



Software 6 Release Notes

Release 6.9.6-RC1

Westermo Network Technologies AB

November 25, 2021

Contents

1	General Information	3
2	Release Highlights	4
2.1	RC0	4
2.2	RC1	4
3	Limitations	5
4	Configuration Parameter Changes	5
5	Changed Configuration Parameter Descriptions	8
5.1	MIB Reference: WESTERMO-SW6-MIB	8
5.2	MIB Reference: WESTERMO-SW6-FIREWALL-MIB	25



1 General Information

Company

Westermo Network Technologies AB

Contact Support

www.westermo.com

Release Number

6.9.6-RC1

Software Build Number

8d00a7cfa869d80b2c1e8626310e20477a83e322

Date of this build

November 25, 2021

2 Release Highlights

2.1 RC0

- Firewall: Add stateful rules
- Firewall: Add 1:1 NAT
- Firewall: Add logging per rule
- Firewall: Add ICMP filtering
- MultiWAN: Add policy routing with multiple routing tables
- MultiWAN: Add prioritised static and DHCP routing tables
- SNMP: Add advanced SNMPv3 security (AES256, SHA384/SHA512)
- SNMP: Add SNMP INFORM notification
- Cellular: Add Cellular module firmware update
- Cellular: Add Cellular signal and connection reporting
- WebInterface: Devices with a single radio now only show a single radio
- DHCP-Client: Support for MTU (Option 26)
- Status: Add hash of current configuration (see [swCfgHash](#))
- Support File: Add qlog cellular debug information (cellular devices)
- Support File: Add qca crashdump debug information (802.11ac devices)

2.2 RC1

- Regulatory: Update EIRP limits for country code SINGAPORE (802.11n products)
- Regulatory: Reduce margin to country code EIRP limits from 2 dB to 1dB (802.11n products)
- Security: Fix CVE-2020-3702 - Cryptographic Issues in WIFI driver (Krook)

3 Limitations

- When the device is reconfigured to Mesh with SAE as encryption, the device has to be rebooted after applying the configuration
- Multi-SSID with DFS channels is not working (802.11n products only)

4 Configuration Parameter Changes

The following configuration items have been added/changed/removed:

- [cfgNetVlanComment](#) (added)
- [cfgNetMacVlanComment](#) (added)
- [cfgRouteTableInterface](#) (added)
- [cfgRouteTableMetric](#) (added)
- [cfgRouteTableRoutingTables](#) (added)
- [cfgRouteTableMonitor](#) (added)
- [cfgRouteTableWeight](#) (added)
- [cfgRouteRuleTable](#) (added)
- [cfgRouteRuleEnabled](#) (added)
- [cfgRouteRulePreference](#) (added)
- [cfgRouteRuleFrom](#) (added)
- [cfgRouteRuleTo](#) (added)
- [cfgRouteRuleInputInterface](#) (added)
- [cfgRouteRuleTos](#) (added)
- [cfgRouteRuleLookupTable](#) (added)
- [cfgRouteDhcpTable](#) (added)

- [cfgRouteDhcpEnabled](#) (added)
- [cfgRouteDhcpInterface](#) (added)
- [cfgRouteDhcpMetric](#) (added)
- [cfgRouteDhcpRoutingTables](#) (added)
- [cfgRouteDhcpMonitor](#) (added)
- [cfgRouteDhcpWeight](#) (added)
- [cfgRouteGlobal](#) (added)
- [cfgRouteGlblReversePathFilter](#) (added)
- [cfgSnmpdAuthType](#) (added)
- [cfgSnmpdPrivType](#) (added)
- [cfgSnmpTrapType](#) (added)
- [cfgSnmpTrapEngineId](#) (added)
- [cfgSnmpTrapUser](#) (added)
- [cfgSnmpTrapAuthProtocol](#) (added)
- [cfgSnmpTrapAuthPassword](#) (added)
- [cfgSnmpTrapPrivProtocol](#) (added)
- [cfgSnmpTrapPrivPassword](#) (added)
- [cfgSnmpTrapTimeout](#) (added)
- [cfgSnmpTrapRetries](#) (added)
- [cfgCellDbgSignal](#) (added)
- [cfgCellDbgConnection](#) (added)
- [rpcCellular](#) (added)
- [rpcCellFwUpgrade](#) (added)
- [setCellular](#) (added)

- [setCellFwUrl](#) (added)
- [setCellSimSlot](#) (added)
- [swSysMsgEpoch](#) (added)
- [swSysMsgDate](#) (added)
- [swCfgHash](#) (added)
- [cfgFwNatOneToOneTable](#) (added)
- [cfgFwNatOneToOneEnabled](#) (added)
- [cfgFwNatOneToOneType](#) (added)
- [cfgFwNatOneToOneInterface](#) (added)
- [cfgFwNatOneToOneSourceNet](#) (added)
- [cfgFwNatOneToOneDestinationNet](#) (added)
- [cfgFwNatOneToOneRewriteNet](#) (added)
- [cfgFwNatOneToOneComment](#) (added)
- [cfgFwFltLogDefaultDrop](#) (added)
- [cfgFwFltLogDefaultAccept](#) (added)
- [cfgFwFltInputSynLimit](#) (added)
- [cfgFwFltInputSynLimitBurst](#) (added)
- [cfgFwFltRlcmpType](#) (added)
- [cfgFwFltRlcmpCode](#) (added)
- [cfgFwFltRLog](#) (added)
- [cfgFwFltRLimit](#) (added)
- [cfgFwFltRLimitBurst](#) (added)
- [cfgRouteTableGateway](#) (changed)
- [cfgNtpServer1](#) (changed)

- [cfgNtpServer2](#) (changed)
- [cfgIpTables](#) (removed)
- [cfgRouteDefGwOverride](#) (removed)

5 Changed Configuration Parameter Descriptions

5.1 MIB Reference: WESTERMO-SW6-MIB

5.1.1 [cfgCellDbgSignal](#)

Disable or Enable Signal Level Trap

Persistent default value to disable or enable the signal level trap 820.

Applies to cellular products only.

<i>Enumeration</i>	disabled (0), enabled (1)
<i>Access</i>	readwrite
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.101.4.1.2

5.1.2 [cfgCellDbgConnection](#)

Disable or Enable Connection State Trap

Persistent default value to disable or enable the connection state trap 821.

Applies to cellular products only.

<i>Enumeration</i>	disabled (0), enabled (1)
<i>Access</i>	readwrite
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.101.4.1.3

5.1.3 [cfgSnmpdAuthType](#)

Authentication Type for SNMPv3

This parameter is used when `cfgSnmpdVersion` is set to **v3usm(1)**.

<i>Enumeration</i>	sha1 (2), sha384 (5), sha512 (6)
<i>Access</i>	readwrite
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.12.1.7

5.1.4 cfgSnmpdPrivType

Encryption Type for SNMPv3

This parameter is used when `cfgSnmpdVersion` is set to **v3usm(1)**.

<i>Enumeration</i>	aes128 (2), aes256 (4)
<i>Access</i>	readwrite
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.12.1.8

5.1.5 cfgSnmpTrapAuthProtocol

SNMP Authentication Protocol

This parameter is active when `cfgSnmpTrapVersion` is set to **v3usm(2)**.

Set the authentication protocol used for authenticated SNMPv3 notifications.

<i>Enumeration</i>	none (0), sha1 (2), sha384 (5), sha512 (6)
<i>Access</i>	readwrite
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.12.10.10

5.1.6 cfgSnmpTrapAuthPassword

Authentication Password

This parameter is active when `cfgSnmpTrapVersion` is set to **v3usm(2)**.

Set the authentication password used for authenticated SNMPv3 notifications.

<i>Type</i>	DisplayString
<i>Range</i>	1 - 255
<i>Access</i>	readwrite
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.12.10.11

5.1.7 cfgSnmpTrapPrivProtocol

SNMP Privacy Protocol

This parameter is active when `cfgSnmpTrapVersion` is set to **v3usm(2)**.

Set the privacy protocol used for authenticated SNMPv3 notifications.

<i>Enumeration</i>	none (0), aes128 (2), aes256 (4)
<i>Access</i>	readwrite
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.12.10.12

5.1.8 cfgSnmpTrapPrivPassword

Privacy Password

This parameter is active when `cfgSnmpTrapVersion` is set to **v3usm(2)**.

Set the privacy password used for encrypted SNMPv3 notifications.

<i>Type</i>	DisplayString
<i>Range</i>	1 - 255
<i>Access</i>	readwrite
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.12.10.13

5.1.9 cfgSnmpTrapTimeout

SNMP INFORM Timeout

This parameter is active when `cfgSnmpTrapType` is set to **inform(2)**.

Specifies the timeout in seconds between retries.

<i>Range</i>	1 - 2147483647
<i>Access</i>	readwrite
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.12.10.14

5.1.10 cfgSnmpTrapRetries

SNMP INFORM Retries

This parameter is active when `cfgSnmptTrapType` is set to **inform(2)**.

Specifies the number of retries.

<i>Range</i>	0 - 2147483647
<i>Access</i>	readwrite
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.12.10.15

5.1.11 `cfgSnmptTrapType`

SNMP Notification Type

- **trap(1)**: Send all SNMP notifications as TRAP
- **inform(2)**: Send all SNMP notifications as INFORM

<i>Enumeration</i>	trap (1), inform (2)
<i>Access</i>	readwrite
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.12.10.5

5.1.12 `cfgSnmptTrapEngineId`

SNMP Authoritative Engine ID

This parameter is active when `cfgSnmptTrapVersion` is set to **v3usm(2)**.

This parameter may be set to 'auto' (recommended). In 'auto' mode the behaviour is as follows:

- If `cfgSnmptTrapType` is **trap(1)** then the engine ID of the SNMP Agent is used in the notification (see `cfgSnmptdVersion`).
- If `cfgSnmptTrapType` is **inform(1)** then the SNMPv3 discovery process is used to retrieve authoritative engine ID from the receiver.

Otherwise, this parameter may be used to define the engine ID used as the authoritative engine ID for SNMPv3 notifications. The engine ID must be given as a hexadecimal string (optionally prefixed by '0x'). The value must be between 5 and 32 octets long. The string has to follow the engine ID definition in RFC 3411.

In this mode the behaviour is as follows:

- If `cfgSnmptTrapType` is **trap(1)** then this engine ID is used as authoritative engine ID in the notification.

- If `cfgSnmpTrapType` is **inform(1)** then this engine ID is enforce to be used as authoritative engine ID in the SNMPv3 discovery process. If the receiver has a different engine ID, sending notifications will fail.

Examples:

- auto
- 0x80001F8801C0A80014
- 0x80001F880300145A035042
- 0x80001F880441424344454647484950

Type	DisplayString
Range	4 - 66
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.12.10.6

5.1.13 cfgSnmpTrapUser

SNMP User / Security Name

This parameter is active when `cfgSnmpTrapVersion` is set to **v3usm(2)**.

Set the user name used for authenticated SNMPv3 notifications.

Type	DisplayString
Range	1 - 255
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.12.10.9

5.1.14 cfgNtpServer1

NTP Server 1

The primary NTP server to update the time from.

If the IP is set to 0.0.0.0 the NTP client will only listen to broadcast packages and `cfgNtpServer2` is ignored.

This entry is an IPv4 address or an FQDN (Fully Qualified Domain Name). An FQDN can only be resolved when a nameserver is configured in `cfgSysNameserverTable` or a nameserver is received via DHCP.

Examples:

- pool.ntp.org
- 192.168.1.2
- 0.0.0.0

Type	DisplayString
Range	1 - 255
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.14.2

5.1.15 cfgNtpServer2

NTP Server 2

The secondary NTP server to update the time from. Used as a fallback when server 1 can not be reached.

When `cfgNtpServer1` is set to 0.0.0.0 this entry is ignored.

This entry is an IPv4 address or an FQDN (Fully Qualified Domain Name). An FQDN can only be resolved when a nameserver is configured in `cfgSysNameserverTable` or a nameserver is received via DHCP.

Examples:

- pool.ntp.org
- 192.168.1.2
- 0.0.0.0

Type	DisplayString
Range	1 - 255
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.14.3

5.1.16 cfgNetVlanComment

User Comment

This parameter has no operational function. It allows to store a comment about the use of this VLAN.

Type	DisplayString
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.2.3.1.19

5.1.17 cfgNetMacVlanComment

User Comment

This parameter has no operational function. It allows to store a comment about the use of this MacVlan.

Type	DisplayString
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.2.9.1.19

5.1.18 cfgRouteTableMetric

Metric of the Route

The metric allows to create multiple routes to the same destination. The higher a metric, the less priority the route has.

Range	0 - 2147483647
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.4.2.1.10

5.1.19 cfgRouteTableRoutingTables

Routing Tables on Which Route are Created

This is a space and/or comma separated list of tables on which the route will be created.

Specify 254 to create routes on the main table.

Valid values are > 0 and < 2000000000. However in this range there are reserved values that may not be used:

- 128: prelocal
- 253: default
- 255: local

Use `cfgRouteRuleTable` to create policies which use the tables specified here.

Examples:

- 5000

- 254, 7000
- 100 254, 8000

Type	DisplayString
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.4.2.1.11

5.1.20 cfgRouteTableMonitor

NLM Monitor Handling This Route

This is a reference to an NLM instance (`cfgNlmMonIndex`).

The referenced monitor has to be of type route (see `cfgNlmMonType`).

Set to -1 to not handle this route by a monitor.

CARP (`cfgNetCarpTable`) and NLM monitoring are exclusive.

Range	-1 - 255
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.4.2.1.12

5.1.21 cfgRouteTableWeight

Weight for ECMP

When multiple gateways for the same destination exist, connections are distributed to the gateways according to their weight. Connections are distinguished on IP source and IP destination.

Use Case:

- The net 192.168.0.0/24 is reachable via 10.0.1.1 and 172.16.1.1
- 10.0.1.1 has a weight of 1 and 172.16.1.1 has a weight of 2
- When 3 connections are opened, 1 connection is sent via 10.0.1.1 and 2 connections via 172.16.1.1

Range	1 - 256
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.4.2.1.14

5.1.22 cfgRouteTableGateway

Gateway to Destination Network

Specify the IP address of the gateway over which the destination specified in `cfgRouteTableDestinationNetwork` is reachable.

Alternatively options are:

- `unreachable`: Create a route that blocks traffic and responds with ICMP code 1 (Host unreachable)
- `prohibit`: Create a route that blocks traffic and responds with ICMP code 13 (Communication administratively filtered)
- `blackhole`: Create a route that silently drops all traffic
- `throw`: Create a route that skips processing on the current routing table. May be used for policy routing

<i>Type</i>	DisplayString
<i>Range</i>	5 - 50
<i>Access</i>	readwrite
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.4.2.1.5

5.1.23 cfgRouteTableInterface

Interface of Static Route

The interface on which the static route is created. Usually this is not required, and the system determines automatically to which interface a route applies to.

Leave this on `none` unless you need an explicit interface for a route. For example, this is required when using IPsec with VTI (see `cfgVpnIpsecGlblVirtualTunnelInterface`).

<i>Type</i>	DisplayString
<i>Range</i>	1 - 15
<i>Access</i>	readwrite
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.4.2.1.9

5.1.24 cfgRouteRuleTable

Routing Rules

Also called `policy routing`.

These rules may be used to match frames based on:

- Source (`cfgRouteRuleFrom`)
- Destination (`cfgRouteRuleTo`)
- Input Interface (`cfgRouteRuleInputInterface`)
- TOS (Type of Service, QoS) value (`cfgRouteRuleTos`)

These frames are then processed with the specified routing table in `cfgRouteRuleLookupTable`.

Each rule should have its own unique preference set via `cfgRouteRulePreference`. Thus multiple overlapping rules may be created.

<i>Range</i>	0 - 255
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.4.4

5.1.25 `cfgRouteRuleLookupTable`

Routing Table to Lookup

Frames which match the configured criteria are processed by the routing table specified here.

The default routing table is table 254 - the main table. All traffic not explicitly directed to a different table is processed by the main table.

Reserved routing table numbers may not be used and are:

- 128: prelocal
- 253: default
- 255: local

<i>Range</i>	1 - 1999999999
<i>Access</i>	readwrite
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.4.4.1.12

5.1.26 `cfgRouteRuleEnabled`

Disable or Enable this Rule

<i>Enumeration</i>	disabled (0), enabled (1)
<i>Access</i>	readwrite
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.4.4.1.2

5.1.27 cfgRouteRulePreference

Preference of the Rule

The preference defines in what order rules are evaluated. This enables multiple overlapping rules. The lower the value, the higher the priority.

Each match should have it's own unique preference.

<i>Range</i>	10000 - 1999999999
<i>Access</i>	readwrite
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.4.4.1.3

5.1.28 cfgRouteRuleFrom

Source Network

This is an address or network in CIDR notation.

Set to 0.0.0.0/0 to match any source.

<i>Type</i>	DisplayString
<i>Access</i>	readwrite
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.4.4.1.4

5.1.29 cfgRouteRuleTo

Destination Network

This is an address or network in CIDR notation.

Set to 0.0.0.0/0 to match any destination.

<i>Type</i>	DisplayString
<i>Access</i>	readwrite
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.4.4.1.5

5.1.30 cfgRouteRuleInputInterface

Input Interface

Match an interface on which traffic is received. If the interface is the loopback `lo`, only traffic originating on the device itself is matched. Thus different rules for local traffic and forwarding traffic can be created.

Set to `any` to match any interface.

Type	DisplayString
Range	1 - 15
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.4.4.1.6

5.1.31 `cfgRouteRuleTos`

Type Of Service

Match frames based on the TOS field in their IP header according to RFC 791.

	7		6		5		4		3		2		1		0	
+	-----	+	-----	+	-----	+	-----	+	-----	+	-----	+	-----	+	-----	+
	PRECEDENCE					TOS					Unused(0)					
+	-----	+	-----	+	-----	+	-----	+	-----	+	-----	+	-----	+	-----	+

Thus valid values are:

- 0
- 4
- 8
- c
- 10
- 14
- 18
- 1c

Set to `any` to match any TOS value.

Type	DisplayString
Range	1 - 3
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.1.1.4.4.1.8

5.1.32 cfgRouteDhcpTable

DHCP Received Routes

This table may be used to specify the metric and table of routes received via DHCP. When using ECMP also the weight may be changed. To dynamically bring the routes up and down, a monitor may be specified.

<i>Range</i>	0 - 255
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.4.5

5.1.33 cfgRouteDhcpEnabled

Disable or Enable This Entry

<i>Enumeration</i>	disabled (0), enabled (1)
<i>Access</i>	readwrite
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.4.5.1.2

5.1.34 cfgRouteDhcpInterface

Interface of DHCP Client

The name of an interface on which a DHCP Client is running.

Examples:

- wlan0
- br0.vlan7

<i>Type</i>	DisplayString
<i>Range</i>	1 - 15
<i>Access</i>	readwrite
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.4.5.1.3

5.1.35 cfgRouteDhcpMetric

Metric of Routes Received

The metric allows to create multiple routes to the same destination. The higher a metric, the less priority the route has.

Set to -1 to not explicitly set a metric. Depending on the DHCP client a different metric is implied:

- **wwan**: 600
- **eth**: 400
- **wlan**: 400
- **ovpn**: 200

<i>Range</i>	-1 - 2147483647
<i>Access</i>	readwrite
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.4.5.1.4

5.1.36 cfgRouteDhcpRoutingTables

Routing Tables on Which Received Routes are Created

This is a space and/or comma separated list of tables on which the route will be created.

Specify 254 to create routes on the main table.

Valid values are > 0 and < 2000000000. However in this range there are reserved values that may not be used:

- 128: prelocal
- 253: default
- 255: local

Use `cfgRouteRuleTable` to create policies which use the table specified here.

Examples:

- 5000
- 254, 7000
- 100 254, 8000

<i>Type</i>	DisplayString
<i>Access</i>	readwrite
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.4.5.1.5

5.1.37 cfgRouteDhcpMonitor

NLM Monitor Handling of Routes Received

This is a reference to an NLM instance (`cfgNlmMonIndex`).

The referenced monitor has to be of type route (see `cfgNlmMonType`).

Set to -1 to not handle this route by a monitor.

<i>Range</i>	-1 - 255
<i>Access</i>	readwrite
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.4.5.1.6

5.1.38 `cfgRouteDhcpWeight`

Weight for ECMP of Routes Received

When multiple gateways for the same destination exist, connections are distributed to the gateways according to their weight. Connections are distinguished on IP source and IP destination.

Use Case:

- The net 192.168.0.0/24 is reachable via 10.0.1.1 and 172.16.1.1
- 10.0.1.1 has a weight of 1 and 172.16.1.1 has a weight of 2
- When 3 connections are opened, 1 connection is sent via 10.0.1.1 and 2 connections via 172.16.1.1

<i>Range</i>	1 - 256
<i>Access</i>	readwrite
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.4.5.1.7

5.1.39 `cfgRouteGlobal`

<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.4.6
------------	---------------------------------

5.1.40 `cfgRouteGlbReversePathFilter`

Reverse Path Filter

The reverse path filter has 3 modes of operation:

- **none(0)**: No source address validation is performed and any packet is forwarded to the destination network.
- **strict(1)**: Strict Mode as defined in RFC 3704. Each incoming packet to a router is tested against the routing table and if the interface that the packet is received on is not the best return path for the packet then the packet is dropped.
- **loose(2)**: Loose mode as defines in RFC 3704 Loose Reverse Path. Each incoming packet is tested against the route table and the packet is dropped if the source address is not routable through any interface. This allows for asymmetric routing where the return path may not be the same as the source path.

<i>Enumeration</i>	none (0), strict (1), loose (2)
<i>Access</i>	readwrite
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.1.4.6.2

5.1.41 rpcCellular

<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.3.101
------------	---------------------------------

5.1.42 rpcCellFwUpgrade

Upgrade Cellular Module

Before starting a firmware upgrade of the cellular module, define a valid URL, which is accessible by the device. This URL can be specified by `setCellFwUrl`.

Writing **upgrade(1)** to this parameter downloads the firmware from the specified URL and upgrades the cellular module.

This RPC parameter follows the standard behaviour, thus a readout indicates the status of the upgrade process. Whilst the upgrade is running, **upgrade(1)** is returned. Otherwise, it returns **nop(0)** or **error(-1)** to indicate that the upgrade is complete or failed.

<i>Enumeration</i>	error (-1), nop (0), upgrade (1)
<i>Access</i>	readwrite
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.3.101.3

5.1.43 setCellular

<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.4.50
------------	--------------------------------

5.1.44 setCellFwUrl

Cellular Firmware URL

This URL specifies where the upgrading process is downloading the cellular firmware.

Supported protocols are: tftp and http.

Example:

- http://192.168.1.1/firmware.rec

<i>Type</i>	DisplayString
<i>Range</i>	1 - 255
<i>Access</i>	readwrite
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.4.50.1

5.1.45 setCellSimSlot

Cellular SIM Slot Selection

This parameter allows to switch the SIM slot at runtime.

If the selected slot is already active or no SIM card is inserted, no action is performed.

<i>Range</i>	1 - 2
<i>Access</i>	readwrite
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.4.50.2

5.1.46 swSysMsgEpoch

Message timestamp as Epoch.

<i>Type</i>	Integer32
<i>Access</i>	readonly
<i>OID</i>	1.3.6.1.4.1.16177.1.400.1.6.3.100.1.5

5.1.47 swSysMsgDate

Message timestamp as date string.

Example:

- Sun Feb 20 18:07:53 2022

Type	DisplayString
Range	1 - 255
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.6.3.100.1.6

5.1.48 swCfgHash

Configuration Hash

Hash in hexadecimal of the current applied configuration which can be used to verify the configuration integrity.

During boot-up or during apply of a new configuration this element will return an empty string.

The used hash algorithm is SHA384.

Type	DisplayString
Range	0 - 96
Access	readonly
OID	1.3.6.1.4.1.16177.1.400.1.6.30.5

5.2 MIB Reference: WESTERMO-SW6-FIREWALL-MIB

5.2.1 cfgFwNatOneToOneTable

Netmap Table

The netmap table allows to rewrite the source respective destination of a whole network. This is an advantage over the `cfgFwNatPortForwardTable` and `cfgFwNatOutboundTable` which only allow to rewrite a single source respective a single destination.

It provides 3 types of operation:

- Auto mode where inbound and outbound frames automatically have their source, respectively destination, rewritten.
- DNAT mode where inbound frames to the specified `cfgFwNatOneToOneDestinationNet` have their destination rewritten to `cfgFwNatOneToOneRewriteNet`
- SNAT mode where outbound frames from the the specified `cfgFwNatOneToOneSourceNet` have their source rewritten to `cfgFwNatOneToOneRewriteNet`.

<i>Range</i>	0 - 255
<i>OID</i>	1.3.6.1.4.1.16177.1.400.2.1.1.2.3

5.2.2 `cfgFwNatOneToOneEnabled`

Disable or Enable this Rule

<i>Enumeration</i>	disabled (0), enabled (1)
<i>Access</i>	readwrite
<i>OID</i>	1.3.6.1.4.1.16177.1.400.2.1.1.2.3.1.2

5.2.3 `cfgFwNatOneToOneType`

Type of Rule

This parameter defines the type of netmap:

- **auto(0)**: A rule in auto mode creates automatically a DNAT and an SNAT rule for the specified source, destination and rewrite networks, also known as 1:1 NAT.
- **dnat(1)**: A rule in DNAT mode rewrites the destination network to the specified rewrite network.
- **snat(2)**: A rule in SNAT mode rewrites the source network to the specified rewrite network.

<i>Enumeration</i>	auto (0), dnat (1), snat (2)
<i>Access</i>	readwrite
<i>OID</i>	1.3.6.1.4.1.16177.1.400.2.1.1.2.3.1.3

5.2.4 `cfgFwNatOneToOneInterface`

Interface

Depending on the selected type in `cfgFwNatOneToOneType`:

For the types **auto(0)** and **dnat(1)** it refers to the interface on which traffic ingresses (input interface).

For the type **snat(2)** it refers to the interface on which traffic egresses (output interface).

Set to -1 to perform netmap on traffic ingressing/egressing on any interface, only constrained by the configuration in `cfgFwNatOneToOneSourceNet` and `cfgFwNatOneToOneDestinationNet`.

Groups of interfaces can be matched by adding the character + at the end. E.g. `eth+` to match the interfaces `eth0`, `eth1`.

Examples:

- -1
- `br0.vlan0`
- `br0.vlan+`
- `wlan0`
- `eth0`

Type	DisplayString
Range	1 - 15
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.2.1.1.2.3.1.4

5.2.5 cfgFwNatOneToOneSourceNet

Source Network

This is a source network in CIDR notation.

When `cfgFwNatOneToOneType` is set to **auto(0)** or **dnat(1)**, frames matched by this field are processed by this rule.

Set to `0.0.0.0/0` to match all inbound traffic.

When `cfgFwNatOneToOneType` is set to **snat(2)** the network specified here will be rewritten to the network specified in `cfgFwNatOneToOneRewriteNet`.

Type	DisplayString
Range	9 - 19
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.2.1.1.2.3.1.5

5.2.6 cfgFwNatOneToOneDestinationNet

Destination Network

This is a destination network in CIDR notation.

When `cfgFwNatOneToOneType` is set to **auto(0)** or **dnat(1)**, the network specified here will be rewritten to the network specified in `cfgFwNatOneToOneRewriteNet`.

When `cfgFwNatOneToOneType` is set to **snat(2)**, only frames matched by this field are processed by this rule.

Set to `0.0.0.0/0` to match all outbound traffic.

Type	DisplayString
Range	9 - 19
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.2.1.1.2.3.1.6

5.2.7 cfgFwNatOneToOneRewriteNet

Rewrite Network

This is the network in CIDR notation to/from which frames are rewritten.

Depending on the setting of `cfgFwNatOneToOneType`:

- **auto(0)**: For inbound traffic `cfgFwNatOneToOneRewriteNet` is the network to which the destination is rewritten to `cfgFwNatOneToOneDestinationNet`. For outbound traffic `cfgFwNatOneToOneRewriteNet` is the network which matches the source that is rewritten to `cfgFwNatOneToOneRewriteNet`.
- **dnat(1)**: `cfgFwNatOneToOneRewriteNet` is the network to which the dest. net (`cfgFwNatOneToOneDestinationNet`) is rewritten to.
- **snat(2)**: `cfgFwNatOneToOneRewriteNet` is the network to which the source net (`cfgFwNatOneToOneSourceNet`) is rewritten to.

Type	DisplayString
Range	9 - 19
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.2.1.1.2.3.1.7

5.2.8 cfgFwNatOneToOneComment

User Comment

This parameter has no operational function. It allows to store a comment about the use of this netmap rule.

Type	DisplayString
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.2.1.1.2.3.1.8

5.2.9 cfgFwFltRlcmpType

ICMP Type

When `cfgFwFltRProtocol` is set to **icmp(1)** this parameter may be used to filter specific ICMP types.

Set to -1 to match any type.

For a list of types officially assigned by IANA see: <https://www.iana.org/assignments/icmp-parameters/icmp-parameters.xhtml>

Examples:

- **any(-1)**
- **echo reply(0)**
- **destination unreachable(3)**
- **redirect(5)**
- **echo(8)**
- **tll exceeded(11)**

When blocking type 3, be aware that this breaks PMTUD (Path MTU Discovery).

Range	-1 - 255
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.2.1.1.4.10.1.15

5.2.10 cfgFwFltRlcmpCode

ICMP Code

When `cfgFwFltRProtocol` is set to **icmp(1)** this parameter may be used to filter specific ICMP codes.

Depending on `cfgFwFltRlcmpType` different types have different codes. This parameter has no function when `cfgFwFltRlcmpType` is set to **any(-1)**.

Set to -1 to match any code.

For a list of codes of each type officially assigned by IANA see: <https://www.iana.org/assignments/icmp-parameters/icmp-parameters.xhtml>

Examples:

- **any(-1)**
- **Protocol Unreachable(2)** of type destination unreachable(3)
- **Redirect Datagram for the Host(1)** of type redirect(5)

<i>Range</i>	-1 - 255
<i>Access</i>	readwrite
<i>OID</i>	1.3.6.1.4.1.16177.1.400.2.1.1.4.10.1.16

5.2.11 cfgFwFltRLog

Enable Logging For This Rule

Configures whether frames handled by this rule are logged.

<i>Enumeration</i>	disabled (0), enabled (1)
<i>Access</i>	readwrite
<i>OID</i>	1.3.6.1.4.1.16177.1.400.2.1.1.4.10.1.17

5.2.12 cfgFwFltRLimit

Rule Frame Limit

This parameter is active when `cfgFwFltRAction` is set to **accept(1)** and `cfgFwFltRLimitBurst` has a value greater than 0.

This is the rate at which ACCEPT-tokens are added to the bucket for this rules.

In the form of `rate[/second|/minute|/hour|/day]`

The maximum rate is 10'000/second, respective 600'000/minute, 36'000'000/hour and 864'000'000/day.

Examples:

- 1/second
- 5/minute
- 100/hour
- 10000/day

Type	DisplayString
Range	1 - 13
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.2.1.1.4.10.1.18

5.2.13 cfgFwFltRLimitBurst

Rule Frame Limit Burst

This parameter is active when `cfgFwFltRAction` is set to **accept(1)**.

This is the size of the bucket containing ACCEPT-tokens, for this rule. Each new frame consumes 1 token. When the bucket has zero tokens, new frames are dropped.

New tokens are added to the bucket at the rate specified in `cfgFwFltRLimit`.

Set to 0 to disable the rate limiter.

This value is also the amount of tokens in the bucket at start.

Range	0 - 10000
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.2.1.1.4.10.1.19

5.2.14 cfgFwFltLogDefaultDrop

Disable or Enable Logging of the Default Drop Policy

When this option is set to **enabled(1)**, all frames that are dropped by the default policy on a chain are logged.

Enumeration	disabled (0), enabled (1)
Access	readwrite
OID	1.3.6.1.4.1.16177.1.400.2.1.1.4.4

5.2.15 cfgFwFltLogDefaultAccept

Disable or Enable Logging of the Default Accept Policy

When this option is set to **enabled(1)**, all frames that are accepted by the default policy on a chain are logged.

<i>Enumeration</i>	disabled (0), enabled (1)
<i>Access</i>	readwrite
<i>OID</i>	1.3.6.1.4.1.16177.1.400.2.1.1.4.5

5.2.16 cfgFwFltInputSynLimit

Input Syn Frame Limit

This parameter is active when `cfgFwFltInputSynLimitBurst` has a value greater than 0.

This is the rate at which ACCEPT-tokens are added to the bucket for the TCP SYN flood protection on the INPUT chain.

In the form of rate[/second|/minute|/hour|/day]

The maximum rate is 10'000/second, respective 600'000/minute, 36'000'000/hour and 864'000'000/day.

Examples:

- 1/second
- 5/minute
- 100/hour
- 10000/day

<i>Type</i>	DisplayString
<i>Range</i>	1 - 13
<i>Access</i>	readwrite
<i>OID</i>	1.3.6.1.4.1.16177.1.400.2.1.1.4.6

5.2.17 cfgFwFltInputSynLimitBurst

Input Syn Frame Limit Burst

This is the size of the bucket containing ACCEPT-tokens, for the TCP syn flood protection on the INPUT chain. Each new TCP SYN frame consumes 1 token. When the bucket has zero tokens, new TCP SYN frames are dropped.

New tokens are added to the bucket at the rate specified in `cfgFwFltInputSynLimit`.

Set to 0 to disable the INPUT TCP SYN rate limiter.

This value is also the amount of tokens in the bucket at start.

<i>Range</i>	0 - 10000
<i>Access</i>	readwrite
<i>OID</i>	1.3.6.1.4.1.16177.1.400.2.1.1.4.7