipv,
table, chain, priority, rule. This concerns only rules
previously added with addRule. Return value is a array of (ipv,
table, chain, priority, array of arguments). For permanent
operation see
org.fedoraproject.FirewallD1.config.direct.Methods.getAllRules.
ipv (s): either ipv4 (iptables) or ipv6 (ip6tables) or eb
(ebtables).
table (s): one of filter, mangle, nat, raw, security
chain (s): name of a chain.
priority (i): used to order rules.
arguments (as): array of commands, parameters and other
iptables/ip6tables/ebtables command line options.

getChains(s: ipv, s: table) ? as

Return an array of chains (s) added to table for ipv being
either ipv4 (iptables) or ipv6 (ip6tables) or eb (ebtables).
This concerns only chains previously added with addChain. For
permanent operation see
org.fedoraproject.FirewallD1.config.direct.Methods.getChains.
Possible errors: INVALID_IPV, INVALID_TABLE

getPassthroughs(s: ipv) ? aas

Get tracked passthrough rules added in either ipv4 (iptables)
or ipv6 (ip6tables) or eb (ebtables). This concerns only rules
previously added with addPassthrough. Return value is a array
of (array of arguments). For permanent operation see
org.fedoraproject.FirewallD1.config.direct.Methods.getPassthroughs.

arguments (as): array of commands, parameters and other iptables/ip6tables/ebtables command line options.

getRules(s: ipv, s: table, s: chain) ? a(ias)

Get all rules added to chain in table for ipv being either ipv4 (iptables) or ipv6 (ip6tables) or eb (ebtables). This concerns only rules previously added with addRule. Return value is a array of (priority, array of arguments). For permanent operation see org.fedoraproject.FirewallD1.config.direct.Methods.getRules.

priority (i): used to order rules.

arguments (as): array of commands, parameters and other iptables/ip6tables/ebtables command line options.

Possible errors: INVALID_IPV, INVALID_TABLE

passthrough(s: ipv, as: args) ? s

Pass a command through to the firewall. ipv can be either ipv4 (iptables) or ipv6 (ip6tables) or eb (ebtables). args can be all iptables, ip6tables and ebtables command line arguments. args can be all iptables, ip6tables and ebtables command line arguments. This command is untracked, which means that firewalld is not able to provide information about this command later on.

Possible errors: COMMAND_FAILED

queryChain(s: ipv, s: table, s: chain) ? b

Return whether a chain exists in table for ipv being either ipv4 (iptables) or ipv6 (ip6tables) or eb (ebtables). This concerns only chains previously added with addChain. For permanent operation see org.fedoraproject.FirewallD1.config.direct.Methods.queryChain.

Possible errors: INVALID_IPV, INVALID_TABLE

queryPassthrough(s: ipv, as: args) ? b

Return whether a tracked passthrough rule with the arguments args exists for ipv being either ipv4 (iptables) or ipv6 (ip6tables) or eb (ebtables). This concerns only rules

previously added with addPassthrough. For permanent operation see org.fedoraproject.FirewallD1.config.direct.Methods.queryPassthrough. Possible errors: INVALID_IPV

queryRule(s: ipv, s: table, s: chain, i: priority, as: args) ? b

Return whether a rule with priority and the arguments args exists in chain in table for ipv being either ipv4 (iptables) or ipv6 (ip6tables) or eb (ebtables). This concerns only rules previously added with addRule. For permanent operation see org.fedoraproject.FirewallD1.config.direct.Methods.queryRule. Possible errors: INVALID_IPV, INVALID_TABLE

removeAllPassthroughs() ? Nothing

Remove all passthrough rules previously added with addPassthrough.

removeChain(s: ipv, s: table, s: chain) ? Nothing

Remove a chain from table for ipv being either ipv4 (iptables) or ipv6 (ip6tables) or eb (ebtables). Only chains previously added with addChain can be removed this way. For permanent operation see org.fedoraproject.FirewallD1.config.direct.Methods.removeChain. Possible errors: INVALID_IPV, INVALID_TABLE, NOT_ENABLED, COMMAND_FAILED

removePassthrough(s: ipv, as: args) ? Nothing

Remove a tracked passthrough rule with arguments args for ipv being either ipv4 (iptables) or ipv6 (ip6tables) or eb (ebtables). Only rules previously added with addPassthrough can be removed this way. For permanent operation see org.fedoraproject.FirewallD1.config.direct.Methods.removePassthrough. Possible errors: INVALID_IPV, NOT_ENABLED, COMMAND_FAILED

removeRule(s: ipv, s: table, s: chain, i: priority, as: args) ? Nothing

Remove a rule with priority and arguments args from chain in table for ipv being either ipv4 (iptables) or ipv6 (ip6tables)

or eb (ebtables). Only rules previously added with addRule can be removed this way. For permanent operation see org.fedoraproject.FirewallD1.config.direct.Methods.removeRule. Possible errors: INVALID_IPV, INVALID_TABLE, NOT_ENABLED, COMMAND_FAILED

removeRules(s: ipv, s: table, s: chain) ? Nothing

Remove all rules from chain in table for ipv being either ipv4 (iptables) or ipv6 (ip6tables) or eb (ebtables). This concerns only rules previously added with addRule. For permanent operation see org.fedoraproject.FirewallD1.config.direct.Methods.removeRules. Possible errors: INVALID_IPV, INVALID_TABLE

Signals

ChainAdded(s: ipv, s: table, s: chain)

Emitted when chain has been added into table for ipv being either ipv4 (iptables) or ipv6 (ip6tables) or eb (ebtables).

ChainRemoved(s: ipv, s: table, s: chain)

Emitted when chain has been removed from table for ipv being either ipv4 (iptables) or ipv6 (ip6tables) or eb (ebtables).

PassthroughAdded(s: ipv, as: args)

Emitted when a tracked passthrough rule with args has been added for ipv being either ipv4 (iptables) or ipv6 (ip6tables) or eb (ebtables).

PassthroughRemoved(s: ipv, as: args)

Emitted when a tracked passthrough rule with args has been removed for ipv being either ipv4 (iptables) or ipv6 (ip6tables) or eb (ebtables).

RuleAdded(s: ipv, s: table, s: chain, i: priority, as: args)

Emitted when a rule with args has been added to chain in table with priority for ipv being either ipv4 (iptables) or ipv6 (ip6tables) or eb (ebtables).

RuleRemoved(s: ipv, s: table, s: chain, i: priority, as: args)

Emitted when a rule with args has been removed from chain in

table with priority for ipv being either ipv4 (iptables) or
ipv6 (ip6tables) or eb (ebtables).

org.fedoraproject.FirewallD1.policies

Enables firewalld to be able to lock down configuration changes from local applications. Local applications or services are able to change the firewall configuration if they are running as root (example: libvirt). With these operations administrator can lock the firewall configuration so that either none or only applications that are in the whitelist are able to request firewall changes. For permanent configuration see org.fedoraproject.FirewallD1.config.policies interface.

Methods

addLockdownWhitelistCommand(s: command) ? Nothing

Add command to whitelist. See command option in firewalld.lockdown-whitelist(5). For permanent operation see org.fedoraproject.FirewallD1.config.policies.Methods.addLockdownWhitelistCommand. Possible errors: ALREADY_ENABLED, INVALID_COMMAND

addLockdownWhitelistContext(s: context) ? Nothing

Add context to whitelist. See selinux option in firewalld.lockdown-whitelist(5). For permanent operation see org.fedoraproject.FirewallD1.config.policies.Methods.addLockdownWhitelistContext. Possible errors: ALREADY_ENABLED, INVALID_COMMAND

addLockdownWhitelistUid(i: uid) ? Nothing

Add user id uid to whitelist. See user option in firewalld.lockdown-whitelist(5). For permanent operation see org.fedoraproject.FirewallD1.config.policies.Methods.addLockdownWhitelistUid. Possible errors: ALREADY_ENABLED, INVALID_COMMAND

addLockdownWhitelistUser(s: user) ? Nothing

Add user name to whitelist. See user option in firewalld.lockdown-whitelist(5). For permanent operation see org.fedoraproject.FirewallD1.config.policies.Methods.addLockdownWhitelistUser. Possible errors: ALREADY_ENABLED, INVALID_COMMAND

disableLockdown() ? Nothing

Disable lockdown. This is a runtime and permanent change.

Possible errors: NOT_ENABLED

enableLockdown() ? Nothing

Enable lockdown. Be careful - if the calling application/user

is not on lockdown whitelist when you enable lockdown you won't

be able to disable it again with the application, you would

need to edit firewalld.conf. This is a runtime and permanent

change.

Possible errors: ALREADY_ENABLED

getLockdownWhitelistCommands() ? as

List all command lines (s) that are on whitelist. For permanent

operation see

org.fedoraproject.FirewallD1.config.policies.Methods.getLockdownWhitelistCommands.

getLockdownWhitelistContexts() ? as

List all contexts (s) that are on whitelist. For permanent

operation see

org.fedoraproject.FirewallD1.config.policies.Methods.getLockdownWhitelistContexts.

getLockdownWhitelistUids() ? ai

List all user ids (i) that are on whitelist. For permanent

operation see

org.fedoraproject.FirewallD1.config.policies.Methods.getLockdownWhitelistUids.

getLockdownWhitelistUsers() ? as

List all users (s) that are on whitelist. For permanent

operation see

org.fedoraproject.FirewallD1.config.policies.Methods.getLockdownWhitelistUsers.

queryLockdown() ? b

Query whether lockdown is enabled.

queryLockdownWhitelistCommand(s: command) ? b

Query whether command is on whitelist. For permanent operation

see

org.fedoraproject.FirewallD1.config.policies.Methods.queryLockdownWhitelistCommand.

queryLockdownWhitelistContext(s: context) ? b

Query whether context is on whitelist. For permanent operation

see

org.fedoraproject.FirewallD1.config.policies.Methods.queryLockdownWhitelistContext.

queryLockdownWhitelistUid(i: uid) ? b

Query whether user id uid is on whitelist. For permanent

operation see

org.fedoraproject.FirewallD1.config.policies.Methods.queryLockdownWhitelistUid.

queryLockdownWhitelistUser(s: user) ? b

Query whether user is on whitelist. For permanent operation see

org.fedoraproject.FirewallD1.config.policies.Methods.queryLockdownWhitelistUser.

removeLockdownWhitelistCommand(s: command) ? Nothing

Remove command from whitelist. For permanent operation see

org.fedoraproject.FirewallD1.config.policies.Methods.removeLockdownWhitelistCommand.

Possible errors: NOT_ENABLED

removeLockdownWhitelistContext(s: context) ? Nothing

Remove context from whitelist. For permanent operation see

org.fedoraproject.FirewallD1.config.policies.Methods.removeLockdownWhitelistContext.

Possible errors: NOT_ENABLED

removeLockdownWhitelistUid(i: uid) ? Nothing

Remove user id uid from whitelist. For permanent operation see

org.fedoraproject.FirewallD1.config.policies.Methods.removeLockdownWhitelistUid.

Possible errors: NOT_ENABLED

removeLockdownWhitelistUser(s: user) ? Nothing

Remove user from whitelist. For permanent operation see

org.fedoraproject.FirewallD1.config.policies.Methods.removeLockdownWhitelistUser.

Possible errors: NOT_ENABLED

Signals

LockdownDisabled()

Emitted when lockdown has been disabled.

LockdownEnabled()

Emitted when lockdown has been enabled.

LockdownWhitelistCommandAdded(s: command)

Emitted when command has been added to whitelist.

LockdownWhitelistCommandRemoved(s: command)

Emitted when command has been removed from whitelist.

LockdownWhitelistContextAdded(s: context)

Emitted when context has been added to whitelist.

LockdownWhitelistContextRemoved(s: context)

Emitted when context has been removed from whitelist.

LockdownWhitelistUidAdded(i: uid)

Emitted when user id uid has been added to whitelist.

LockdownWhitelistUidRemoved(i: uid)

Emitted when user id uid has been removed from whitelist.

LockdownWhitelistUserAdded(s: user)

Emitted when user has been added to whitelist.

LockdownWhitelistUserRemoved(s: user)

Emitted when user has been removed from whitelist.

org.fedoraproject.FirewallD1.zone

Operations in this interface allows one to get, add, remove and query runtime zone's settings. For permanent settings see org.fedoraproject.FirewallD1.config.zone interface.

Methods

getZoneSettings2(s: zone) ? a{sv}

Return runtime settings of given zone. For getting permanent settings see org.fedoraproject.FirewallD1.config.zone.Methods.getSettings2. Settings are a dictionary indexed by keywords. For the type of each value see below. If the value is empty it may be omitted.

version (s): see version attribute of zone tag in firewalld.zone(5).

name (s): see short tag in firewalld.zone(5).

description (s): see description tag in firewalld.zone(5).

target (s): see target attribute of zone tag in firewalld.zone(5).

services (as): array of service names, see service tag in firewalld.zone(5).

ports (a(ss)): array of port and protocol pairs. See port tag

in firewalld.zone(5).

icmp_blocks (as): array of icmp-blocks. See icmp-block tag in firewalld.zone(5).

masquerade (b): see masquerade tag in firewalld.zone(5).

forward_ports (a(ssss)): array of (port, protocol, to-port, to-addr). See forward-port tag in firewalld.zone(5).

interfaces (as): array of interfaces. See interface tag in firewalld.zone(5).

sources (as): array of source addresses. See source tag in firewalld.zone(5).

rules_str (as): array of rich-language rules. See rule tag in firewalld.zone(5).

protocols (as): array of protocols, see protocol tag in firewalld.zone(5).

source_ports (a(ss)): array of port and protocol pairs. See source-port tag in firewalld.zone(5).

icmp_block_inversion (b): see icmp-block-inversion tag in firewalld.zone(5).

forward (b): see forward tag in firewalld.zone(5).

Possible errors: INVALID_ZONE

setZoneSettings2(s: zone, a{sv}: settings, i: timeout)

Set runtime settings of given zone. For setting permanent settings see org.fedoraproject.FirewallD1.config.zone.Methods.update2. Settings are a dictionary indexed by keywords. For the type of each value see below. To zero a value pass an empty string or list.

services (as): array of service names, see service tag in firewalld.zone(5).

ports (a(ss)): array of port and protocol pairs. See port tag in firewalld.zone(5).

icmp_blocks (as): array of icmp-blocks. See icmp-block tag in firewalld.zone(5).

masquerade (b): see masquerade tag in firewalld.zone(5).

forward_ports (a(ssss)): array of (port, protocol, to-port,

to-addr). See forward-port tag in firewalld.zone(5).

interfaces (as): array of interfaces. See interface tag in

firewalld.zone(5).

sources (as): array of source addresses. See source tag in

firewalld.zone(5).

rules_str (as): array of rich-language rules. See rule tag in

firewalld.zone(5).

protocols (as): array of protocols, see protocol tag in

firewalld.zone(5).

source_ports (a(ss)): array of port and protocol pairs. See

source-port tag in firewalld.zone(5).

icmp_block_inversion (b): see icmp-block-inversion tag in

firewalld.zone(5).

forward (b): see forward tag in firewalld.zone(5).

Possible errors: INVALID_ZONE

addForwardPort(s: zone, s: port, s: protocol, s: toport, s: toaddr,

i: timeout) ? s

Add the IPv4 forward port into zone. If zone is empty, use

default zone. The port can either be a single port number

portid or a port range portid-portid. The protocol can either

be tcp or udp. The destination address is a simple IP address.

If timeout is non-zero, the operation will be active only for

the amount of seconds. For permanent settings see

org.fedoraproject.FirewallD1.config.zone.Methods.addForwardPort.

Returns name of zone to which the forward port was added.

Possible errors: INVALID_ZONE, INVALID_PORT, MISSING_PROTOCOL,

INVALID_PROTOCOL, INVALID_ADDR, INVALID_FORWARD,

ALREADY_ENABLED, INVALID_COMMAND

addIcmpBlock(s: zone, s: icmp, i: timeout) ? s

Add an ICMP block icmp into zone. The icmp is the one of the

icmp types firewalld supports. To get a listing of supported

icmp types use

org.fedoraproject.FirewallD1.Methods.listIcmpTypes If zone is

empty, use default zone. If timeout is non-zero, the operation

will be active only for the amount of seconds. For permanent

settings see

org.fedoraproject.FirewallD1.config.zone.Methods.addIcmpBlock.

Returns name of zone to which the ICMP block was added.

Possible errors: INVALID_ZONE, INVALID_ICMPTYPE,

ALREADY_ENABLED, INVALID_COMMAND

addIcmpBlockInversion(s: zone) ? s

Add ICMP block inversion to zone. If zone is empty, use default

zone. For permanent settings see

org.fedoraproject.FirewallD1.config.zone.Methods.addIcmpBlockInversion.

Returns name of zone to which the ICMP block inversion was

added.

Possible errors: INVALID_ZONE, ALREADY_ENABLED, INVALID_COMMAND

addInterface(s: zone, s: interface) ? s

Bind interface with zone. From now on all traffic going through

the interface will respect the zone's settings. If zone is

empty, use default zone. For permanent settings see

org.fedoraproject.FirewallD1.config.zone.Methods.addInterface.

Returns name of zone to which the interface was bound.

Possible errors: INVALID_ZONE, INVALID_INTERFACE,

ALREADY_ENABLED, INVALID_COMMAND

addMasquerade(s: zone, i: timeout) ? s

Enable masquerade in zone. If zone is empty, use default zone.

If timeout is non-zero, masquerading will be active for the

amount of seconds. For permanent settings see

org.fedoraproject.FirewallD1.config.zone.Methods.addMasquerade.

Returns name of zone in which the masquerade was enabled.

Possible errors: INVALID_ZONE, ALREADY_ENABLED, INVALID_COMMAND

addPort(s: zone, s: port, s: protocol, i: timeout) ? s

Add port into zone. If zone is empty, use default zone. The

port can either be a single port number or a port range
portid-portid. The protocol can either be tcp or udp. If
timeout is non-zero, the operation will be active only for the
amount of seconds. For permanent settings see
org.fedoraproject.FirewallD1.config.zone.Methods.addPort.
Returns name of zone to which the port was added.
Possible errors: INVALID_ZONE, INVALID_PORT, MISSING_PROTOCOL,
INVALID_PROTOCOL, ALREADY_ENABLED, INVALID_COMMAND

addProtocol(s: zone, s: protocol, i: timeout) ? s

Add protocol into zone. If zone is empty, use default zone. The
protocol can be any protocol supported by the system. Please
have a look at /etc/protocols for supported protocols. If
timeout is non-zero, the operation will be active only for the
amount of seconds. For permanent settings see
org.fedoraproject.FirewallD1.config.zone.Methods.addProtocol.
Returns name of zone to which the protocol was added.
Possible errors: INVALID_ZONE, INVALID_PROTOCOL,
ALREADY_ENABLED, INVALID_COMMAND

addRichRule(s: zone, s: rule, i: timeout) ? s

Add rich language rule into zone. For the rich language rule
syntax, please have a look at firewalld.direct(5). If zone is
empty, use default zone. If timeout is non-zero, the operation
will be active only for the amount of seconds. For permanent
settings see
org.fedoraproject.FirewallD1.config.zone.Methods.addRichRule.
Returns name of zone to which the rich language rule was added.
Possible errors: INVALID_ZONE, INVALID_RULE, ALREADY_ENABLED,
INVALID_COMMAND

addService(s: zone, s: service, i: timeout) ? s

Add service into zone. If zone is empty, use default zone. If
timeout is non-zero, the operation will be active only for the
amount of seconds. To get a list of supported services, use
org.fedoraproject.FirewallD1.Methods.listServices. For

permanent settings see
org.fedoraproject.FirewallD1.config.zone.Methods.addService.
Returns name of zone to which the service was added.
Possible errors: INVALID_ZONE, INVALID_SERVICE,
ALREADY_ENABLED, INVALID_COMMAND

addSource(s: zone, s: source) ? s

Bind source with zone. From now on all traffic going from this
source will respect the zone's settings. A source address or
address range is either an IP address or a network IP address
with a mask for IPv4 or IPv6. For IPv4, the mask can be a
network mask or a plain number. For IPv6 the mask is a plain
number. Use of host names is not supported. If zone is empty,
use default zone. For permanent settings see
org.fedoraproject.FirewallD1.config.zone.Methods.addSource.
Returns name of zone to which the source was bound.
Possible errors: INVALID_ZONE, INVALID_ADDR, ALREADY_ENABLED,
INVALID_COMMAND

addSourcePort(s: zone, s: port, s: protocol, i: timeout) ? s

Add source port into zone. If zone is empty, use default zone.
The port can either be a single port number or a port range
portid-portid. The protocol can either be tcp or udp. If
timeout is non-zero, the operation will be active only for the
amount of seconds. For permanent settings see
org.fedoraproject.FirewallD1.config.zone.Methods.addSourcePort.
Returns name of zone to which the port was added.
Possible errors: INVALID_ZONE, INVALID_PORT, MISSING_PROTOCOL,
INVALID_PROTOCOL, ALREADY_ENABLED, INVALID_COMMAND

changeZone(s: zone, s: interface) ? s

This function is deprecated, use
org.fedoraproject.FirewallD1.zone.Methods.changeZoneOfInterface
instead.

changeZoneOfInterface(s: zone, s: interface) ? s

Change a zone an interface is bound to to zone. It's basically

removeInterface(interface) followed by addInterface(zone, interface). If interface has not been bound to a zone before, it behaves like addInterface. If zone is empty, use default zone.

Returns name of zone to which the interface was bound.

Possible errors: INVALID_ZONE, ZONE_ALREADY_SET, ZONE_CONFLICT

changeZoneOfSource(s: zone, s: source) ? s

Change a zone an source is bound to to zone. It's basically removeSource(source) followed by addSource(zone, source). If source has not been bound to a zone before, it behaves like addSource. If zone is empty, use default zone.

Returns name of zone to which the source was bound.

Possible errors: INVALID_ZONE, ZONE_ALREADY_SET, ZONE_CONFLICT

getActiveZones() ? a{sa{sas}}

Return dictionary of currently active zones altogether with interfaces and sources used in these zones. Active zones are zones, that have a binding to an interface or source.

Return value is a dictionary where keys are zone names (s) and values are again dictionaries where keys are either 'interfaces' or 'sources' and values are arrays of interface names (s) or sources (s).

getForwardPorts(s: zone) ? aas

Return array of IPv4 forward ports previously added into zone. If zone is empty, use default zone. For getting permanent settings see org.fedoraproject.FirewallD1.config.zone.Methods.getForwardPorts.

Return value is array of 4-tuples, where each 4-tuple consists of (port, protocol, to-port, to-addr). to-addr might be empty in case of local forwarding.

Possible errors: INVALID_ZONE

getIcmpBlocks(s: zone) ? as

Return array of ICMP type (s) blocks previously added into zone. If zone is empty, use default zone. For getting permanent

settings see

org.fedoraproject.FirewallD1.config.zone.Methods.getIcmpBlocks.

Possible errors: INVALID_ZONE

getIcmpBlockInversion(s: zone) ? b

Return whether ICMP block inversion was previously added to

zone. If zone is empty, use default zone. For getting permanent

settings see

org.fedoraproject.FirewallD1.config.zone.Methods.getIcmpBlockInversion.

Possible errors: INVALID_ZONE

getInterfaces(s: zone) ? as

Return array of interfaces (s) previously bound with zone. If

zone is empty, use default zone. For getting permanent settings

see

org.fedoraproject.FirewallD1.config.zone.Methods.getInterfaces.

Possible errors: INVALID_ZONE

getPorts(s: zone) ? aas

Return array of ports (2-tuple of port and protocol) previously

enabled in zone. If zone is empty, use default zone. For

getting permanent settings see

org.fedoraproject.FirewallD1.config.zone.Methods.getPorts.

Possible errors: INVALID_ZONE

getProtocols(s: zone) ? as

Return array of protocols (s) previously enabled in zone. If

zone is empty, use default zone. For getting permanent settings

see

org.fedoraproject.FirewallD1.config.zone.Methods.getProtocols.

Possible errors: INVALID_ZONE

getRichRules(s: zone) ? as

Return array of rich language rules (s) previously added into

zone. If zone is empty, use default zone. For getting permanent

settings see

org.fedoraproject.FirewallD1.config.zone.Methods.getRichRules.

Possible errors: INVALID_ZONE

getServices(s: zone) ? as

Return array of services (s) previously enabled in zone. If zone is empty, use default zone. For getting permanent settings see org.fedoraproject.FirewallD1.config.zone.Methods.getServices. Possible errors: INVALID_ZONE

getSourcePorts(s: zone) ? aas

Return array of source ports (2-tuple of port and protocol) previously enabled in zone. If zone is empty, use default zone. For getting permanent settings see org.fedoraproject.FirewallD1.config.zone.Methods.getSourcePorts. Possible errors: INVALID_ZONE

getSources(s: zone) ? as

Return array of sources (s) previously bound with zone. If zone is empty, use default zone. For getting permanent settings see org.fedoraproject.FirewallD1.config.zone.Methods.getSources. Possible errors: INVALID_ZONE

getZoneOfInterface(s: interface) ? s

Return name (s) of zone the interface is bound to or empty string.

getZoneOfSource(s: source) ? s

Return name (s) of zone the source is bound to or empty string.

getZones() ? as

Return array of names (s) of predefined zones known to current runtime environment. For list of zones known to permanent environment see org.fedoraproject.FirewallD1.config.Methods.listZones. The lists (of zones known to runtime and permanent environment) will contain same zones in most cases, but might differ for example if org.fedoraproject.FirewallD1.config.Methods.addZone has been called recently, but firewalld has not been reloaded since then.

isImmutable(s: zone) ? b

Deprecated.

queryForwardPort(s: zone, s: port, s: protocol, s: toport, s:

toaddr) ? b

Return whether the IPv4 forward port (port, protocol, toport,

toaddr) has been added into zone. If zone is empty, use default

zone. For permanent operation see

org.fedoraproject.FirewallD1.config.zone.Methods.queryForwardPort.

Possible errors: INVALID_ZONE, INVALID_PORT, MISSING_PROTOCOL,

INVALID_PROTOCOL, INVALID_ADDR, INVALID_FORWARD

queryIcmpBlock(s: zone, s: icmp) ? b

Return whether an ICMP block for icmp has been added into zone.

If zone is empty, use default zone. For permanent operation see

org.fedoraproject.FirewallD1.config.zone.Methods.queryIcmpBlock.

Possible errors: INVALID_ZONE, INVALID_ICMPTYPE

queryIcmpBlockInversion(s: zone) ? b

Return whether ICMP block inversion has been added to zone. If

zone is empty, use default zone. For permanent operation see

org.fedoraproject.FirewallD1.config.zone.Methods.queryIcmpBlockInversion.

Possible errors: INVALID_ZONE, INVALID_ICMPTYPE

queryInterface(s: zone, s: interface) ? b

Query whether interface has been bound to zone. If zone is

empty, use default zone. For permanent operation see

org.fedoraproject.FirewallD1.config.zone.Methods.queryInterface.

Possible errors: INVALID_ZONE, INVALID_INTERFACE

queryMasquerade(s: zone) ? b

Return whether masquerading has been enabled in zone If zone is

empty, use default zone. For permanent operation see

org.fedoraproject.FirewallD1.config.zone.Methods.queryMasquerade.

Possible errors: INVALID_ZONE

queryPort(s: zone, s: port, s: protocol) ? b

Return whether port/protocol has been added in zone. If zone is

empty, use default zone. For permanent operation see

org.fedoraproject.FirewallD1.config.zone.Methods.queryPort.

Possible errors: INVALID_ZONE, INVALID_PORT, MISSING_PROTOCOL, INVALID_PROTOCOL

queryProtocol(s: zone, s: protocol) ? b

Return whether protocol has been added in zone. If zone is empty, use default zone. For permanent operation see org.fedoraproject.FirewallD1.config.zone.Methods.queryProtocol. Possible errors: INVALID_ZONE, INVALID_PROTOCOL

queryRichRule(s: zone, s: rule) ? b

Return whether rich rule rule has been added in zone. If zone is empty, use default zone. For permanent operation see org.fedoraproject.FirewallD1.config.zone.Methods.queryRichRule. Possible errors: INVALID_ZONE, INVALID_RULE

queryService(s: zone, s: service) ? b

Return whether service has been added for zone. If zone is empty, use default zone. For permanent operation see org.fedoraproject.FirewallD1.config.zone.Methods.queryService. Possible errors: INVALID_ZONE, INVALID_SERVICE

querySource(s: zone, s: source) ? b

Query whether sourcehas been bound to zone. If zone is empty, use default zone. For permanent operation see org.fedoraproject.FirewallD1.config.zone.Methods.querySource. Possible errors: INVALID_ZONE, INVALID_ADDR

querySourcePort(s: zone, s: port, s: protocol) ? b

Return whether port/protocol has been added in zone. If zone is empty, use default zone. For permanent operation see org.fedoraproject.FirewallD1.config.zone.Methods.querySourcePort. Possible errors: INVALID_ZONE, INVALID_PORT, MISSING_PROTOCOL, INVALID_PROTOCOL

removeForwardPort(s: zone, s: port, s: protocol, s: toport, s: toaddr) ? s

Remove IPv4 forward port ((port, protocol, toport, toaddr)) from zone. If zone is empty, use default zone. For permanent operation see

org.fedoraproject.FirewallD1.config.zone.Methods.removeForwardPort.

Returns name of zone from which the forward port was removed.

Possible errors: INVALID_ZONE, INVALID_PORT, MISSING_PROTOCOL, INVALID_PROTOCOL, INVALID_ADDR, INVALID_FORWARD, NOT_ENABLED, INVALID_COMMAND

removeIcmpBlock(s: zone, s: icmp) ? s

Remove ICMP block icmp from zone. If zone is empty, use default zone. For permanent operation see org.fedoraproject.FirewallD1.config.zone.Methods.removeIcmpBlock.

Returns name of zone from which the ICMP block was removed.

Possible errors: INVALID_ZONE, INVALID_ICMPTYPE, NOT_ENABLED, INVALID_COMMAND

removeIcmpBlockInversion(s: zone) ? s

Remove ICMP block inversion from zone. If zone is empty, use default zone. For permanent operation see org.fedoraproject.FirewallD1.config.zone.Methods.removeIcmpBlockInversion.

Returns name of zone from which the ICMP block inversion was removed.

Possible errors: INVALID_ZONE, NOT_ENABLED, INVALID_COMMAND

removeInterface(s: zone, s: interface) ? s

Remove binding of interface from zone. If zone is empty, the interface will be removed from zone it belongs to. For permanent operation see org.fedoraproject.FirewallD1.config.zone.Methods.removeInterface.

Returns name of zone from which the interface was removed.

Possible errors: INVALID_ZONE, INVALID_INTERFACE, NOT_ENABLED, INVALID_COMMAND

removeMasquerade(s: zone) ? s

Disable masquerade for zone. If zone is empty, use default zone. For permanent operation see org.fedoraproject.FirewallD1.config.zone.Methods.removeMasquerade.

Returns name of zone for which the masquerade was disabled.

Possible errors: INVALID_ZONE, NOT_ENABLED, INVALID_COMMAND

removePort(s: zone, s: port, s: protocol) ? s

Remove port/protocol from zone. If zone is empty, use default zone. For permanent operation see org.fedoraproject.FirewallD1.config.zone.Methods.removePort. Returns name of zone from which the port was removed. Possible errors: INVALID_ZONE, INVALID_PORT, MISSING_PROTOCOL, INVALID_PROTOCOL, NOT_ENABLED, INVALID_COMMAND

removeProtocol(s: zone, s: protocol) ? s

Remove protocol from zone. If zone is empty, use default zone. For permanent operation see org.fedoraproject.FirewallD1.config.zone.Methods.removeProtocol. Returns name of zone from which the protocol was removed. Possible errors: INVALID_ZONE, INVALID_PROTOCOL, NOT_ENABLED, INVALID_COMMAND

removeRichRule(s: zone, s: rule) ? s

Remove rich language rule from zone. If zone is empty, use default zone. For permanent operation see org.fedoraproject.FirewallD1.config.zone.Methods.removeRichRule. Returns name of zone from which the rich language rule was removed. Possible errors: INVALID_ZONE, INVALID_RULE, NOT_ENABLED, INVALID_COMMAND

removeService(s: zone, s: service) ? s

Remove service from zone. If zone is empty, use default zone. For permanent operation see org.fedoraproject.FirewallD1.config.zone.Methods.removeService. Returns name of zone from which the service was removed. Possible errors: INVALID_ZONE, INVALID_SERVICE, NOT_ENABLED, INVALID_COMMAND

removeSource(s: zone, s: source) ? s

Remove binding of source from zone. If zone is empty, the source will be removed from zone it belongs to. For permanent operation see

org.fedoraproject.FirewallD1.config.zone.Methods.removeSource.

Returns name of zone from which the source was removed.

Possible errors: INVALID_ZONE, INVALID_ADDR, NOT_ENABLED, INVALID_COMMAND

removeSourcePort(s: zone, s: port, s: protocol) ? s

Remove port/protocol from zone. If zone is empty, use default zone. For permanent operation see org.fedoraproject.FirewallD1.config.zone.Methods.removeSourcePort.

Returns name of zone from which the source port was removed.

Possible errors: INVALID_ZONE, INVALID_PORT, MISSING_PROTOCOL, INVALID_PROTOCOL, NOT_ENABLED, INVALID_COMMAND

Signals

ForwardPortAdded(s: zone, s: port, s: protocol, s: toport, s: toaddr, i: timeout)

Emitted when forward port has been added to zone with timeout.

ForwardPortRemoved(s: zone, s: port, s: protocol, s: toport, s: toaddr)

Emitted when forward port has been removed from zone.

IcmpBlockAdded(s: zone, s: icmp, i: timeout)

Emitted when ICMP block for icmp has been added to zone with timeout.

IcmpBlockInversionAdded(s: zone)

Emitted when ICMP block inversion has been added to zone.

IcmpBlockInversionRemoved(s: zone)

Emitted when ICMP block inversion has been removed from zone.

IcmpBlockRemoved(s: zone, s: icmp)

Emitted when ICMP block for icmp has been removed from zone.

InterfaceAdded(s: zone, s: interface)

Emitted when interface has been added to zone.

InterfaceRemoved(s: zone, s: interface)

Emitted when interface has been removed from zone.

MasqueradeAdded(s: zone, i: timeout)

Emitted when masquerade has been enabled for zone.

MasqueradeRemoved(s: zone)

Emitted when masquerade has been disabled for zone.

PortAdded(s: zone, s: port, s: protocol, i: timeout)

Emitted when port/protocol has been added to zone with timeout.

PortRemoved(s: zone, s: port, s: protocol)

Emitted when port/protocol has been removed from zone.

ProtocolAdded(s: zone, s: protocol, i: timeout)

Emitted when protocol has been added to zone with timeout.

ProtocolRemoved(s: zone, s: protocol)

Emitted when protocol has been removed from zone.

RichRuleAdded(s: zone, s: rule, i: timeout)

Emitted when rich language rule has been added to zone with timeout.

RichRuleRemoved(s: zone, s: rule)

Emitted when rich language rule has been removed from zone.

ServiceAdded(s: zone, s: service, i: timeout)

Emitted when service has been added to zone with timeout.

ServiceRemoved(s: zone, s: service)

Emitted when service has been removed from zone.

SourceAdded(s: zone, s: source)

Emitted when source has been added to zone.

SourcePortAdded(s: zone, s: port, s: protocol, i: timeout)

Emitted when source-port/protocol has been added to zone with timeout.

SourcePortRemoved(s: zone, s: port, s: protocol)

Emitted when source-port/protocol has been removed from zone.

SourceRemoved(s: zone, s: source)

Emitted when source has been removed from zone.

ZoneChanged(s: zone, s: interface)

Deprecated

ZoneOfInterfaceChanged(s: zone, s: interface)

Emitted when a zone an interface is part of has been changed to zone.

ZoneOfSourceChanged(s: zone, s: source)

Emitted when a zone an source is part of has been changed to zone.

ZoneUpdated2(s: zone, a{sv}: settings)

Emitted when a zone's settings are updated via org.fedoraproject.FirewallD1.zone.Methods.setZoneSettings2

org.fedoraproject.FirewallD1.policy

Operations in this interface allows one to get, add, remove and query runtime policy settings. For permanent settings see org.fedoraproject.FirewallD1.config.policy interface.

Methods

getActivePolicies() ? a{sa{sas}}

Return dictionary of currently active policies altogether with ingress zones and egress zones used in these policies. Active policies are policies, that have a binding to an active ingress zone and an active egress zone.

Return value is a dictionary where keys are policy names (s) and values are again dictionaries where keys are either 'ingress_zones' or 'egress_zones' and values are arrays of zone names (s).

getPolicies() ? as

Return array of names (s) of predefined policies known to current runtime environment. For list of policies known to permanent environment see org.fedoraproject.FirewallD1.config.Methods.listPolicies. The lists (of policies known to runtime and permanent environment) will contain same policies in most cases, but might differ for example if org.fedoraproject.FirewallD1.config.Methods.addPolicy has been called recently, but firewalld has not been reloaded since then.

getPolicySettings(s: policy) ? a{sv}

Return runtime settings of given policy. For getting permanent

settings see

org.fedoraproject.FirewallD1.config.policy.Methods.getSettings.

Settings are a dictionary indexed by keywords. For possible

keywords see

org.fedoraproject.FirewallD1.config.Methods.addPolicy. If the

value is empty it may be omitted.

Possible errors: INVALID_POLICY

setPolicySettings(s: policy, a{sv}: settings, i: timeout)

Set runtime settings of given policy. For setting permanent

settings see

org.fedoraproject.FirewallD1.config.policy.Methods.update.

Settings are a dictionary indexed by keywords. For possible

keywords see

org.fedoraproject.FirewallD1.config.Methods.addPolicy. To zero

a value pass an empty string or list. Some keywords are not

available to modify in the runtime: description, name,

priority, target, version.

Possible errors: INVALID_POLICY

Signals

ForwardPortAdded(s: policy, a{sv}: settings)

Emitted when a policy's settings are updated via

org.fedoraproject.FirewallD1.policy.Methods.setPolicySettings

org.fedoraproject.FirewallD1.config

Allows one to permanently add, remove and query zones, services and

icmp types.

Methods

addIPSet(s: ipset, (ssssa{ss}as): settings) ? o

Add ipset with given settings into permanent configuration.

Settings are in format: version, name, description, type,

dictionary of options and array of entries.

version (s): see version attribute of ipset tag in

firewalld.ipset(5).

name (s): see short tag in firewalld.ipset(5).

description (s): see description tag in firewalld.ipset(5).

type (s): see type attribute of ipset tag in

firewalld.ipset(5).

options (a{ss}): dictionary of {option : value} . See options

tag in firewalld.ipset(5).

entries (as): array of entries, see entry tag in

firewalld.ipset(5).

Possible errors: NAME_CONFLICT, INVALID_NAME, INVALID_TYPE

addIcmpType(s: icmptype, (sssas): settings) ? o

Add icmptype with given settings into permanent configuration.

Settings are in format: version, name, description, array of

destinations. Returns object path of the new icmp type.

version (s): see version attribute of icmptype tag in

firewalld.icmptype(5).

name (s): see short tag in firewalld.icmptype(5).

description (s): see description tag in firewalld.icmptype(5).

destinations (as): array, either empty or containing strings

'ipv4' or 'ipv6', see destination tag in firewalld.icmptype(5).

Possible errors: NAME_CONFLICT, INVALID_NAME, INVALID_TYPE

addService(s: service, (sssa(ss)asa{ss}asa(ss)): settings) ? o

This function is deprecated, use

org.fedoraproject.FirewallD1.config.Methods.addService2

instead.

addService2s: service, a{sv}: settings) ? o

Add service with given settings into permanent configuration.

Settings are a dictionary indexed by keywords. For the type of

each value see below. To zero a value pass an empty string or

list.

version (s): see version attribute of service tag in

firewalld.service(5).

name (s): see short tag in firewalld.service(5).

description (s): see description tag in firewalld.service(5).

ports (a(ss)): array of port and protocol pairs. See port tag

in firewalld.service(5).

module names (as): array of kernel netfilter helpers, see

module tag in firewalld.service(5).

destinations (a{ss}): dictionary of {IP family : IP address}

where 'IP family' key can be either 'ipv4' or 'ipv6'. See

destination tag in firewalld.service(5).

protocols (as): array of protocols, see protocol tag in

firewalld.service(5).

source_ports (a(ss)): array of port and protocol pairs. See

source-port tag in firewalld.service(5).

includes (as): array of service includes, see include tag in

firewalld.service(5).

helpers (as): array of service helpers, see helper tag in

firewalld.service(5).

Possible errors: NAME_CONFLICT, INVALID_NAME, INVALID_TYPE

addZone(s: zone, (sssbsasa(ss)asba(ssss)asasasasa(ss)b): settings)

? o

This function is deprecated, use

org.fedoraproject.FirewallD1.config.Methods.addZone2 instead.

addZone2(s: zone, a{sv}: settings) ? o

Add zone with given settings into permanent configuration.

Settings are a dictionary indexed by keywords. For the type of

each value see below. To zero a value pass an empty string or

list.

version (s): see version attribute of zone tag in

firewalld.zone(5).

name (s): see short tag in firewalld.zone(5).

description (s): see description tag in firewalld.zone(5).

target (s): see target attribute of zone tag in

firewalld.zone(5).

services (as): array of service names, see service tag in

firewalld.zone(5).

ports (a(ss)): array of port and protocol pairs. See port tag

in firewalld.zone(5).

icmp_blocks (as): array of icmp-blocks. See icmp-block tag in firewalld.zone(5).

masquerade (b): see masquerade tag in firewalld.zone(5).

forward_ports (a(ssss)): array of (port, protocol, to-port, to-addr). See forward-port tag in firewalld.zone(5).

interfaces (as): array of interfaces. See interface tag in firewalld.zone(5).

sources (as): array of source addresses. See source tag in firewalld.zone(5).

rules_str (as): array of rich-language rules. See rule tag in firewalld.zone(5).

protocols (as): array of protocols, see protocol tag in firewalld.zone(5).

source_ports (a(ss)): array of port and protocol pairs. See source-port tag in firewalld.zone(5).

icmp_block_inversion (b): see icmp-block-inversion tag in firewalld.zone(5).

forward (b): see forward tag in firewalld.zone(5).

Possible errors: NAME_CONFLICT, INVALID_NAME, INVALID_TYPE

addPolicy(s: policy, a{sv}: settings) ? o

Add policy with given settings into permanent configuration.

Settings are a dictionary indexed by keywords. For the type of each value see below. If a keyword is omitted the default value will be used.

description (s): see description tag in firewalld.policy(5).

egress_zones as: array of zone names. See egress-zone tag in firewalld.policy(5).

forward_ports (a(ssss)): array of (port, protocol, to-port, to-addr). See forward-port tag in firewalld.policy(5).

icmp_blocks (as): array of icmp-blocks. See icmp-block tag in firewalld.policy(5).

ingress_zones as: array of zone names. See ingress-zone tag in

firewalld.policy(5).

masquerade (b): see masquerade tag in firewalld.policy(5).

ports (a(ss)): array of port and protocol pairs. See port tag in firewalld.policy(5).

priority (i): see priority tag in firewalld.policy(5).

protocols (as): array of protocols, see protocol tag in firewalld.policy(5).

rich_rules (as): array of rich-language rules. See rule tag in firewalld.policy(5).

services (as): array of service names, see service tag in firewalld.policy(5).

short (s): see short tag in firewalld.policy(5).

source_ports (a(ss)): array of port and protocol pairs. See source-port tag in firewalld.policy(5).

target (s): see target attribute of policy tag in firewalld.policy(5).

version (s): see version attribute of policy tag in firewalld.policy(5).

Possible errors: NAME_CONFLICT, INVALID_NAME, INVALID_TYPE

getHelperByName(s: helper) ? o

Return object path (permanent configuration) of helper with given name.

Possible errors: INVALID_HELPER

getHelperNames() ? as

Return list of helper names (permanent configuration).

getIPSetByName(s: ipset) ? o

Return object path (permanent configuration) of ipset with given name.

Possible errors: INVALID_IPSET

getIPSetNames() ? as

Return list of ipset names (permanent configuration).

getIcmpTypeByName(s: icmptype) ? o

Return object path (permanent configuration) of icmptype with

given name.

Possible errors: INVALID_ICMPTYPE

getIcmpTypeNames() ? as

Return list of icmptype names (permanent configuration).

getServiceByName(s: service) ? o

Return object path (permanent configuration) of service with given name.

Possible errors: INVALID_SERVICE

getServiceNames() ? as

Return list of service names (permanent configuration).

getZoneByName(s: zone) ? o

Return object path (permanent configuration) of zone with given name.

Possible errors: INVALID_ZONE

getZoneNames() ? as

Return list of zone names (permanent configuration) of.

getZoneOfInterface(s: iface) ? s

Return name of zone the iface is bound to or empty string.

getZoneOfSource(s: source) ? s

Return name of zone the source is bound to or empty string.

getPolicyByName(s: policy) ? o

Return object path (permanent configuration) of policy with given name.

Possible errors: INVALID_POLICY

getPolicyNames() ? as

Return list of policy names (permanent configuration).

listHelpers() ? ao

Return array of object paths (o) of helper in permanent configuration. For runtime configuration see org.fedoraproject.FirewallD1.Methods.getHelpers.

listIPSets() ? ao

Return array of object paths (o) of ipset in permanent configuration. For runtime configuration see

org.fedoraproject.FirewallD1.ipset.Methods.getIPSets.

listIcmpTypes() ? ao

Return array of object paths (o) of icmp types in permanent configuration. For runtime configuration see org.fedoraproject.FirewallD1.Methods.listIcmpTypes.

listServices() ? ao

Return array of objects paths (o) of services in permanent configuration. For runtime configuration see org.fedoraproject.FirewallD1.Methods.listServices.

listZones() ? ao

List object paths of zones known to permanent environment. For list of zones known to runtime environment see org.fedoraproject.FirewallD1.zone.Methods.getZones. The lists (of zones known to runtime and permanent environment) will contain same zones in most cases, but might differ for example if org.fedoraproject.FirewallD1.config.Methods.addZone has been called recently, but firewalld has not been reloaded since then.

listPolicies() ? ao

List object paths of policies known to permanent environment. For list of policies known to runtime environment see org.fedoraproject.FirewallD1.policy.Methods.getPolicies. The lists (of policies known to runtime and permanent environment) will contain same policies in most cases, but might differ for example if org.fedoraproject.FirewallD1.config.Methods.addPolicy has been called recently, but firewalld has not been reloaded since then.

Signals

HelperAdded(s: helper)

Emitted when helper has been added.

IPSetAdded(s: ipset)

Emitted when ipset has been added.

IcmpTypeAdded(s: icmptype)

Emitted when icmptype has been added.

ServiceAdded(s: service)

Emitted when service has been added.

ZoneAdded(s: zone)

Emitted when zone has been added.

Properties

AllowZoneDrifting - s - (rw)

Deprecated. Getting this value always returns "no". Setting this value is ignored.

AutomaticHelpers - s - (rw)

Deprecated. Getting this value always returns "no". Setting this value is ignored.

CleanupModulesOnExit - s - (rw)

Setting this option to yes or true unloads all firewall-related kernel modules when firewalld is stopped.

CleanupOnExit - s - (rw)

If firewalld stops, it cleans up all firewall rules. Setting this option to no or false leaves the current firewall rules untouched.

DefaultZone - s - (ro)

Default zone for connections or interfaces if the zone is not selected or specified by NetworkManager, initscripts or command line tool.

FirewallBackend - s - (rw)

Selects the firewalld backend for all rules except the direct interface. Valid options are; nftables, iptables. Default in nftables.

Note: The iptables backend is deprecated. It will be removed in a future release.

FlushAllOnReload - s - (rw)

Flush all runtime rules on a reload. Valid options are; yes, no.

IPv6_rpfilter - s - (rw)

Indicates whether the reverse path filter test on a packet for IPv6 is enabled. If a reply to the packet would be sent via the same interface that the packet arrived on, the packet will match and be accepted, otherwise dropped.

IndividualCalls - s - (ro)

Indicates whether individual calls combined -restore calls are used. If enabled, this increases the time that is needed to apply changes and to start the daemon, but is good for debugging.

Lockdown - s - (rw)

If this property is enabled, firewall changes with the D-Bus interface will be limited to applications that are listed in the lockdown whitelist.

LogDenied - s - (rw)

If LogDenied is enabled, then logging rules are added 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. Possible values are: all, unicast, broadcast, multicast and off.

MinimalMark - i - (rw)

Deprecated. This option is ignored and no longer used. Marks are no longer used internally.

RFC3964_IPv4 - s - (rw)

As per RFC 3964, filter IPv6 traffic with 6to4 destination addresses that correspond to IPv4 addresses that should not be routed over the public internet. Valid options are; yes, no.

org.fedoraproject.FirewallD1.config.direct

DEPRECATED

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

Interface for permanent direct configuration, see also

firewalld.direct(5). For runtime direct configuration see org.fedoraproject.FirewallD1.direct interface.

Methods

addChain(s: ipv, s: table, s: chain) ? Nothing

Add a new chain to table for ipv being either ipv4 (iptables) or ipv6 (ip6tables) or eb (ebtables). Make sure there's no other chain with this name already. There already exist basic chains to use with direct methods, for example INPUT_direct chain. These chains are jumped into before chains for zones, i.e. every rule put into INPUT_direct will be checked before rules in zones. For runtime operation see org.fedoraproject.FirewallD1.direct.Methods.addChain. Possible errors: INVALID_IPV, INVALID_TABLE, ALREADY_ENABLED

addPassthrough(s: ipv, as: args) ? Nothing

Add a passthrough rule with the arguments args for ipv being either ipv4 (iptables) or ipv6 (ip6tables) or eb (ebtables). For runtime operation see org.fedoraproject.FirewallD1.direct.Methods.addPassthrough. Possible errors: INVALID_IPV, ALREADY_ENABLED

addRule(s: ipv, s: table, s: chain, i: priority, as: args) ? Nothing

Add a rule with the arguments args to chain in table with priority for ipv being either ipv4 (iptables) or ipv6 (ip6tables) or eb (ebtables). 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. For runtime operation see org.fedoraproject.FirewallD1.direct.Methods.addRule. Possible errors: INVALID_IPV, INVALID_TABLE, ALREADY_ENABLED

getAllChains() ? a(sss)

Get all chains added to all tables in format: ipv, table, chain. This concerns only chains previously added with addChain. Return value is a array of (ipv, table, chain). For runtime operation see org.fedoraproject.FirewallD1.direct.Methods.getAllChains.

ipv (s): either ipv4 (iptables) or ipv6 (ip6tables) or eb (ebtables).

table (s): one of filter, mangle, nat, raw, security

chain (s): name of a chain.

getAllPassthroughs() ? a(sas)

Get all passthrough rules added in all ipv types in format: ipv, rule. This concerns only rules previously added with addPassthrough. Return value is a array of (ipv, array of arguments). For runtime operation see org.fedoraproject.FirewallD1.direct.Methods.getAllPassthroughs.

ipv (s): either ipv4 (iptables) or ipv6 (ip6tables) or eb (ebtables).

arguments (as): array of commands, parameters and other iptables/ip6tables/ebtables command line options.

getAllRules() ? a(sssias)

Get all rules added to all chains in all tables in format: ipv, table, chain, priority, rule. This concerns only rules previously added with addRule. Return value is a array of (ipv, table, chain, priority, array of arguments). For runtime operation see org.fedoraproject.FirewallD1.direct.Methods.getAllRules.

ipv (s): either ipv4 (iptables) or ipv6 (ip6tables) or eb (ebtables).

table (s): one of filter, mangle, nat, raw, security

chain (s): name of a chain.

priority (i): used to order rules.

arguments (as): array of commands, parameters and other

iptables/ip6tables/ebtables command line options.

getChains(s: ipv, s: table) ? as

Return an array of chains (s) added to table for ipv being either ipv4 (iptables) or ipv6 (ip6tables) or eb (ebtables). This concerns only chains previously added with addChain. For runtime operation see org.fedoraproject.FirewallD1.direct.Methods.getChains. Possible errors: INVALID_IPV, INVALID_TABLE

getPassthroughs(s: ipv) ? aas

Get tracked passthrough rules added in either ipv4 (iptables) or ipv6 (ip6tables) or eb (ebtables). This concerns only rules previously added with addPassthrough. Return value is a array of (array of arguments). For runtime operation see org.fedoraproject.FirewallD1.direct.Methods.getPassthroughs. arguments (as): array of commands, parameters and other iptables/ip6tables/ebtables command line options.

getRules(s: ipv, s: table, s: chain) ? a(ias)

Get all rules added to chain in table for ipv being either ipv4 (iptables) or ipv6 (ip6tables) or eb (ebtables). This concerns only rules previously added with addRule. Return value is a array of (priority, array of arguments). For runtime operation see org.fedoraproject.FirewallD1.direct.Methods.getRules. priority (i): used to order rules. arguments (as): array of commands, parameters and other iptables/ip6tables/ebtables command line options. Possible errors: INVALID_IPV, INVALID_TABLE

getSettings() ? (a(sss)a(sssias)a(sas))

Get settings of permanent direct configuration in format: array of chains, array of rules, array of passthroughs. chains (a(sss)): array of (ipv, table, chain), see 'chain' in firewalld.direct(5).

.

.PP rules (a(sssias)): array of (ipv, table,

chain, priority, array of arguments), see 'rule' in
firewalld.direct(5).
.
.PP passthroughs (a(sas)): array of (ipv,
array of arguments), see passthrough in firewalld.direct(5).
.
.sp
queryChain(s: ipv, s: table, s: chain) ? b
Return whether a chain exists in table for ipv being either
ipv4 (iptables) or ipv6 (ip6tables) or eb (ebtables). This
concerns only chains previously added with addChain. For
runtime operation see
org.fedoraproject.FirewallD1.direct.Methods.queryChain.
Possible errors: INVALID_IPV, INVALID_TABLE
queryPassthrough(s: ipv, as: args) ? b
Return whether a tracked passthrough rule with the arguments
args exists for ipv being either ipv4 (iptables) or ipv6
(ip6tables) or eb (ebtables). This concerns only rules
previously added with addPassthrough. For runtime operation see
org.fedoraproject.FirewallD1.direct.Methods.queryPassthrough.
Possible errors: INVALID_IPV
queryRule(s: ipv, s: table, s: chain, i: priority, as: args) ? b
Return whether a rule with priority and the arguments args
exists in chain in table for ipv being either ipv4 (iptables)
or ipv6 (ip6tables) or eb (ebtables). This concerns only rules
previously added with addRule. For runtime operation see
org.fedoraproject.FirewallD1.direct.Methods.queryRule.
Possible errors: INVALID_IPV, INVALID_TABLE
removeChain(s: ipv, s: table, s: chain) ? Nothing
Remove a chain from table for ipv being either ipv4 (iptables)
or ipv6 (ip6tables) or eb (ebtables). Only chains previously
added with addChain can be removed this way. For runtime
operation see

org.fedoraproject.FirewallD1.direct.Methods.removeChain.

Possible errors: INVALID_IPV, INVALID_TABLE, NOT_ENABLED

removePassthrough(s: ipv, as: args) ? Nothing

Remove a passthrough rule with arguments args for ipv being either ipv4 (iptables) or ipv6 (ip6tables) or eb (ebtables).

Only rules previously added with addPassthrough can be removed this way. For runtime operation see org.fedoraproject.FirewallD1.direct.Methods.removePassthrough.

Possible errors: INVALID_IPV, NOT_ENABLED

removeRule(s: ipv, s: table, s: chain, i: priority, as: args) ? Nothing

Remove a rule with priority and arguments args from chain in table for ipv being either ipv4 (iptables) or ipv6 (ip6tables) or eb (ebtables). Only rules previously added with addRule can be removed this way. For runtime operation see org.fedoraproject.FirewallD1.direct.Methods.removeRule.

Possible errors: INVALID_IPV, INVALID_TABLE, NOT_ENABLED

removeRules(s: ipv, s: table, s: chain) ? Nothing

Remove all rules from chain in table for ipv being either ipv4 (iptables) or ipv6 (ip6tables) or eb (ebtables). This concerns only rules previously added with addRule. For runtime operation see org.fedoraproject.FirewallD1.direct.Methods.removeRules.

Possible errors: INVALID_IPV, INVALID_TABLE

update((a(sss)a(sssias)a(sas)): settings) ? Nothing

Update permanent direct configuration with given settings.

Settings are in format: array of chains, array of rules, array of passthroughs.

chains (a(sss)): array of (ipv, table, chain), see 'chain' in firewalld.direct(5).

.

.PP rules (a(sssias)): array of (ipv, table, chain, priority, array of arguments), see 'rule' in firewalld.direct(5).

.

.PP passthroughs (a(sas)): array of (ipv,

array of arguments), see passthrough in firewalld.direct(5).

.

.sp Possible errors: INVALID_TYPE

Signals

Updated()

Emitted when configuration has been updated.

org.fedoraproject.FirewallD1.config.policies

Interface for permanent lockdown-whitelist configuration, see also firewalld.lockdown-whitelist(5). For runtime configuration see org.fedoraproject.FirewallD1.policies interface.

Methods

addLockdownWhitelistCommand(s: command) ? Nothing

Add command to whitelist. See command option in firewalld.lockdown-whitelist(5). For runtime operation see org.fedoraproject.FirewallD1.policies.Methods.addLockdownWhitelistCommand. Possible errors: ALREADY_ENABLED, INVALID_TYPE

addLockdownWhitelistContext(s: context) ? Nothing

Add context to whitelist. See selinux option in firewalld.lockdown-whitelist(5). For runtime operation see org.fedoraproject.FirewallD1.policies.Methods.addLockdownWhitelistContext. Possible errors: ALREADY_ENABLED, INVALID_TYPE

addLockdownWhitelistUid(i: uid) ? Nothing

Add user id uid to whitelist. See user option in firewalld.lockdown-whitelist(5). For runtime operation see org.fedoraproject.FirewallD1.policies.Methods.addLockdownWhitelistUid. Possible errors: ALREADY_ENABLED, INVALID_TYPE

addLockdownWhitelistUser(s: user) ? Nothing

Add user name to whitelist. See user option in firewalld.lockdown-whitelist(5). For runtime operation see org.fedoraproject.FirewallD1.policies.Methods.addLockdownWhitelistUser. Possible errors: ALREADY_ENABLED, INVALID_TYPE

getLockdownWhitelist() ? (asasasai)

Get settings of permanent lockdown-whitelist configuration in

format: commands, selinux contexts, users, uids

commands (as): see command option in firewalld.lockdown-

whitelist(5).

selinux contexts (as): see selinux option in

firewalld.lockdown-whitelist(5).

users (as): see name attribute of user option in

firewalld.lockdown-whitelist(5).

uids (ai): see id attribute of user option in

firewalld.lockdown-whitelist(5).

getLockdownWhitelistCommands() ? as

List all command lines (s) that are on whitelist. For runtime

operation see

org.fedoraproject.FirewallD1.policies.Methods.getLockdownWhitelistCommands.

getLockdownWhitelistContexts() ? as

List all contexts (s) that are on whitelist. For runtime

operation see

org.fedoraproject.FirewallD1.policies.Methods.getLockdownWhitelistContexts.

getLockdownWhitelistUids() ? ai

List all user ids (i) that are on whitelist. For runtime

operation see

org.fedoraproject.FirewallD1.policies.Methods.getLockdownWhitelistUids.

getLockdownWhitelistUsers() ? as

List all users (s) that are on whitelist. For runtime operation

see

org.fedoraproject.FirewallD1.policies.Methods.getLockdownWhitelistUsers.

queryLockdownWhitelistCommand(s: command) ? b

Query whether command is on whitelist. For runtime operation

see

org.fedoraproject.FirewallD1.policies.Methods.queryLockdownWhitelistCommand.

queryLockdownWhitelistContext(s: context) ? b

Query whether context is on whitelist. For runtime operation

see

org.fedoraproject.FirewallD1.policies.Methods.queryLockdownWhitelistContext.

queryLockdownWhitelistUid(i: uid) ? b

Query whether user id uid is on whitelist. For runtime

operation see

org.fedoraproject.FirewallD1.policies.Methods.queryLockdownWhitelistUid.

queryLockdownWhitelistUser(s: user) ? b

Query whether user is on whitelist. For runtime operation see

org.fedoraproject.FirewallD1.policies.Methods.queryLockdownWhitelistUser.

removeLockdownWhitelistCommand(s: command) ? Nothing

Remove command from whitelist. For runtime operation see

org.fedoraproject.FirewallD1.policies.Methods.removeLockdownWhitelistCommand.

Possible errors: NOT_ENABLED

removeLockdownWhitelistContext(s: context) ? Nothing

Remove context from whitelist. For runtime operation see

org.fedoraproject.FirewallD1.policies.Methods.removeLockdownWhitelistContext.

Possible errors: NOT_ENABLED

removeLockdownWhitelistUid(i: uid) ? Nothing

Remove user id uid from whitelist. For runtime operation see

org.fedoraproject.FirewallD1.policies.Methods.removeLockdownWhitelistUid.

Possible errors: NOT_ENABLED

removeLockdownWhitelistUser(s: user) ? Nothing

Remove user from whitelist. For runtime operation see

org.fedoraproject.FirewallD1.policies.Methods.removeLockdownWhitelistUser.

Possible errors: NOT_ENABLED

setLockdownWhitelist((asasasai): settings) ? Nothing

Set permanent lockdown-whitelist configuration to settings.

Settings are in format: commands, selinux contexts, users, uids

commands (as): see command option in firewalld.lockdown-

whitelist(5).

selinux contexts (as): see selinux option in

firewalld.lockdown-whitelist(5).

users (as): see name attribute of user option in

firewalld.lockdown-whitelist(5).

uids (ai): see id attribute of user option in

firewalld.lockdown-whitelist(5).

Possible errors: INVALID_TYPE

Signals

LockdownWhitelistUpdated()

Emitted when permanent lockdown-whitelist configuration has

been updated.

org.fedoraproject.FirewallD1.config.ipset

Interface for permanent ipset configuration, see also

firewalld.ipset(5).

Methods

addEntry(s: entry) ? Nothing

Permanently add entry to list of entries of ipset. See entry

tag in firewalld.ipset(5). For runtime operation see

org.fedoraproject.FirewallD1.ipset.Methods.addEntry.

Possible errors: ALREADY_ENABLED

addOption(s: key, s: value) ? Nothing

Permanently add (key, value) to the ipset. See option tag in

firewalld.ipset(5).

Possible errors: ALREADY_ENABLED

getDescription() ? s

Get description of ipset. See description tag in

firewalld.ipset(5).

getEntries() ? as

Get list of entries added to ipset. See entry tag in

firewalld.ipset(5). For runtime operation see

org.fedoraproject.FirewallD1.ipset.Methods.getEntries.

Possible errors: IPSET_WITH_TIMEOUT

getOptions() ? a{ss}

Get dictionary of options set for ipset. See option tag in

firewalld.ipset(5).

getSettings() ? (ssssa{ss}as)

Return permanent settings of the ipset. For getting runtime settings see org.fedoraproject.FirewallD1.ipset.Methods.getIPSetSettings. Settings are in format: version, name, description, type, dictionary of options and array of entries. version (s): see version attribute of ipset tag in firewalld.ipset(5). name (s): see short tag in firewalld.ipset(5). description (s): see description tag in firewalld.ipset(5). type (s): see type attribute of ipset tag in firewalld.ipset(5). options (a{ss}): dictionary of {option : value} . See options tag in firewalld.ipset(5). entries (as): array of entries, see entry tag in firewalld.ipset(5).

getShort() ? s

Get name of ipset. See short tag in firewalld.ipset(5).

getType() ? s

Get type of ipset. See type attribute of ipset tag in firewalld.ipset(5).

getVersion() ? s

Get version of ipset. See version attribute of ipset tag in firewalld.ipset(5).

loadDefaults() ? Nothing

Load default settings for built-in ipset. Possible errors: NO_DEFAULTS

queryEntry(s: entry) ? b

Return whether entry has been added to ipset. For runtime operation see org.fedoraproject.FirewallD1.ipset.Methods.queryEntry.

queryOption(s: key, s: value) ? b

Return whether (key, value) has been added to options of the ipset.

remove() ? Nothing

Remove not built-in ipset.

Possible errors: BUILTIN_IPSET

removeEntry(s: entry) ? Nothing

Permanently remove entry from ipset. See entry tag in firewalld.ipset(5). For runtime operation see org.fedoraproject.FirewallD1.ipset.Methods.removeEntry.

Possible errors: NOT_ENABLED

removeOption(s: key) ? Nothing

Permanently remove key from the ipset. See option tag in firewalld.ipset(5).

Possible errors: NOT_ENABLED

rename(s: name) ? Nothing

Rename not built-in ipset to name.

Possible errors: BUILTIN_IPSET

setDescription(s: description) ? Nothing

Permanently set description of ipset to description. See description tag in firewalld.ipset(5).

setEntries(as: entries) ? Nothing

Permanently set list of entries to entries. See entry tag in firewalld.ipset(5).

setOptions(a{ss}: options) ? Nothing

Permanently set dict of options to options. See option tag in firewalld.ipset(5).

setShort(s: short) ? Nothing

Permanently set name of ipset to short. See short tag in firewalld.ipset(5).

setType(s: ipset_type) ? Nothing

Permanently set type of ipset to ipset_type. See type attribute of ipset tag in firewalld.ipset(5).

setVersion(s: version) ? Nothing

Permanently set version of ipset to version. See version attribute of ipset tag in firewalld.ipset(5).

update((ssssa{ss}as): settings) ? Nothing

Update settings of ipset to settings. Settings are in format:

version, name, description, type, dictionary of options and

array of entries.

version (s): see version attribute of ipset tag in

firewalld.ipset(5).

name (s): see short tag in firewalld.ipset(5).

description (s): see description tag in firewalld.ipset(5).

type (s): see type attribute of ipset tag in

firewalld.ipset(5).

options (a{ss}): dictionary of {option : value} . See options

tag in firewalld.ipset(5).

entries (as): array of entries, see entry tag in

firewalld.ipset(5).

Possible errors: INVALID_TYPE

Signals

Removed(s: name)

Emitted when ipset with name has been removed.

Renamed(s: name)

Emitted when ipset has been renamed to name.

Updated(s: name)

Emitted when ipset with name has been updated.

Properties

builtin - b - (ro)

True if ipset is build-in, false else.

default - b - (ro)

True if build-in ipset has default settings. False if it has

been modified. Always False for not build-in ipsets.

filename - s - (ro)

Name (including .xml extension) of file where the configuration

is stored.

name - s - (ro)

Name of ipset.

path - s - (ro)

Path to directory where the ipset configuration is stored.

Should be either /usr/lib/firewalld/ipsets or

/etc/firewalld/ipsets.

org.fedoraproject.FirewallD1.config.zone

Interface for permanent zone configuration, see also firewalld.zone(5).

Methods

addForwardPort(s: port, s: protocol, s: toport, s: toaddr) ?

Nothing

Permanently add (port, protocol, toport, toaddr) to list of

forward ports of zone. See forward-port tag in

firewalld.zone(5). For runtime operation see

org.fedoraproject.FirewallD1.zone.Methods.addForwardPort.

Possible errors: ALREADY_ENABLED

addIcmpBlock(s: icmptype) ? Nothing

Permanently add icmptype to list of icmp types blocked in zone.

See icmp-block tag in firewalld.zone(5). For runtime operation

see org.fedoraproject.FirewallD1.zone.Methods.addIcmpBlock.

Possible errors: ALREADY_ENABLED

addIcmpBlock(s: icmptype) ? Nothing

Permanently add icmp block inversion to zone. See

icmp-block-inversion tag in firewalld.zone(5). For runtime

operation see

org.fedoraproject.FirewallD1.zone.Methods.addIcmpBlockInversion.

Possible errors: ALREADY_ENABLED

addInterface(s: interface) ? Nothing

Permanently add interface to list of interfaces bound to zone.

See interface tag in firewalld.zone(5). For runtime operation

see org.fedoraproject.FirewallD1.zone.Methods.addInterface.

Possible errors: ALREADY_ENABLED

addMasquerade() ? Nothing

Permanently enable masquerading in zone. See masquerade tag in

firewalld.zone(5). For runtime operation see

org.fedoraproject.FirewallD1.zone.Methods.addMasquerade.

Possible errors: ALREADY_ENABLED

addPort(s: port, s: protocol) ? Nothing

Permanently add (port, protocol) to list of ports of zone. See port tag in firewalld.zone(5). For runtime operation see org.fedoraproject.FirewallD1.zone.Methods.addPort.

Possible errors: ALREADY_ENABLED

addProtocol(s: protocol) ? Nothing

Permanently add protocol into zone. The protocol can be any protocol supported by the system. Please have a look at /etc/protocols for supported protocols. For runtime operation see org.fedoraproject.FirewallD1.zone.Methods.addProtocol.

Possible errors: INVALID_PROTOCOL, ALREADY_ENABLED

addRichRule(s: rule) ? Nothing

Permanently add rule to list of rich-language rules in zone. See rule tag in firewalld.zone(5). For runtime operation see org.fedoraproject.FirewallD1.zone.Methods.addRichRule.

Possible errors: ALREADY_ENABLED

addService(s: service) ? Nothing

Permanently add service to list of services used in zone. See service tag in firewalld.zone(5). For runtime operation see org.fedoraproject.FirewallD1.zone.Methods.addService.

Possible errors: ALREADY_ENABLED

addSource(s: source) ? Nothing

Permanently add source to list of source addresses bound to zone. See source tag in firewalld.zone(5). For runtime operation see org.fedoraproject.FirewallD1.zone.Methods.addSource.

Possible errors: ALREADY_ENABLED

addSourcePort(s: port, s: protocol) ? Nothing

Permanently add (port, protocol) to list of source ports of zone. See source-port tag in firewalld.zone(5). For runtime operation see

org.fedoraproject.FirewallD1.zone.Methods.addSourcePort.

Possible errors: ALREADY_ENABLED

getDescription() ? s

Get description of zone. See description tag in firewalld.zone(5).

getForwardPorts() ? a(ssss)

Get list of (port, protocol, toport, toaddr) defined in zone. See forward-port tag in firewalld.zone(5). For runtime operation see org.fedoraproject.FirewallD1.zone.Methods.getForwardPorts.

getIcmpBlockInversion() ? b

Get icmp block inversion flag of zone. See icmp-block-inversion tag in firewalld.zone(5).

getIcmpBlocks() ? as

Get list of icmp type names blocked in zone. See icmp-block tag in firewalld.zone(5). For runtime operation see org.fedoraproject.FirewallD1.zone.Methods.getIcmpBlocks.

getInterfaces() ? as

Get list of interfaces bound to zone. See interface tag in firewalld.zone(5). For runtime operation see org.fedoraproject.FirewallD1.zone.Methods.getInterfaces.

getMasquerade() ? b

Return whether masquerade is enabled in zone. This is the same as queryMasquerade() method. See masquerade tag in firewalld.zone(5).

getPorts() ? a(ss)

Get list of (port, protocol) defined in zone. See port tag in firewalld.zone(5). For runtime operation see org.fedoraproject.FirewallD1.zone.Methods.getPorts.

getProtocols() ? as

Return array of protocols (s) previously enabled in zone. For getting runtime settings see org.fedoraproject.FirewallD1.zone.Methods.getProtocols.

getRichRules() ? as

Get list of rich-language rules in zone. See rule tag in firewalld.zone(5). For runtime operation see org.fedoraproject.FirewallD1.zone.Methods.getRichRules.

getServices() ? as

Get list of service names used in zone. See service tag in firewalld.zone(5). For runtime operation see org.fedoraproject.FirewallD1.zone.Methods.getServices.

getSettings() ? (sssbsasa(ss)asba(ssss)asasasasa(ss)b)

This function is deprecated, use org.fedoraproject.FirewallD1.config.zone.Methods.getSettings2 instead.

getSettings2() ? a{sv}

Return permanent settings of given zone. For getting runtime settings see org.fedoraproject.FirewallD1.zone.Methods.getZoneSettings2. Settings are a dictionary indexed by keywords. For the type of each value see below. If the value is empty it may be omitted.

version (s): see version attribute of zone tag in firewalld.zone(5).

name (s): see short tag in firewalld.zone(5).

description (s): see description tag in firewalld.zone(5).

target (s): see target attribute of zone tag in firewalld.zone(5).

services (as): array of service names, see service tag in firewalld.zone(5).

ports (a(ss)): array of port and protocol pairs. See port tag in firewalld.zone(5).

icmp_blocks (as): array of icmp-blocks. See icmp-block tag in firewalld.zone(5).

masquerade (b): see masquerade tag in firewalld.zone(5).

forward_ports (a(ssss)): array of (port, protocol, to-port, to-addr). See forward-port tag in firewalld.zone(5).

interfaces (as): array of interfaces. See interface tag in firewalld.zone(5).

sources (as): array of source addresses. See source tag in firewalld.zone(5).

rules_str (as): array of rich-language rules. See rule tag in firewalld.zone(5).

protocols (as): array of protocols, see protocol tag in firewalld.zone(5).

source_ports (a(ss)): array of port and protocol pairs. See source-port tag in firewalld.zone(5).

icmp_block_inversion (b): see icmp-block-inversion tag in firewalld.zone(5).

forward (b): see forward tag in firewalld.zone(5).

getShort() ? s

Get name of zone. See short tag in firewalld.zone(5).

getSourcePorts() ? a(ss)

Get list of (port, protocol) defined in zone. See source-port tag in firewalld.zone(5). For runtime operation see org.fedoraproject.FirewallD1.zone.Methods.getSourcePorts.

getSources() ? as

Get list of source addresses bound to zone. See source tag in firewalld.zone(5). For runtime operation see org.fedoraproject.FirewallD1.zone.Methods.getSources.

getTarget() ? s

Get target of zone. See target attribute of zone tag in firewalld.zone(5).

getVersion() ? s

Get version of zone. See version attribute of zone tag in firewalld.zone(5).

loadDefaults() ? Nothing

Load default settings for built-in zone.

Possible errors: NO_DEFAULTS

queryForwardPort(s: port, s: protocol, s: toport, s: toaddr) ? b

Return whether (port, protocol, toport, toaddr) is in list of forward ports of zone. See forward-port tag in firewalld.zone(5). For runtime operation see org.fedoraproject.FirewallD1.zone.Methods.queryForwardPort.

queryIcmpBlock(s: icmptype) ? b

Return whether icmptype is in list of icmp types blocked in zone. See icmp-block tag in firewalld.zone(5). For runtime operation see org.fedoraproject.FirewallD1.zone.Methods.queryIcmpBlock.

queryIcmpBlockInversion() ? b

Return whether icmp block inversion is in enabled in zone. See icmp-block-inversion tag in firewalld.zone(5). For runtime operation see org.fedoraproject.FirewallD1.zone.Methods.queryIcmpBlockInversion.

queryInterface(s: interface) ? b

Return whether interface is in list of interfaces bound to zone. See interface tag in firewalld.zone(5). For runtime operation see org.fedoraproject.FirewallD1.zone.Methods.queryInterface.

queryMasquerade() ? b

Return whether masquerade is enabled in zone. This is the same as getMasquerade() method. See masquerade tag in firewalld.zone(5). For runtime operation see org.fedoraproject.FirewallD1.zone.Methods.queryMasquerade.

queryPort(s: port, s: protocol) ? b

Return whether (port, protocol) is in list of ports of zone. See port tag in firewalld.zone(5). For runtime operation see org.fedoraproject.FirewallD1.zone.Methods.queryPort.

queryProtocol(s: protocol) ? b

Return whether protocol has been added in zone. For runtime operation see org.fedoraproject.FirewallD1.zone.Methods.queryProtocol. Possible errors: INVALID_PROTOCOL

queryRichRule(s: rule) ? b

Return whether rule is in list of rich-language rules in zone.

See rule tag in firewalld.zone(5). For runtime operation see

org.fedoraproject.FirewallD1.zone.Methods.queryRichRule.

queryService(s: service) ? b

Return whether service is in list of services used in zone. See

service tag in firewalld.zone(5). For runtime operation see

org.fedoraproject.FirewallD1.zone.Methods.queryService.

querySource(s: source) ? b

Return whether source is in list of source addresses bound to

zone. See source tag in firewalld.zone(5). For runtime

operation see

org.fedoraproject.FirewallD1.zone.Methods.querySource.

querySourcePort(s: port, s: protocol) ? b

Return whether (port, protocol) is in list of source ports of

zone. See source-port tag in firewalld.zone(5). For runtime

operation see

org.fedoraproject.FirewallD1.zone.Methods.querySourcePort.

remove() ? Nothing

Remove not built-in zone.

Possible errors: BUILTIN_ZONE

removeForwardPort(s: port, s: protocol, s: toport, s: toaddr) ?

Nothing

Permanently remove (port, protocol, toport, toaddr) from list

of forward ports of zone. See forward-port tag in

firewalld.zone(5). For runtime operation see

org.fedoraproject.FirewallD1.zone.Methods.removeForwardPort.

Possible errors: NOT_ENABLED

removeIcmpBlock(s: icmptype) ? Nothing

Permanently remove icmptype from list of icmp types blocked in

zone. See icmp-block tag in firewalld.zone(5). For runtime

operation see

org.fedoraproject.FirewallD1.zone.Methods.removeIcmpBlock.

Possible errors: NOT_ENABLED

removeIcmpBlockInversion() ? Nothing

Permanently remove icmp block inversion from the zone. See icmp-block-inversion tag in firewalld.zone(5). For runtime operation see org.fedoraproject.FirewallD1.zone.Methods.removeIcmpBlockInversion.

Possible errors: NOT_ENABLED

removeInterface(s: interface) ? Nothing

Permanently remove interface from list of interfaces bound to zone. See interface tag in firewalld.zone(5). For runtime operation see org.fedoraproject.FirewallD1.zone.Methods.removeInterface.

Possible errors: NOT_ENABLED

removeMasquerade() ? Nothing

Permanently disable masquerading in zone. See masquerade tag in firewalld.zone(5). For runtime operation see org.fedoraproject.FirewallD1.zone.Methods.removeMasquerade.

Possible errors: NOT_ENABLED

removePort(s: port, s: protocol) ? Nothing

Permanently remove (port, protocol) from list of ports of zone. See port tag in firewalld.zone(5). For runtime operation see org.fedoraproject.FirewallD1.zone.Methods.removePort.

Possible errors: NOT_ENABLED

removeProtocol(s: protocol) ? Nothing

Permanently remove protocol from zone. For runtime operation see org.fedoraproject.FirewallD1.zone.Methods.removeProtocol.

Possible errors: INVALID_PROTOCOL, NOT_ENABLED

removeRichRule(s: rule) ? Nothing

Permanently remove rule from list of rich-language rules in zone. See rule tag in firewalld.zone(5). For runtime operation see org.fedoraproject.FirewallD1.zone.Methods.removeRichRule.

Possible errors: NOT_ENABLED

removeService(s: service) ? Nothing

Permanently remove service from list of services used in zone. See service tag in firewalld.zone(5). For runtime operation see org.fedoraproject.FirewallD1.zone.Methods.removeService. Possible errors: NOT_ENABLED

removeSource(s: source) ? Nothing

Permanently remove source from list of source addresses bound to zone. See source tag in firewalld.zone(5). For runtime operation see org.fedoraproject.FirewallD1.zone.Methods.removeSource. Possible errors: NOT_ENABLED

removeSourcePort(s: port, s: protocol) ? Nothing

Permanently remove (port, protocol) from list of source ports of zone. See source-port tag in firewalld.zone(5). For runtime operation see org.fedoraproject.FirewallD1.zone.Methods.removeSourcePort. Possible errors: NOT_ENABLED

rename(s: name) ? Nothing

Rename not built-in zone to name. Possible errors: BUILTIN_ZONE

setDescription(s: description) ? Nothing

Permanently set description of zone to description. See description tag in firewalld.zone(5).

setForwardPorts(a(ssss): ports) ? Nothing

Permanently set forward ports of zone to list of (port, protocol, toport, toaddr). See forward-port tag in firewalld.zone(5).

setIcmpBlockInversion(b: flag) ? Nothing

Permanently set icmp block inversion flag of zone to flag. See icmp-block-inversion tag in firewalld.zone(5).

setIcmpBlocks(as: icmptypes) ? Nothing

Permanently set list of icmp types blocked in zone to icmptypes. See icmp-block tag in firewalld.zone(5).

setInterfaces(as: interfaces) ? Nothing

Permanently set list of interfaces bound to zone to interfaces.

See interface tag in firewalld.zone(5).

setMasquerade(b: masquerade) ? Nothing

Permanently set masquerading in zone to masquerade. See masquerade tag in firewalld.zone(5).

setPorts(a(ss): ports) ? Nothing

Permanently set ports of zone to list of (port, protocol). See port tag in firewalld.zone(5).

setProtocols(as: protocols) ? Nothing

Permanently set list of protocols used in zone to protocols.

See protocol tag in firewalld.zone(5).

setRichRules(as: rules) ? Nothing

Permanently set list of rich-language rules to rules. See rule tag in firewalld.zone(5).

setServices(as: services) ? Nothing

Permanently set list of services used in zone to services. See service tag in firewalld.zone(5).

setShort(s: short) ? Nothing

Permanently set name of zone to short. See short tag in firewalld.zone(5).

setSourcePorts(a(ss): ports) ? Nothing

Permanently set source-ports of zone to list of (port, protocol). See source-port tag in firewalld.zone(5).

setSources(as: sources) ? Nothing

Permanently set list of source addresses bound to zone to sources. See source tag in firewalld.zone(5).

setTarget(s: target) ? Nothing

Permanently set target of zone to target. See target attribute of zone tag in firewalld.zone(5).

setVersion(s: version) ? Nothing

Permanently set version of zone to version. See version attribute of zone tag in firewalld.zone(5).

update((sssbsasa(ss)asba(ssss)asasasasa(ss)b): settings) ? Nothing

This function is deprecated, use

org.fedoraproject.FirewallD1.config.zone.Methods.update2

instead.

update2(a{sv}: settings) ? Nothing

Update settings of zone to settings. Settings are a dictionary

indexed by keywords. For the type of each value see below. To

zero a value pass an empty string or list.

version (s): see version attribute of zone tag in

firewalld.zone(5).

name (s): see short tag in firewalld.zone(5).

description (s): see description tag in firewalld.zone(5).

target (s): see target attribute of zone tag in

firewalld.zone(5).

services (as): array of service names, see service tag in

firewalld.zone(5).

ports (a(ss)): array of port and protocol pairs. See port tag

in firewalld.zone(5).

icmp_blocks (as): array of icmp-blocks. See icmp-block tag in

firewalld.zone(5).

masquerade (b): see masquerade tag in firewalld.zone(5).

forward_ports (a(ssss)): array of (port, protocol, to-port,

to-addr). See forward-port tag in firewalld.zone(5).

interfaces (as): array of interfaces. See interface tag in

firewalld.zone(5).

sources (as): array of source addresses. See source tag in

firewalld.zone(5).

rules_str (as): array of rich-language rules. See rule tag in

firewalld.zone(5).

protocols (as): array of protocols, see protocol tag in

firewalld.zone(5).

source_ports (a(ss)): array of port and protocol pairs. See

source-port tag in firewalld.zone(5).

icmp_block_inversion (b): see icmp-block-inversion tag in

firewalld.zone(5).

forward (b): see forward tag in firewalld.zone(5).

Possible errors: INVALID_TYPE

Signals

Removed(s: name)

Emitted when zone with name has been removed.

Renamed(s: name)

Emitted when zone has been renamed to name.

Updated(s: name)

Emitted when zone with name has been updated.

Properties

builtin - b - (ro)

True if zone is build-in, false else.

default - b - (ro)

True if build-in zone has default settings. False if it has

been modified. Always False for not build-in zones.

filename - s - (ro)

Name (including .xml extension) of file where the configuration

is stored.

name - s - (ro)

Name of zone.

path - s - (ro)

Path to directory where the zone configuration is stored.

Should be either /usr/lib/firewalld/zones or

/etc/firewalld/zones.

org.fedoraproject.FirewallD1.config.policy

Interface for permanent policy configuration, see also

firewalld.policy(5).

Methods

getSettings() ? a{sv}

Return permanent settings of given policy. For getting runtime

settings see

org.fedoraproject.FirewallD1.policy.Methods.getPolicySettings.

Settings are a dictionary indexed by keywords. For possible keywords see org.fedoraproject.FirewallD1.config.Methods.addPolicy.

loadDefaults() ? Nothing

Load default settings for built-in policy.

Possible errors: NO_DEFAULTS

remove() ? Nothing

Remove not built-in policy.

Possible errors: BUILTIN_POLICY

rename(s: name) ? Nothing

Rename not built-in policy to name.

Possible errors: BUILTIN_POLICY

update(a{sv}: settings) ? Nothing

Update settings of policy to settings. Settings are a dictionary indexed by keywords. For possible keywords see org.fedoraproject.FirewallD1.config.Methods.addPolicy. To zero a value pass an empty string or list.

Possible errors: INVALID_TYPE

Signals

Removed(s: name)

Emitted when policy with name has been removed.

Renamed(s: name)

Emitted when policy has been renamed to name.

Updated(s: name)

Emitted when policy with name has been updated.

Properties

builtin - b - (ro)

True if policy is build-in, false else.

default - b - (ro)

True if build-in policy has default settings. False if it has been modified. Always False for not build-in policies.

filename - s - (ro)

Name (including .xml extension) of file where the configuration

is stored.

name - s - (ro)

Name of policy.

path - s - (ro)

Path to directory where the policy configuration is stored.

Should be either /usr/lib/firewalld/policies or

/etc/firewalld/policies.

org.fedoraproject.FirewallD1.config.service

Interface for permanent service configuration, see also

firewalld.service(5).

Methods

addModule(s: module) ? Nothing

This method is deprecated. Please use "helpers" in the

update2() method.

addPort(s: port, s: protocol) ? Nothing

Permanently add (port, protocol) to list of ports in service.

See port tag in firewalld.service(5).

Possible errors: ALREADY_ENABLED

addProtocol(s: protocol) ? Nothing

Permanently add protocol into zone. The protocol can be any

protocol supported by the system. Please have a look at

/etc/protocols for supported protocols. See protocol tag in

firewalld.service(5).

Possible errors: INVALID_PROTOCOL, ALREADY_ENABLED

addSourcePort(s: port, s: protocol) ? Nothing

Permanently add (port, protocol) to list of source ports in

service. See source-port tag in firewalld.service(5).

Possible errors: ALREADY_ENABLED

getDescription() ? s

Get description of service. See description tag in

firewalld.service(5).

getDestination(s: family) ? s

Get destination for IP family being either 'ipv4' or 'ipv6'.

See destination tag in firewalld.service(5).

Possible errors: ALREADY_ENABLED

getDestinations() ? a{ss}

Get list of destinations. Return value is a dictionary of {IP family : IP address} where 'IP family' key can be either 'ipv4' or 'ipv6'. See destination tag in firewalld.service(5).

getModules() ? as

This method is deprecated. Please use "helpers" in the getSettings2() method.

getPorts() ? a(ss)

Get list of (port, protocol) defined in service. See port tag in firewalld.service(5).

getProtocols() ? as

Return array of protocols (s) defined in service. See protocol tag in firewalld.service(5).

getSettings() ? (sssa(ss)asa{ss}asa(ss))

This function is deprecated, use org.fedoraproject.FirewallD1.config.service.Methods.getSettings2 instead.

getSettings2(s: service) ? s{sv}

Return runtime settings of given service. For getting runtime settings see org.fedoraproject.FirewallD1.Methods.getServiceSettings2. Settings are a dictionary indexed by keywords. For the type of each value see below. If the value is empty it may be omitted.

version (s): see version attribute of service tag in firewalld.service(5).

name (s): see short tag in firewalld.service(5).

description (s): see description tag in firewalld.service(5).

ports (a(ss)): array of port and protocol pairs. See port tag in firewalld.service(5).

module names (as): array of kernel netfilter helpers, see module tag in firewalld.service(5).

destinations (a{ss}): dictionary of {IP family : IP address}
where 'IP family' key can be either 'ipv4' or 'ipv6'. See
destination tag in firewalld.service(5).
protocols (as): array of protocols, see protocol tag in
firewalld.service(5).
source_ports (a(ss)): array of port and protocol pairs. See
source-port tag in firewalld.service(5).
includes (as): array of service includes, see include tag in
firewalld.service(5).
helpers (as): array of service helpers, see helper tag in
firewalld.service(5).

getShort() ? s

Get name of service. See short tag in firewalld.service(5).

getSourcePorts() ? a(ss)

Get list of (port, protocol) defined in service. See
source-port tag in firewalld.service(5).

getVersion() ? s

Get version of service. See version attribute of service tag in
firewalld.service(5).

loadDefaults() ? Nothing

Load default settings for built-in service.
Possible errors: NO_DEFAULTS

queryDestination(s: family, s: address) ? b

Return whether a destination is in dictionary of destinations
of this service. destination is in format: (IP family, IP
address) where IP family can be either 'ipv4' or 'ipv6'. See
destination tag in firewalld.service(5).

queryModule(s: module) ? b

This method is deprecated. Please use "helpers" in the
getSettings2() method.

queryPort(s: port, s: protocol) ? b

Return whether (port, protocol) is in list of ports in service.
See port tag in firewalld.service(5).

queryProtocol(s: protocol) ? b

Return whether protocol is in list of protocols in service. See protocol tag in firewalld.service(5).

querySourcePort(s: port, s: protocol) ? b

Return whether (port, protocol) is in list of source ports in service. See source-port tag in firewalld.service(5).

remove() ? Nothing

Remove not built-in service.

Possible errors: BUILTIN_SERVICE

removeDestination(s: family) ? Nothing

Permanently remove a destination with family ('ipv4' or 'ipv6') from service. See destination tag in firewalld.service(5).

Possible errors: NOT_ENABLED

removeModule(s: module) ? Nothing

This method is deprecated. Please use "helpers" in the update2() method.

removePort(s: port, s: protocol) ? Nothing

Permanently remove (port, protocol) from list of ports in service. See port tag in firewalld.service(5).

Possible errors: NOT_ENABLED

removeProtocol(s: protocol) ? Nothing

Permanently remove protocol from list of protocols in service. See protocol tag in firewalld.service(5).

Possible errors: NOT_ENABLED

removeSourcePort(s: port, s: protocol) ? Nothing

Permanently remove (port, protocol) from list of source ports in service. See source-port tag in firewalld.service(5).

Possible errors: NOT_ENABLED

rename(s: name) ? Nothing

Rename not built-in service to name.

Possible errors: BUILTIN_SERVICE

setDescription(s: description) ? Nothing

Permanently set description of service to description. See

description tag in firewalld.service(5).

setDestination(s: family, s: address) ? Nothing

Permanently set a destination address. destination is in format: (IP family, IP address) where IP family can be either 'ipv4' or 'ipv6'. See destination tag in firewalld.service(5). Possible errors: ALREADY_ENABLED

setDestinations(a{ss}: destinations) ? Nothing

Permanently set destinations of service to destinations, which is a dictionary of {IP family : IP address} where 'IP family' key can be either 'ipv4' or 'ipv6'. See destination tag in firewalld.service(5).

setModules(as: modules) ? Nothing

This method is deprecated. Please use "helpers" in the update2() method.

setPorts(a(ss): ports) ? Nothing

Permanently set ports of service to list of (port, protocol). See port tag in firewalld.service(5).

setProtocols(as: protocols) ? Nothing

Permanently set protocols of service to list of protocols. See protocol tag in firewalld.service(5).

setShort(s: short) ? Nothing

Permanently set name of service to short. See short tag in firewalld.service(5).

setSourcePorts(a(ss): ports) ? Nothing

Permanently set source-ports of service to list of (port, protocol). See source-port tag in firewalld.service(5).

setVersion(s: version) ? Nothing

Permanently set version of service to version. See version attribute of service tag in firewalld.service(5).

update((sssa(ss)asa{ss}asa(ss)): settings) ? Nothing

This function is deprecated, use org.fedoraproject.FirewallD1.config.service.Methods.update2 instead.

update2a{sv}: settings) ? Nothing

Update settings of service to settings. Settings are a dictionary indexed by keywords. For the type of each value see below. To zero a value pass an empty string or list.

version (s): see version attribute of service tag in firewalld.service(5).

name (s): see short tag in firewalld.service(5).

description (s): see description tag in firewalld.service(5).

ports (a(ss)): array of port and protocol pairs. See port tag in firewalld.service(5).

module names (as): array of kernel netfilter helpers, see module tag in firewalld.service(5).

destinations (a{ss}): dictionary of {IP family : IP address} where 'IP family' key can be either 'ipv4' or 'ipv6'. See destination tag in firewalld.service(5).

protocols (as): array of protocols, see protocol tag in firewalld.service(5).

source_ports (a(ss)): array of port and protocol pairs. See source-port tag in firewalld.service(5).

includes (as): array of service includes, see include tag in firewalld.service(5).

helpers (as): array of service helpers, see helper tag in firewalld.service(5).

Possible errors: INVALID_TYPE

Signals

Removed(s: name)

Emitted when service with name has been removed.

Renamed(s: name)

Emitted when service has been renamed to name.

Updated(s: name)

Emitted when service with name has been updated.

Properties

builtin - b - (ro)

True if service is build-in, false else.

default - b - (ro)

True if build-in service has default settings. False if it has been modified. Always False for not build-in services.

filename - s - (ro)

Name (including .xml extension) of file where the configuration is stored.

name - s - (ro)

Name of service.

path - s - (ro)

Path to directory where the configuration is stored. Should be either /usr/lib/firewalld/services or /etc/firewalld/services.

org.fedoraproject.FirewallD1.config.helper

Interface for permanent helper configuration, see also firewalld.helper(5).

Methods

addPort(s: port, s: protocol) ? Nothing

Permanently add (port, protocol) to list of ports in helper. See port tag in firewalld.helper(5).

Possible errors: ALREADY_ENABLED

getDescription() ? s

Get description of helper. See description tag in firewalld.helper(5).

getFamily() ? s

Get family being 'ipv4', 'ipv6' or empty for both. See family tag in firewalld.helper(5).

getModule() ? s

Get modules (netfilter kernel helpers) used in helper. See module tag in firewalld.helper(5).

getPorts() ? a(ss)

Get list of (port, protocol) defined in helper. See port tag in firewalld.helper(5).

getSettings() ? (sssssa(ss))

Return permanent settings of a helper. For getting runtime settings see org.fedoraproject.FirewallD1.Methods.getHelperSettings. Settings are in format: version, name, description, family, module, array of ports (port, protocol).

version (s): see version attribute of helper tag in firewalld.helper(5).

name (s): see short tag in firewalld.helper(5).

description (s): see description tag in firewalld.helper(5).

family (s): see family tag in firewalld.helper(5).

module (s): see module tag in firewalld.helper(5).

ports (a(ss)): array of port and protocol pairs. See port tag in firewalld.helper(5).

getShort() ? s

Get name of helper. See short tag in firewalld.helper(5).

getVersion() ? s

Get version of helper. See version attribute of helper tag in firewalld.helper(5).

loadDefaults() ? Nothing

Load default settings for built-in helper.

Possible errors: NO_DEFAULTS

queryFamily(s: module) ? b

Return whether family is set for helper. See family tag in firewalld.helper(5).

queryModule(s: module) ? b

Return whether module (netfilter kernel helpers) is used in helper. See module tag in firewalld.helper(5).

queryPort(s: port, s: protocol) ? b

Return whether (port, protocol) is in list of ports in helper. See port tag in firewalld.helper(5).

remove() ? Nothing

Remove not built-in helper.

Possible errors: BUILTIN_HELPER

removePort(s: port, s: protocol) ? Nothing

Permanently remove (port, protocol) from list of ports in helper. See port tag in firewalld.helper(5).

Possible errors: NOT_ENABLED

rename(s: name) ? Nothing

Rename not built-in helper to name.

Possible errors: BUILTIN_HELPER

setDescription(s: description) ? Nothing

Permanently set description of helper to description. See description tag in firewalld.helper(5).

setFamily(s: family) ? Nothing

Permanently set family of helper to family. See family tag in firewalld.helper(5).

setModule(s: module) ? Nothing

Permanently set module of helper to description. See module tag in firewalld.helper(5).

setPorts(a(ss): ports) ? Nothing

Permanently set ports of helper to list of (port, protocol). See port tag in firewalld.helper(5).

setShort(s: short) ? Nothing

Permanently set name of helper to short. See short tag in firewalld.helper(5).

setVersion(s: version) ? Nothing

Permanently set version of helper to version. See version attribute of helper tag in firewalld.helper(5).

update((sssssa(ss)): settings) ? Nothing

Update settings of helper to settings. Settings are in format: version, name, description, family, module and array of ports.

version (s): see version attribute of helper tag in firewalld.helper(5).

name (s): see short tag in firewalld.helper(5).

description (s): see description tag in firewalld.helper(5).

family (s): see family tag in firewalld.helper(5).

module (s): see module tag in firewalld.helper(5).

ports (a(ss)): array of port and protocol pairs. See port tag in firewalld.helper(5).

Possible errors: INVALID_HELPER

Signals

Removed(s: name)

Emitted when helper with name has been removed.

Renamed(s: name)

Emitted when helper has been renamed to name.

Updated(s: name)

Emitted when helper with name has been updated.

Properties

builtin - b - (ro)

True if helper is build-in, false else.

default - b - (ro)

True if build-in helper has default settings. False if it has been modified. Always False for not build-in helpers.

filename - s - (ro)

Name (including .xml extension) of file where the configuration is stored.

name - s - (ro)

Name of helper.

path - s - (ro)

Path to directory where the configuration is stored. Should be either /usr/lib/firewalld/helpers or /etc/firewalld/helpers.

org.fedoraproject.FirewallD1.config.icmptype

Interface for permanent icmp type configuration, see also firewalld.icmptype(5).

Methods

addDestination(s: destination) ? Nothing

Permanently add a destination ('ipv4' or 'ipv6') to list of destinations of this icmp type. See destination tag in firewalld.icmptype(5).

Possible errors: ALREADY_ENABLED

getDescription() ? s

Get description of icmp type. See description tag in firewalld.icmptype(5).

getDestinations() ? as

Get list of destinations. See destination tag in firewalld.icmptype(5).

getSettings() ? (sssas)

Return permanent settings of icmp type. For getting runtime settings see org.fedoraproject.FirewallD1.Methods.getIcmpTypeSettings. Settings are in format: version, name, description, array of destinations.

version (s): see version attribute of icmptype tag in firewalld.icmptype(5).

name (s): see short tag in firewalld.icmptype(5).

description (s): see description tag in firewalld.icmptype(5).

destinations (as): array, either empty or containing strings 'ipv4' and/or 'ipv6', see destination tag in firewalld.icmptype(5).

getShort() ? s

Get name of icmp type. See short tag in firewalld.icmptype(5).

getVersion() ? s

Get version of icmp type. See version attribute of icmptype tag in firewalld.icmptype(5).

loadDefaults() ? Nothing

Load default settings for built-in icmp type.

Possible errors: NO_DEFAULTS

queryDestination(s: destination) ? b

Return whether a destination ('ipv4' or 'ipv6') is in list of destinations of this icmp type. See destination tag in firewalld.icmptype(5).

remove() ? Nothing

Remove not built-in icmp type.

Possible errors: BUILTIN_ICMPTYPE

removeDestination(s: destination) ? Nothing

Permanently remove a destination ('ipv4' or 'ipv6') from list of destinations of this icmp type. See destination tag in firewalld.icmptype(5).

Possible errors: NOT_ENABLED

rename(s: name) ? Nothing

Rename not built-in icmp type to name.

Possible errors: BUILTIN_ICMPTYPE

setDescription(s: description) ? Nothing

Permanently set description of icmp type to description. See description tag in firewalld.icmptype(5).

setDestinations(as: destinations) ? Nothing

Permanently set destinations of icmp type to destinations, which is array, either empty or containing strings 'ipv4' and/or 'ipv6'. See destination tag in firewalld.icmptype(5).

setShort(s: short) ? Nothing

Permanently set name of icmp type to short. See short tag in firewalld.icmptype(5).

setVersion(s: version) ? Nothing

Permanently set version of icmp type to version. See version attribute of icmptype tag in firewalld.icmptype(5).

update((sssas): settings) ? Nothing

Update permanent settings of icmp type to settings. Settings are in format: version, name, description, array of destinations.

version (s): see version attribute of icmptype tag in firewalld.icmptype(5).

name (s): see short tag in firewalld.icmptype(5).

description (s): see description tag in firewalld.icmptype(5).

destinations (as): array, either empty or containing strings 'ipv4' and/or 'ipv6', see destination tag in

firewalld.icmptype(5).

Signals

Removed(s: name)

Emitted when icmp type with name has been removed.

Renamed(s: name)

Emitted when icmp type has been renamed to name.

Updated(s: name)

Emitted when icmp type with name has been updated.

Properties

builtin - b - (ro)

True if icmptype is build-in, false else.

default - b - (ro)

True if build-in icmp type has default settings. False if it has been modified. Always False for not build-in zones.

filename - s - (ro)

Name (including .xml extension) of file where the configuration is stored.

name - s - (ro)

Name of icmp type.

path - s - (ro)

Path to directory where the icmp type configuration is stored. Should be either /usr/lib/firewalld/icmptypes or /etc/firewalld/icmptypes.

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

AUTHORS

Thomas Woerner <twoerner@redhat.com>

Developer

Jiri Popelka <jpopelka@redhat.com>

Developer

Eric Garver <eric@garver.life>

Developer