Prepared by	Document	
Jon-Olov Vatn	Release Notes WeOS 4.32.2	
Approved by	Date	Document No
Fredrik Pettersson	July 4, 2022	089604-g53ec73b

Release Notes WeOS 4.32.2

Contents

Prepared by	Document	
Jon-Olov Vatn	Release Notes WeOS 4.32.2	
Approved by	Date	Document No
Fredrik Pettersson	July 4, 2022	089604-g53ec73b

Prepared by	Document	
Jon-Olov Vatn	Release Notes WeOS 4.32.2	
Approved by	Date	Document No
Fredrik Pettersson	July 4, 2022	089604-g53ec73b

Legal Information

The contents of this document are provided “as is”. Except as required by applicable law, no warranties of any kind, either express or implied, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose, are made in relation to the accuracy and reliability or contents of this document. Westermo reserves the right to revise this document or withdraw it at any time without prior notice. Under no circumstances shall Westermo be responsible for any loss of data or income or any special, incidental, and consequential or indirect damages howsoever caused. More information about Westermo can be found at the following Internet address: http://www.westermo.com.

Prepared by	Document	
Jon-Olov Vatn	Release Notes WeOS 4.32.2	
Approved by	Date	Document No
Fredrik Pettersson	July 4, 2022	089604-g53ec73b

About

Westermo WeOS is a network operating system designed for industrial grade rugged Ethernet switches and routers. Fully supporting WeOS units on Basis platform (Lynx 2^{nd} gen., Wolverine 2^{nd} gen., Viper 2^{nd} gen., and Falcon), Corazon platform (RedFox 2^{nd} gen.) and Coronet platform (Viper-A, a.k.a, Viper 3^{rd} gen.).

WeOS is a Linux based software platform that has been in operation since 2006 on custom made RedFox Mil, RedFox Aero and RedFox Rail products. With the advent of the RedFox Industrial line of products the platform was given a major overhaul to improve standards compliance as well as compatibility requirements with existing Westermo product offerings. The result is WeOS, the Westermo Operating System.

For more information about Westermo and other product offerings see http://www.westermo.com/.

Version Number Format

WeOS version numbers have three fields. The main reason for the third field is to emphasise the difference between feature and bug fix releases.

The generally available (GA) releases are named 4.X.Y. The number four (4) denotes the platform generation. The X is the feature release number, where new functionality is introduced, and Y is the patch revision number, reserved for security and bug fix releases. E.g., 4.15.1 would be the first patch release in the 4.15.0 series.

For customers in our beta release program it is worth pointing out that previously version numbers 9.00 – 9.99 were used for beta releases and developer builds. This custom has now been replaced by the more common –betaN notation, for internal and limited distribution beta releases, and –rcN, for release candidates. We believe this to be easier to keep track of since the base release version is visible in all stages of the release cycle.

Prepared by	Document	
Jon-Olov Vatn	Release Notes WeOS 4.32.2	
Approved by	Date	Document No
Fredrik Pettersson	July 4, 2022	089604-g53ec73b

1 Summary of Changes

1.1 News in WeOS 4.32.0

WeOS 4.32.0 contains new or updated features as described in the subsections below. It also includes bug fixes as listed in section 3.1.

1.1.1 Modbus Firewall (DPI)

The WeOS firewall has been extended with a Deep Packet Inspection (DPI) filter for *Modbus/TCP* traffic. This filter allows for matching on Modbus *function code*, *unit ID* and *register address*, in addition to the matching settings available for regular packet filter rules. See section *DPI filtering* in the Firewall chapter of the Management Guide for more information.

CLI Example (full firewall output not shown). Allow *function* codes *1-4* (the most common read functions) to pass through the firewall to *destination 198.18.3.99*. Implicit rule(s) will be automatically added after the last DPI rule dropping all other Modbus traffic, including Modbus write requests. Another implicit rule is also added to let the first TCP packet through, with state NEW.

```
example:/#> config
example:/config/#> ip
example:/config/ip/#> firewall
example:/config/ip/firewall/#> dpi allow proto tcp dport 502 dst 198.18.3.99 modbus function 1-4
example:/config/ip/firewall/#> show dpi
  1 dpi allow dst 198.18.3.99/32 proto tcp dport 502 modbus function 1-4
example:/config/ip/firewall/#> leave
example:/#> show firewall
...
=== Forwarding Packet Filter Rules ========================================
Forwarding Policy DROP
target  prot in   out  source     destination
ACCEPT  tcp  any  any  anywhere   198.18.3.99     tcp dpt:modbus modbus fc 1:4
DROP    tcp  any  any  anywhere   anywhere        tcp dpt:modbus modbus         <= implicit
ACCEPT  tcp  any  any  anywhere   198.18.3.99     tcp dpt:modbus state NEW      <= implicit
...
example:/#>
```

1.1.2 Ability to Disable HTTP and HTTPS Services Individually

It is now possible to disable HTTP and HTTPS individually, using 'no port' or 'no ssl-port' settings in the Web Configuration Context. The typical use case would be to disable HTTP, while keeping HTTPS enabled, as shown in the example below.

```
example:/#> configure
example:/config/#> web
example:/config/web/#> no port
example:/config/web/#> leave
```



```
example:/#> show running-config web
Press Ctrl-C or Q(uit) to quit viewer, Space for next page, <CR> for next line.
web
        no port
        ssl-port 443
        end
example:/#>
```

Note: 'no port' and 'no ssl-port' commands both have new semantics in 4.32.0 and onward. Now they mean 'disable' HTTP and HTTPS respectively, while in earlier releases they implied reset to default port ('port 80' and 'ssl-port 443' respectively).

See the 'General System Settings' chapter of the WeOS Management Guide for information on how to manage the Web service.

1.1.3 'ipconfig' Service Limited to Discovery and IP Settings

The *ipconfig* service is used by the WeConfig tool for discovery of WeOS4 units and for initial configuration of IP settings. With WeOS 4.32.0, the ability to conduct upgrade via the ipconfig service has been removed.

See the 'General System Settings' chapter of the WeOS Management Guide for information on how to manage the ipconfig service.

1.1.4 New Default Boot-loader Version

Units shipped with 4.32.0 hold a new version of the Barebox boot-loader (2017.12.0-8). This is also the boot-loader version included in the WeOS 4.32.0 package file.

This new Barebox version contains no changes applicable for WeOS 4 units as compared to the previous versions included in the WeOS 4.29, 4.30 and 4.31 releases. The reason for including this new version is that Westermo's ambition is to have a common preferred Barebox version both for WeOS 4 and WeOS 5.

In general, there is no need to upgrade WeOS units in the field to the latest boot-loader, however, see section 7.4 for special considerations.

1.1.5 Fixed CVEs

WeOS 4.32.0 includes fixes for the following CVE(s):

- CVE-2018-12684
- CVE-2020-27304

- CVE-2022-0778

1.1.6 Web/HTTPS Support for TLS v1.0 and v1.1 Removed

As part of fixing Web issue #18426 (section 3.1) and for security reasons, the Web HTTPS access support for TLS v1.0 and v1.1 has been removed. Thus, customers using Web/HTTPS to manage the WeOS units are requested to use Web clients supporting TLS v1.2.

1.2 News in WeOS 4.32.1

WeOS 4.32.1 is a bug fix release, see section 3.2 for a list of fixed issues.

One of them fixes an issue (#18748) introduced in WeOS 4.32.0. The bug prohibited adding and editing firewall filter rules via the Web interface. Customers using WeOS 4.32.0 and managing the unit via the Web have this specific reason to upgrade, but other customers are also recommended to upgrade to WeOS 4.32.1.

Note: The 4.32.1 release notes includes information on a behaviour change introduced in WeOS 4.32.0, see section 1.1.6 above. Information on this change was missing in the WeOS 4.32.0 release notes.

1.3 News in WeOS 4.32.2

WeOS 4.32.2 is a bug fix release, see section 3.3 for a list of fixed issues. The release also includes CVE fixes, see below.

1.3.1 Fixed CVEs

WeOS 4.32.2 includes fixes for the following CVE(s):

- CVE-2021-44225
- CVE-2018-19115

2 Known Limitations

This section includes known reported bugs and missing features, which may not necessarily be *limitations*, in many cases they may constitute severe operational drawbacks.

2.1 Platform

- A system with many VLANs setup requires more time at boot. This was first reported in #3291, but even after having fully optimised all data paths there still remains a significant delay. E.g., creating 128 VLANs on a RedFox Industrial takes apx. 6 seconds longer than creating a single VLAN.
- The new alarm configuration lacks support for RMON triggers.
- Running an FRNT ring over copper SFPs is not recommended, due to slow response time from copper SFPs.
- Limited support for low–level interaction with PHYs and link partners.
- Port monitoring fails to preserve the VLAN priority, issue #4152.
- When toggling bridge priority on the elected root bridge storm is easily provoked, issue #4203. Avoid reconfiguring bridge priority at runtime.
- In some setups when RSTP gets link up it has been reported to take very long to reconfigure, issue #4707.

2.2 CLI

- When issuing, e.g., `show running` not all settings are shown. This is due to WeOS 4.3.0 and later only showing differences to the system default. Support for `show running [all]`, where the optional 'all' keyword would list everything, is planned for a later release.
- The on–line help is not only insufficient, it is sometimes even misleading. E.g., some commands do not support the `no` prefix, some commands do not support `show` and no commands in configure context support `repeat`. Cleanup and improvement is a work in progress.

2.3 NAT translation on 'Basis' products

When using a product based on the 'Basis' platform as NAT router, it may fail to translate the source IP address when experiencing high network loads close to the units' routing capacity. To mitigate this issue, it is either recommended to limit the offered routing load, to adapt ingress rate-limit settings on adequate ports, or to adapt the unit's CPU bandwidth limit setting.

For information about products and which platform they are based on, see the 'Introduction' Chapter in the WeOS Management Guide.

2.4 DHCP Static Lease Preemption

The WeOS DHCP Server can be configured to preempt statically assigned leases. The preempt setting can be helpful to facilitate replacing (client) units getting their IP address statically.

Known limitation: In order for lease preemption to work, the DHCP server needs to have 'ping-check' enabled.

2.5 DHCP classless static routes

While WeOS now supports configuring up to 512 DHCP option 121/249 routes when acting as a DHCP server, this feature is not without limitations. Discussed below are two cases where DHCP clients may not receive any or all of the routes configured on the server. These cases are somewhat extreme and should be avoided by considering the DHCP server configuration.

- DHCP message size limitation

  The first case is related to the maximum allowed size of the DHCP message. This parameter is set by the client and transmitted to the DHCP server as option 57 in DHCPDISCOVER and/or DHCPREQUEST messages (see RFC2132 9.10 for more information). WeOS, when acting as a DHCP client, sets this parameter to the lowest allowed value of 576 bytes.

  When this option is sent by the client, the server is responsible for ensuring that DHCP messages sent back to the client do not exceed the given message size. If it would, the server may handle this situation in a number of different ways - for instance, it might discard the lease request completely, or try to remove options from the message so as to make it fit below the size limit. While WeOS will attempt to do the latter, options 121 and 249 will often contain large amounts of data if many routes have been configured on the DHCP server. This can lead to options 121 and/or 249 being stripped from the DHCP offer to the client, and the client thus never receiving the routes.

  This issue is partially dependent on the DHCP client, but can be mitigated by performing the following actions:

  – If possible, use only one of options 121 and 249.

    Doing this reduces the size of the data used for routes by up to nearly half. WeOS currently does not support configuring different routes to be used for options 121 and 249, and whether the WeOS DHCP server will reply with either or both of these options depends on whether the client requests them.

  - Ensure that any individual lease will only contain the routes it needs.

    This can be achieved by carefully choosing which routes are put in the global, subnet and host scopes, and by consolidating several smaller routable CIDR networks to fewer larger ones.

  This issue starts to become relevant at around 15 routes sent to a single client, if the minimum legal DHCP message size is requested, and if options 121 and 249 are used simultaneously. Due to certain peculiarities in how the DHCP options are defined, this limit is somewhat dependent on the routes in question.

- DHCP server limitation

  When a high number of routes is configured for an individual scope - including routes inherited from other scopes - the DHCP server may be unable to correctly function, leading to a loss of DHCP service. This limit is at about - again, depending on the specific values used - 33 routes.

  This issue can be mitigated by performing the following action:

  - Ensure that any individual lease will only contain the routes it needs.

    This can be achieved by carefully choosing which routes are put in the global, subnet and host scopes, and by consolidating several smaller routable CIDR networks to fewer larger ones.

2.6 IEEE 802.1X

When changing the VLAN configuration (of any VLAN) on a unit acting as IEEE 802.1X authenticator, authorized supplicants will be treated as unauthorized until a new successful 802.1X authentication handshake has been performed.

2.7 MRP

MRP ring ports have to be on the same slot when used on products with multiple slots. MRP does not support aggregate ports.

2.8 SNMP

The SNMP chapter of the WeOS Management Guide lists supported standard MIBs, including limitations to specific tables for some MIBs. Additional deviations from the standard MIBs may exist. For some MIBs, you find more detailed MIB conformance information in the WeOS release zip-archive.

2.9 Web

- Inspecting RMON counters in the Port Statistics page may need a manual reload before the actual values are displayed.
- Due to security reasons the username and password must always be provided when logging in, i.e auto-completion is not supported in the login form.

2.10 IPsec

- MTU override may not work as expected, sending a message over the IPsec tunnel will not respect mtu override on the other end. **Workaround:** Always have the same MTU on the interfaces on both ends of the tunnel.
- The remote IP address of the IPsec gateway may in some circumstances not be reachable from an IP address associated with the IPsec tunnel.
  Workaround: Always connect to an IP on the IPsec gateway that is reachable from within the tunnel.
- DPD restart/clear is sometimes unreliable. If the responder is configured with dpd-action clear and then rebooted, the tunnel will sometimes not be renegotiated.
  Workaround 1: If only using static IP addresses and only one initiator, change both nodes to be initiators and set dpd-action hold on both sites.
  Workaround 2: Use a ping trigger towards an IP inside the tunnel and connect the ping trigger to a tunnel action (See section 5.18). Remember to set 'retrain interval 30' in the action configured, this will create another level of DPD outside the IKE traffic, but it will be encrypted.
  Workaround 3: If it is not possible to use DPD hold (multiple initiators or not static IP) you can on the initiator(s), create a tunnel action (see section 5.18) and set it to be retrained after a few seconds. Use a ping trigger and set peer as an IP inside the tunnel, and connect it to the tunnel action.

2.11 SSL VPN

WeOS implements SSL VPN using OpenVPN with the following known limitations

- To be able to use dynamic or static routing over a ssl interface you will have to use a layer 2 tunnel. Layer 3 tunnels will not work as expected in this scenario.
- No support for revocation lists
- No check for certificate type, a client certificate can be used as a server certificate and reversed.
- When using layer3, OpenVPN supports multiple topologies, p2p, net30 and subnet, WeOS only support subnet.

- Not possible to add a layer2 interface to a VLAN.

2.12 Link Aggregation

WeOS supports link aggregation in line with IEEE 802.3ad. However, the current support for link aggregation contains several limitations such as:

- VLAN support: There is no support to add a link aggregate to a VLAN. Instead, each of the individual member links need to be added to the appropriate VLANs.
- Port settings: There is no support to configure port settings for the link aggregate. Instead, each of the individual member ports need to be configured uniformly, e.g., with respect to port speed/duplex mode.
- Only link aggregation of Ethernet ports is supported. Configuration of SHDSL ports or xDSL ports (ADSL/VDSL) ports in an aggregate, or mixing Ethernet and SHDSL/xDSL ports in an aggregate, may be possible, but this is not supported and the behaviour is undefined.

2.13 Bandwidth Limiting

The frames per second (fps) mode for the ”traffic shaping” setting (layer-2 feature) is

- only supported on Ethernet ports (not on DSL ports)
- not supported on certain RedFox models. See management guide for further details.

Note: There are currently *no warnings* when the ”fps” does not apply for traffic shaping. In such cases, the rate setting will ignore the fps attribute and interpret the rate in bps.

2.14 Flow Control

The WeOS flow control support has the following two limitations:

- When enabling *802.3x flow-control* on a port, that port will permanently operate in flow-control mode without involving Ethernet auto-negotiation mechanisms.
- When configuring flow-control in ”slot based” WeOS products (for example RFI), a port with 802.3x flow-control enabled will only send Pause frames when causing congestion on another port in the same slot; not when causing congestion on a port in another slot. For example, if port 1/1 has flow control enabled, it will send Pause frames when causing congestion on on port 1/2, but not when causing congestion on port 2/1. Similar restrictions apply to WeOS Viper, RedFox Rail (RFR) and RedFox Industrial Rack (RFIR) products.

2.15 LLDP

The WeOS LLDP support has the following known limitation:

When connected to Windows 10 PC with LLDP driver enabled, the WeOS unit will not respond to SNMP requests for the “remsysname” LLDP SNMP OID. The proposed work-around is to disable the LLDP driver on the Windows 10 PC.

2.16 TTDP (IEC 61375-2-5)

This limitation only applies RFR-212-FB (the only WeOS 4 product capable of running IEC 61375-2-5 TTDP).

The WeOS 4 TTDP implementation sends TTDP TOPOLOGY frames with the *ETBN-CN-CNX* field(s) encoded in ’big endian’ byte order. There has been (and still is) some confusion on how to interpret the standard, and there are vendors encoding this field either as ’big endian’ or as ’little endian’.

The WeOS 4 TTDP implementation only inter-operates with TTDP implementations capable of encoding the ETBN-CN-CNX field as ’big endian’.

3 Fixed Issues

3.1 WeOS 4.32.0

Fixed issues in WeOS 4.32.0 (as relative to 4.31.0).

Issue	Category	Description
#18721	IGMP	Querier information cleared upon access-port link down in proxy-querier mode
#18673	IGMP	IGMP does not always update correctly when main Querier is lost
#18669	Logging	Turning off LLDP on ports fills log with disabled-messages every 10 min
#18660	Ports	Running config incorrectly lists auto-neg-capability setting for SFP ports
#18654	NTP	Setting NTP server address via DHCP option is not accepted by DHCP client
#18649	MRP	MRP secondary port is not sending all TC messages
#18632	System	Custom interface MTU not applied if interface lacks IP address
#18631	CLI	WeOS 4.31 skips % when using it in (aaa) 'password' command
#18594	IGMP	IGMP filters do not age out on proxy devices
#18520	PoE	Small fluctuations in PoE effect monitoring may disable low priority PoE port if total effect is close to maximum
#18426	WEB	SSL Handshake failing towards HTTPS web interface via satellite link
#18423	SNMP	MAU-MIB ifMauAutoNegTable not properly supported
#18370	FRNT	DDW 225/226 FRNT and RSTP LEDs always active on bootup
#18351	Logging	Syslog with domain name as target not working if failing to resolve at boot
#17551	WEB	Inbound rate limit not the same in WEB as in CLI
#17531	System	DDNS client hard-coded to do forced update only once a week
#14307	System	A SSH session will not time out if idle timeout is longer than keepalive

Note: Issue #14307 was fixed already in WeOS 4.29.0, but was not included in the list of fixed issues until release notes of 4.32.0.

3.2 WeOS 4.32.1

Fixed issues in WeOS 4.32.1:

Issue	Category	Description
#18755	Documentation	Text in 3rd party licence document wider than column width
#18748	WEB	Adding or editing firewall 'filter' or 'modify' rule in Web gives error message stating a DPI rule is needed
#18737	FRNT	Occasional storm when FRNT member neighbour to focal point starts up
#18468	LED	On LED blinking constantly (instead of 10 seconds) after Web Login or when in CLI config context
#18264	Ports	Port speed and duplex not set correctly after bootp configuration file download fails

Note: Issue #18264 was fixed in WeOS 4.29.0, but was not included in the list of fixed issues until release notes of 4.32.1.

3.3 WeOS 4.32.2

Fixed issues in WeOS 4.32.2:

Issue	Category	Description
#18811	MRP	MRM does not respect 'react-on-link-change' setting, sending too many 'Topology Change' messages upon MRC link-down
#18770	VRRP	VRRP virtual MAC address still in FDB while in state backup
#18760	MRP	MRM does not send 'MRP_TopologyChange' on secondary port upon MRC link-up

4 Known Issues

Known issues as of WeOS 4.32.2.

Issue	Category	Description
#18761	QoS	Port priority is not saved in config
#18752	802.1X	RADIUS NAS-Port-Type field reports 'Wireless' instead of 'Ethernet'
#18734	LLDP	IP address is not sent in the LLDP information
#18636	CLI	CLI does not allow "?" when configuring admin password using cleartext
#18610	FRNT	Synchronized reboot of two FRNT nodes may occasionally give storm
#18586	Ports	Ingress rate limit lets through 50% more than the configured rate on ports connected to MV6352 switchcore (Viper-A/RFI)
#18552	802.1X	RADIUS server using local host entry as FQDN failing to resolve on startup and delays retrying for extended period of time
#18529	WEB	Compressed log files downloaded from web can not be opened
#18521	System	Setting non-default VLAN CPU channel stops traffic on slot 1 ports (RedFox)
#18480	FRNT	SHDSL port not progressing to Forwarding on FRNTv2 topology change
#18446	VPN	SSL tunnel with password protected PEM certificate can lock up client side
#18395	FRNT	Connecting 100 and 1000 Mbit FRNT ports cause packet loss on link-up
#18390	802.1X	802.1X with RADIUS group does not try next RADIUS server if failing to contact first
#18373	AAA	RADIUS request has 127.0.1.1 in NAS-IP_Address field
#18327	RIP	RIP IP Membership Failing to Add Membership
#18323	NTP	When changing NTP setting one of the interfaces loses all receive packets
#18308	SNMP	FRNTv0 instance available in WESTERMO-WEOS-MIB but missing in WESTERMO-FRNT-MIB
#18247	TCN	Prio handling of the TTDP Hello and Topology telegrams
#18118	DHCP	Problems with DHCP relay when used over a VPN tunnel
#18108	VPN	Openvpn on does not reestablish SSL tunnel when 'no compression' and 'push-network' feature is used
#18001	GRE	Ping alarm Trigger towards GRE/lo interface not responding over IPSEC/GRE tunnel

Issue	Category	Description
#17950	SNMP	SNMP response for lldpremsysname sometimes miss neighbour hostname information
#17845	Ports	Applying (tech preview) 'storm control' trigger on disabled ports will enable them in absence of storm
#17602	WEB	Selecting PAF changes SHDL port 2 to CO from CPE automatically
#17597	Alarm	Automatic selection of source IP for SNMP Traps causes problems for WeConfig to display the alarm.
#17554	OSPF	OSPF network type misleading for OSPF over GRE (type is Point-to-Point)
#17220	RSTP	STP sending BPDUs on non STP-port's after reconfiguration.
#16970	WEB	Configuring DHCP Server static lease matching clientid (option 61) via Web lacks '01' prefix
#16801	Serial	Termination for RS485 not working
#16652	VRRP	Not possible to access services (SSH, etc) on VRRP router when sending to its virtual IP address
#15885	System	IPsec/SSL peer domain names longer than 65 characters truncated (should be 253)
#14661	VPN	OpenVPN: CLI reports tunnel up when interface is down
#14363	DSL	Communication over DSL port sometimes uni-directional after recovery
#14325	DHCP	DHCP-server refuses to (re-)deliver option 82 static and client ID based leases after configuration change.
#14041	Firewall	DHCP relay configured with multiple VLANs may not be able to handle server response (dropped in firewall)
#12717	QoS	ARP packets treated with lowest priority
#12663	Ports	CLI freezes when connecting a gigabit fiber link with autoneg disabled to a WeOS device with autoneg on
#10516	VPN	VPN LED for SSL VPN server should indicate up only when at least one client is connected
#10336	Serial	Serial driver sometimes introduce gaps between characters in a data stream
#6180	System	RedFox 8FX: System instability issues with 1000Mbps fiber in 100Mbps SFP slot
#4929	RSTP	Looping admin edge ports causing a storm
#4707	RSTP	Long reconfiguration time for RSTP at link up, up to 32 sec
#4203	RSTP	Storm occurs quite frequently when toggling RSTP bridge priority

The following three issues have been removed from the list of *known issues* in 4.32.1 and 4.32.2, and

are instead listed as *known limitations*, see sections 2.4 (#18353), 2.6 (#17295), and 2.3 (#14378) respectively.

Issue	Category	Description
#18353	DHCP	Lease preempt setting does not work with 'ping-check' disabled
#17295	802.1X	802.1X will wipe authenticated users if changes are made to a VLAN
#14378	Firewall	NAT fails to translate source IP at high packet load ('Basis' products)

5 Technology Previews

WeOS contains hidden and undocumented features called technology previews. Westermo provides no support for undocumented features. Features specifically marked as tech previews can be completely redesigned, removed or changed in such a way that after an upgrade they are *not guaranteed* to work!

The following is by no means a complete list, but details features that may become supported in the next upcoming feature release.

- *Password Encryption*: CLI only, section 5.1
- *Trigger to control Interface Admin Distance*: CLI only, section 5.2
- *Setting default gateway per interface*: CLI only, section 5.3)
- *SFP DDM Alarm and SNMP Trap Support*: Alarm settings in CLI only, section 5.4
- *Serial Low Latency*: CLI only, section 5.5
- *SNMP MIBs for RIP and OSPF:* See section 5.6
- *NTP Server with GPS support*: CLI only, section 5.7
- *TFTP Server*: CLI only, section 5.8
- *Guest User*: CLI only, section 5.10
- *New VLAN Features:* CLI only support for disabling 'secure' mode and MAC address learning. See section 5.11
- *Additional SSH settings:* CLI only support for setting SSL port, idle-timeout and keepalive interval for the WeOS SSH server. See section 5.12
- *Additional Telnet settings:* CLI only support for setting Telnet port for the WeOS Telnet server. See section 5.13
- *Tcpdump:* CLI only. See section 5.14
- *CLI welcome message:* Ability to set custom CLI welcome message. See section 5.15
- *Additional DDNS features*: See section 5.16
- *USB boot*: CLI only, section 5.17.

  Separate feature from "USB Autobackup/restore" and "USB Configuration Deployment"!
- *IPsec and SSL VPN extensions:* Ability for IPsec initiators to be configured with two responder addresses (*IPsec Backup Peer*), IPsec and SSL VPN tunnels can be enabled/disabled via alarm trigger and action, and SSL VPN tunnels can be configured using a standard OpenVPN configuration file (.ovpn). See section 5.18.
- *PPPoE Server*: CLI only, section 5.19

- *Serial HDX mode*: CLI only, section 5.20
- *DHCP client arping option*: CLI only, section 5.21
- *Support for disabling DHCP snooping in DHCP relay agent*: CLI only, section 5.22
- *Firewall conntrack flushing*: CLI only, section 5.23
- *RSTP support for VLAN tagging of BPDU's*: CLI only, section 5.24
- *Remote IO support*: CLI only, section 5.25
- *Storm control*: CLI only, section 5.26

5.1 Password Encryption

Passwords for PPP, DDNS, RADIUS, IPsec secrets, SNMP v2 community strings, etc., are by default stored in clear text in the WeOS configuration. As of WeOS 4.15.0 these strings can be encrypted using a built-in secret key to provide a very basic level of security. This is by no means a cryptographically secure encryption, and can possibly more be likened to obfuscation rather than true encryption. Nevertheless, it is likely good enough for most users.

To enable password encryption in the running configuration and save it to the startup configuration, simply type:

```
example:/#> config
example:/config/#> encrypt passwords
example:/config/#> leave
example:/#> copy run start
```

To further secure an installation the user can provide a custom encryption key. This key will be device specific and must be entered again if exporting the configuration to another device. The key can be at most 64 characters long and will be securely[1] stored in built-in flash of the device to be able to boot.

```
example:/#> config
example:/config/#> encrypt passwords key XYZZY
example:/config/#> leave
example:/#> copy run start
```

To change custom key from 'XYZZY' to 'QWERTY' the user will be prompted to input the current custom encryption key. This prompt will not appear when changing from the default built-in key. To change from a custom key back to the default built-in key type:

```
example:/#> config
example:/config/#> encrypt passwords default
Configuration encrypted with a custom key, please input current key.
Password: *****                              (Silent prompt, no feedback)
```

[1] The custom key is in itself encrypted before stored in a file on built-in flash.


```
example:/config/#> leave
example:/#> copy run start
```

Password encryption is a per-file feature of WeOS. If you change to another configuration file, using the `copy` command, that file determines if password encryption is enabled or disabled. When changing to a file *with* encryption you can be prompted for its secret key, if a custom secret key was used to encrypt its passwords.

When disabling password encryption, using the `no encrypt` command, all password strings will be scrambled using a random secret key. This maybe seems a bit unintuitive, but is a security measure to protect your secrets from being decrypted by someone with access to a copy of your encrypted configuration and rogue WeOS device.

A factory reset, using crossed cables or the factory reset login on the console, will wipe all configurations, including any custom secret keys.

5.2 Interface Admin Distance Trigger

In a setup with multiple upstreams, e.g. at least two different Internet Service Providers (ISPs) on two separate VLAN interfaces, the device selects the primary ISP based on the configured interface distance.

To enable fail-over between multiple ISPs a *ping trigger* can be configured and connected to the interface's distance. (CLI only atm.)

As long as the ping reaches its target beyond the ISP, e.g. BBC.com, the interface distance remains at its configured setting, but when the trigger fires, due to BBC.com becoming unreachable via that ISP, the distance is automatically adjusted to 255 (infinity) for the associated default route.

While connectivity to BBC.com via the primary ISP is down, the secondary ISPs default route will be used instead, but as soon as connectivity is restored the system will fall-back to the primary ISP again.

```
example:/config/iface-vlan1/#> distance 10 trigger 2
```

NOTE: Make sure to configure the trigger (ID 2 in this example) to use the *correct outbound interface*, otherwise the ping will use the default route, and you will get interesting flapping.

5.3 Interface Gateway

As a sign of things to come it is also possible to set the gateway address on interfaces with a static address. This as a complement to the possibility to setup a default route in IP configuration context.

Future additions will include DNS and NTP servers, as well as domain search prefix configurable on a per-interface basis, all activated according to the interface distance.

```
example:/config/iface-vlan1/#> gateway 192.168.2.1
```

Please note, this setting is only available for interfaces of type **inet static**.

5.4 SFP DDM Alarm

Support to read status of Westermo DDM SFPs has been available since WeOS 4.13.1.

As a tech preview, there is now support for SFP DDM Alarm handling (of Westermo DDM SFPs), including SNMP DDM Alarm Trap support in the Westermo Private MIB. SFP DDM alarm triggers are configurable from the CLI only. Alarm can be configured set from voltage, bias-current, temperature, rx-power and tx-power.

Example:

```
example:/config/#> alarm
example:/config/alarm/#> trigger ddm-temperature              (Create DDM temp. trigger)
example:/config/alarm/trigger-3/#> port 1                        (Trigger on SFP port 1)
example:/config/alarm/trigger-3/#> threshold rising 60        (Rising threshold at 60°C)
example:/config/alarm/trigger-3/#> threshold falling 58      (Falling threshold at 58°C)
```

For 'rx-power', the alarm condition should be set to 'low' as a low value means a weaker signal. For values below the falling threshold, alarm will be indicated.

```
example:/config/#> alarm
example:/config/alarm/#> trigger ddm-rx-power                 (Create DDM temp. trigger)
example:/config/alarm/trigger-4/#> condition low                   (Alarm on low signal)
example:/config/alarm/trigger-4/#> port 1                        (Trigger on SFP port 1)
example:/config/alarm/trigger-4/#> threshold rising -20    (Rising threshold at -20 dBm)
example:/config/alarm/trigger-4/#> threshold falling -25  (Falling threshold at -25 dBm)
```

5.5 Serial Low Latency

Tech preview of *Serial Low Latency* is configurable, from the CLI only. The Serial Over IP application is extended with an additional mode: <seriallowlatency>. This mode is only for use in a point-to-point application where one serial port is connected to the remote unit's serial port over SHDSL (Wolverine units) or Ethernet. No addressing possibility exist.

The Serial Low Latency function is optimised for transferring serial characters at the lowest possible latency to the remote unit with as low jitter as possible. This function is only valid for one instance of serial over IP.

Syntax:

```
example:/config/#> seroip 1
example:/config/seroip-1/#> mode seriallowlatency
example:/config/seroip-1/#> port <SERIAL-PORT>
example:/config/seroip-1/#> remote-frame-delay
example:/config/seroip-1/#> remote-frame-delay <0-2147483647>                (0..2^31)
```



```
example:/config/seroip-1/#> remote-frame-size <0-512>
example:/config/seroip-1/#> local-frame-delay <0-2147483647>          (0..2^31)
example:/config/seroip-1/#> local-frame-size <0-512>
example:/config/seroip-1/#> iface <IFNAME>
```

In order to keep characters back-to-back, a data-packing algorithm has been implemented.

-delay parameters are in micro seconds,

-size parameters are in number of characters

local- parameters are used to configure the data-packet algorithm for characters received from the serial port before sending them to the remote unit

remote- parameters are used to configure the data-packet algorithm for characters received from the network side (SHDSL or Ethernet) before transmitting them to the physical serial port.

5.6 SNMP RIPv2-MIB and OSPF-MIB

Tech preview support of SNMP MIBs:

- RFC1724 RIP Version 2 MIB
- RFC1850 OSPF Version 2 MIB

5.7 NTP Server with GPS Time-Base Support

As described in the WeOS Management Guide, the WeOS unit can act both as NTP client and NTP server. However, the supported NTP server functionality assumes the WeOS unit has a remote NTP server as reference clock, e.g., as shown below:

```
example:/config/ntp/#> server pool.ntp.org
```

Th local NTP server is enabled as follows:

```
example:/config/ntp/#> listen              (Start NTP Server on all interfaces)
```

Hint: To limit access to the NTP server to a specific interface you currently have to use the WeOS firewall functionality (firewall functionality is available on the 200-series of WeOS products).

Technology preview additions to NTP server function:
As a complementary technology preview, it is possible to let the WeOS unit be the top-level NTP server, using it's own system clock as source to synchronise time on connected NTP clients.

WeOS also have support for using a GPS receiver connected using RS232/422/485 and use it as a reference clock for NTP. This requires a preconfigured GPS receiver, it has to be configured to send NMEA reports. The correct serial port configuration also has to be entered into the serial port context in the CLI. Pulse per second (PPS) is currently not supported, but may be supported in future releases.

Enable GPS support (technology preview):

```
example:/config/#> gps 1                                    (Create GPS instance '1')
example:/config/gps-1/#> port 1                  (Use GPS attached to serial port '1')
example:/config/gps-1/#> end
example:/config/#> ntp
example:/config/ntp/#> gps 1                (Define GPS instance '1' as clock source.)
```

In the example above, the GPS was attached to serial port ”1”. Additional configuration of serial port 1 (e.g., bit-rate) may be required to match your GPS.

5.8 TFTP Server

Units running software level WeOS Extended can act as TFTP server in the network. This can be a very useful feature when combining it with the BOOTP configuration deployment, introduced in 4.13. Make sure you have an USB stick inserted in the USB port and then enable the TFTP server.

```
example:/config/tftp-server/#> path usb://
```

5.9 Preferred (Remote) NTP Server

If more than one clock source is configured (multiple remote NTP servers, or a remote NTP server and a local GPS), the unit will synchronise to the source with the best *stratum*. To override this behaviour, an NTP server could be configured to be *preferred*.

```
example:/#> configure
example:/config/#> ntp
example:/config/ntp/#> server ntp.example.com
example:/config/ntp/server-ntp.example.com/#> preferred
```

5.10 Guest User

Basic guest user support is now possible to enable in WeOS. The reserved local username ’guest’ must be setup in AAA configuration context to enable this feature:

```
example:/config/aaa/#> username guest guest
Adding new user guest.
example:/config/aaa/#> leave
example:/#> exit
example login: guest
Password:
example:/$>
```


The guest account is very restricted, e.g., it cannot configure the system, read passwords by from configuration files, or otherwise manipulate the state of the system. Only inspect status of ports, VLANs, interfaces and RMON, and do basic network debugging using, e.g., ping or traceroute.

5.11 New VLAN Features

By default WeOS VLANs are setup in 'secure' mode, IEEE 802.1q, so any traffic that, e.g., tries to ingress with an unknown[2] VLAN tag is silently dropped.

With some equipment, or in some setups, this 'secure' mode is not desired behaviour. A user may simply want traffic to pass-through the switch unaffected. For this purpose it is now possible to disable the secure mode on a per VLAN basis.

```
example:/config/vlan-1/#> no dot1q
```

It is also possible to disable MAC address learning on a per VLAN basis.

```
example:/config/vlan-1/#> no learning
```

5.12 Additional SSH Server Settings

For the WeOS SSH server it is possible to tweak the default settings for

- Listening (TCP) port (default: 22)
- Keepalive interval (default: 60 seconds)
- Idle timeout (default: Disabled)

These settings are available in the SSH server configuration context (CLI only), for example:

```
example:/#> configure
example:/config/#> ssh
example:/config/ssh/#> port 12345
```

5.13 Additional Telnet Server Settings

For the WeOS Telnet server it is possible to change the default listening port (CLI only).

```
example:/#> configure
example:/config/#> telnet
Activating Telnet with default settings, type 'abort' to cancel.
example:/config/telnet/#> port 34567
```

Note: For security reasons, the Telnet server is disabled by default.

[2] I.e., a VLAN ID not configured for the given port, in either tagged or untagged mode.

5.14 Packet Capture – TCP Dump

Previously only available to developers and support personnel, this release now adds support for the tcpdump packet capture tool in the CLI. Due to the design of the device's hardware, it is not possible to capture packets on a per-port basis (Layer-2), only per interface (Layer-3), but if a single port is setup in a VLAN the effect will in most cases be the same. With the exception of certain control traffic like IGMP, RSTP, FRNT, 802.1X, etc. Such frames will not be possible to capture, unless the functions in WeOS are completely disabled.

The exposed tcpdump features are limited, but should be sufficient for most use-cases. One such feature is the ability to save the PCAP files to a USB stick, if the device is equipped with a USB port.

See the online help in the CLI for more information and some useful examples to get started.

```
example:/#> tcpdump vlan1
```

5.15 CLI Welcome Message

Support for personalising the WeOS CLI welcome banner is another new feature. It is now possible to add a message that shows up before login, different depending on trying to login from from the console or SSH, and another after successful login, called Message of the Day, or MOTD.

- **Console login:**
  `example:/config/system/#>` **[no] issue <MESSAGE>**
- **SSH login:**
  `example:/config/system/#>` **[no] issue-net <MESSAGE>**
- **Message of the Day:**
  `example:/config/system/#>` **[no] motd <MESSAGE>**

Example:

```
example:/config/system/#> issue "Company Inc. Gateway | Welcome operator!"
example:/config/system/#> issue-net "Only authorized personnel!\nThis
session is logged!"
example:/config/system/#> motd "Site policy:\n o Do not change live
   system!\n o Contact sysadmin for help or system problems."
```

The messages of the 'issue' and 'issue-net' settings are limited to 64 characters, while the 'motd' message can be longer.

5.16 Additional DDNS Features

More DDNS Providers

The WeOS DDNS client, Inadyn, now has support for a few more DDNS providers: 3322, ZoneEdit, easyDNS, DNS-O-Matic, ChangeIP, nsupdate.info, DuckDNS, and Loopia. This in addition to the already supported: DynDNS, FreeDNS, and No-IP.

HTTPS/SSL Support

Some DDNS providers support HTTPS update, this WeOS 4.15.1 and later support an SSL check box in the WebUI and a 'ssl' setting in the CLI to enable this feature. Please note, you need to make sure your DDNS provider supports this before enabling SSL.

Forced Update

The DDNS client only sends an update to your DDNS provider when the IP address changes, and a forced update every week. In some cases, however, you may need to manually force an update. Currently this is only possible from the CLI (web support is planned for a later release)

```
example:/#> ip ddns update
```

See the system log file for both the action and results of the update. The actual DNS update may take a while to propagate to your Internet Service Provider (ISP), so please don't issue this command multiple times thinking this will speed up the process. It all depends on how your DNS record is setup at the DDNS provider.

Also, if you do this too often some DDNS providers will disable your account, or your DNS entry, for excessive updates. This is a policy of the DDNS provider.

5.17 USB Boot

An exciting USB function referred to as "USB boot" is available as technology preview. Instead of using the USB stick as (continuous) backup, it can also be used to boot from. This has been available from WeOS 4.6.0, but is still only a technology preview. The directory structure used in 4.6.0 has changed in 4.8.0. To activate this, on the unit, simply log in to the CLI.

```
example:/#> boot
example:/boot/#> boot-order usb
```

See the WeOS Management Guide (in particular the chapter "General Switch Maintenance") for more information on USB support (bootstrap settings, supported USB sticks and file systems, etc.)

5.18 IPsec and SSL VPN Extensions

- Import an SSL tunnel using a OpenVPN configuration file This can configure up a SSL VPN tunnel in WeOS using a standard OpenVPN configuration file (.ovpn) either with the certificates embedded or you can upload the certificates on your own.

```
server:/#> tunnel import ssl 100 tftp://192.168.1.2/test.ovpn
```

- IPsec Backup Peer This is a technology preview of upcoming IPsec redundancy support.

  IPsec initiators may be configured with two responder addresses. If IPsec fails to connect to the primary responder, it will try to connect to the backup responder. The primary responder will periodically be checked, and a switch back is initiated if possible.

```
example:/config/alarm/#> trigger ping
example:/config/alarm/trigger-1/#> peer 192.168.22.2
example:/config/alarm/trigger-1/#> end
example:/config/alarm/#> end
example:/config/#> tunnel
example:/config/tunnel/ipsec-0/#> backup 192.168.23.2 trigger 1
example:/config/alarm/#> leave
```

- Enable/disable IPsec and SSL VPN tunnels via alarm trigger and action

```
example:/config/alarm/action-4/#> target tunnel
example:/config/alarm/action-4/#> tunnel ipsec 0
```

  This can be used to for example have a service tunnel that you want to enable from digital in. In that case you just create a digital in trigger and connect it to the action created above. When the trigger is “true”, the VPN tunnel will be enabled.

  It is also possible to retrain the tunnel, to not keep it in error state, this is useful if you want to restart the tunnel on an event:

```
example:/config/alarm/action-4/#> retrain interval 30
```

5.19 PPPoE Server

PPPoE *server* support is available from the CLI only. This allows a WeOS unit to serve up to 16 PPPoE clients, using a local user database for client authentication and authorisation.

By default, the PPPoE server will have address `10.2.0.1`, and hand out addresses in the range `10.2.0.2 - 10.2.0.9`, but this can easily be changed, as is illustrated in the following example:

```
example:/config/pppoe-0/#> server                    (Enables PPPoE server)
example:/config/pppoe-0/#> address 1.2.3.4           (Set local IP address)
example:/config/pppoe-0/#> pool 1.2.3.10 10                    (Using size)
example:/config/pppoe-0/#> pool 1.2.3.10 1.2.3.19             (Using range)
```


Use local user database ”0” to authenticate/authorise PPPoE clients:

```
example:/config/pppoe-0/#> aaa-auth local-db 0
```

5.20 Serial HDX Mode

Serial HDX mode, to handle legacy Serial – V.23 HDX application/equipment with RTS/CTS and DCD control on serial port devices like the Lynx, Wolverine and the Falcon. Currently available from the CLI only.

```
example:/config/serial-1/#> hdx                    (Enable HDX mode)
example:/config/serial-1/#> no hdx
```

This function is limited to handle RTS-CTS delay of 23 ms and a guard time for the DCD signal of 10 ms. By default, serial HDX mode is disabled.

5.21 DHCP Client ”ARP Ping” Option

This allows a WeOS unit to disable the “ARP ping” setting in the DHCP client.

```
example:/config/iface/vlan-1/#> no arping          (Disable ARP ping)
example:/config/iface/vlan-1/#> arping             (Default, enabled)
```

If the arping setting is disabled, the DHCP client in WeOS sets the IP address assigned by the DHCP server without first performing an ARP ping of the new IP Address.

By default, the arping option is enabled.

5.22 Support for Disabling DHCP Snooping in DHCP Relay Agent

The MV88E6097 chipset has a bug in its DHCP snooping functionality where it can mistake UDP fragments for DHCP frames. Customers have reported this as an issue when using the Network File System (NFS). As of WeOS 4.15.0 there exist a CLI only setting in the DHCP relay agent to disable the DHCP snooping feature of the MV88E6097 chipset on a per-port basis.

```
example:/config/dhcp-relay/#> option82
example:/config/dhcp-relay/#> port eth 1
example:/config/dhcp-relay/port-eth1/#> no snooping    (Default, enabled)
```

The problem with disabling DHCP snooping is that in “flat” networks where the DHCP client, relay and server are in the same broadcast domain (same LAN). The DHCP server will receive two DHCP requests from the client. The recommended workaround is to run the DHCP server on a different port, e.g. 6767, and have the relay agent forward all requests to the server on that port. That way the server will ignore broadcasted DHCP requests. However, this requires that all client requests pass through a relay agent, which in many setups may not be possible.

5.23 Firewall Conntrack Flushing

This CLI only feature controls automatic flushing of the firewall connection tracking mechanism on route changes.

The connection tracking mechanism is an optimisation in the firewall. Firewall rules are only evaluated once per connection, and are placed in a cache. This cache speeds things up for the rest of the packets belonging to the same session.

This may have some side effects if dynamic routing is enabled. A deny rule on a specific interface may not be respected if a connection is enabled through some other interface and then moves to the interface through dynamic routing events.

Enabling automatic flushing on route events makes traffic to be re-evaluated in the firewall at route changes, thus solving this problem.

Note: NAT also uses the same connection tracking cache for its internal state. Flushing the cache may result in that existing *NAT:ed connections can break and reset*. Please use with care!

Since WeOS 4.14.2, this setting *flushes everything* in the connection tracking cache at routing events. This feature will be changed in a future version to enable a more selective flushing that will avoid flushing connections that are not affected by a specific routing change.

```
example:/config/ip/firewall/#> conntrack-flush routes     (Flush at route events)
example:/config/ip/firewall/#> conntrack-flush auto       (Default, no flushing)
```

5.24 RSTP Support for VLAN Tagging of BPDUs

WeOS 4.15.1 introduced very experimental support for transmitting VLAN tagged RSTP frames (BPDU's). This can be used to interface with equipment from other manufacturers on a limited set of ports.

```
example:/config/spanning-tree/#> stp-port 1                          (Select port/ports)
example:/config/spanning-tree/stp-port-eth1/#> vid 1234  (Set VLAN tag 1234 on all BPDU's)
```

This feature does not affect reception of RSTP BPDUs, nor does it introduce support for RSTP per VLAN, or any similar variant offered by other manufacturers. All it does is on a per-port basis enable a WeOS device to add an IEEE 802.1Q VLAN tag to all BPDUs egressing an RSTP port.

At this point it is unclear if this feature will ever become anything more than a technology preview. Instead, later versions of WeOS are more likely to add actual RSTP per VLAN support, or even true MSTP.

5.25 Remote IO Support for Digital Output

WeOS 4.18.0 introduced support for remote control of the digital output. The remote control is done via CGI web requests. WeOS 4.19.0 added SNMP trap for remote IO control.

The remote IO control is by default inactive and need to be configured and enabled before use. You also need to configure the alarm and trigger with Remote IO to activate the digital output.

Example of how to enable Remote IO via CLI configuration:

```
example:/config/#> alarm
example:/config/alarm#> trigger rio-cgi
example:/config/alarm/trigger-2#> leave
```

Alarm action 1 is used unless an other action is set. The standard configuration action 1 sets the digital out as one of its targets.

The digital output is active for a certain time. For making the output active for 25000 milliseconds (25 s) request this URL:

```
http://192.168.2.200/cgi-bin/adm/io?timer1=25000
```

5.26 Storm Control

In WeOS 4.18.0 the tech preview storm control alarm was introduced. Storm control consists of a trigger (storm-detect) and an action (shutdown-port). The storm-detect trigger monitors the traffic (frames per second) for a selected port. If the traffic count exceeded the threshold the trigger will activate the selected alarm action. The storm-detect trigger can be made persistent. The action shutdown-port disables a selected port.

In WeOS 4.20.0 the storm control feature remains a tech preview but has been changed slightly:

- The feature has been temporarily limited to Lynx-family devices.
- The action has been renamed to 'port' (from 'shutdown-port')

When applying storm control to the network, the network has to be set up to minimise disturbances from the storm. E.g. the following settings on other devices in the network has to be considered:

- CPU-bandwidth limiting should be set on the devices where storm control is set up.
- FDB aging-timeout, since a storm may replace correct FDB entries and thus traffic may be directed the wrong way.

Also note that use of the *1100-0148 SFP* will disturb the storm-control functionality in the device and the pertinent SFP is not recommended for use together with storm control.

Example of a storm-control configuration via CLI.:


```
example:/config/#> alarm
example:/config/alarm/#> action 2
example:/config/alarm/action-2/#> target port led
example:/config/alarm/action-2/#> end
example:/config/alarm/#> trigger storm-detect
example:/config/alarm/trigger-2/#> action 2
example:/config/alarm/trigger-2/#> port 1-2
example:/config/alarm/trigger-2/#> persistent 300
example:/config/alarm/trigger-2/#> threshold 500
example:/config/alarm/trigger-2/#> end
example:/config/alarm/#> show
Trigger  Type          Enabled    Action  Source
================================================================================
      1  frnt          YES             1  Instance 1
      2  storm-detect  YES             2  1-2

Action  Targets
================================================================================
      1  snmp log led digout
      2  log led port
```

6 Accessing the Command Line Interface

The WeOS switch supports a classic Command Line Interface (CLI) that can be accessed via the console port at 115200@8N1 or Secure Shell (SSH), for details see the Secure Shell RFC4251. WeOS supports protocol version 2 only.

Issue `help` or `show tutorial` at the prompt to access the built–in help and tutorials. See the WeOS Management Guide for more information.

Recommended Clients

UNIX OpenSSH, `http://www.openssh.com`

Win32 PuTTY, `http://www.chiark.greenend.org.uk/~sgtatham/putty/`, note that PuTTY is also useful for connecting to serial port consoles.

Please follow the directions for installation and usage applicable to your system and client.

Logging In

To gain access to the CLI you need:

- An SSH client
- The switch IP#
- The user name and password

Units shipping with WeOS have by default all ports assigned untagged to VLAN 1[3], and is configured to acquire an IP address via DHCP, but also with a static IP address: 192.168.2.200 with netmask 255.255.255.0. The unit's will also be reachable via a link-local address, i.e, an address in range 169.254.x.x (where 'x' is a number 0-255).

Use the WeConfig tool, an LLDP client or nmap to find your device. If you have a DHCP server available you can set it up to hand out a known IP addresses for the registered devices MAC addresses. Each unit comes with 16 or 32 MAC addresses assigned, depending on the port count, the base address should be printed on the box and on the unit itself.

The unit is fairly quick to boot, in under 10 seconds is the unit up requesting an IP address — depending on the existence of a DHCP server the fall back to link–local address can take a while. To be on the safe side while scanning for your device, expect it to take anything from 30 seconds to one minute after power–on.

[3]Falcon units come with a slightly different factory configuration. The Ethernet ports on Falcon belong to VLAN1 and are reachable via IP address 192.168.2.200. The xDSL port belongs to VLAN1006 and use DHCP for address assignment.

The following example illustrates how to login to the switch using OpenSSH from a GNU/Linux based host system. The process is similar with PuTTY or other SSH clients.

The operator lists the running configuration with the command `show running`, an overview of ports, vlans and interfaces is available by typing `show ports`, `show vlans` and `show ifaces`. See the `help` or the `show tutorial` for more on line help.

To change some settings, enter the configuration context with the command `conf`, short for "configure". The same commands as shown above also apply here, but now display configured settings.

To show or change the interface and VLAN properties the operator uses the command: `interface vlan2` and `vlan 2`, respectively, with an optional "show" as prefix. E.g. `show iface vlan2`.

To leave a level the operator must use the command `end` to save and `abort` to cancel.

Any new settings are activated only when the operator leaves the configuration context, using "end".

To save settings to non–volatile RAM (flash disk), the operation uses `copy run start` from admin–exec context.

```
$ ping -b 192.168.2.255
PING 192.168.2.255 (192.168.2.255) 56(84) bytes of data.
64 bytes from 192.168.2.200: icmp_seq=1 ttl=64 time=10.4 ms
64 bytes from 192.168.2.200: icmp_seq=2 ttl=64 time=0.895 ms
^C

$ ssh admin@192.168.2.200
The authenticity of host '192.168.2.200 (192.168.2.200)' can't be established.
RSA key fingerprint is 1d:ce:fe:4b:8e:c2:73:42:11:68:73:02:e5:a6:e4:8b.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '192.168.2.200' (RSA) to the list of known hosts.
admin@192.168.2.200's password: westermo

.--.--.--.-----.-----.------.-----.-.--.--------.-----.
|  |  |  |  -__|__ --|_    _|  -__|   _|  .  .  |  _  | http://www.westermo.com
 \__/\__/|_____|_____| |__| |_____|__| |__|__|__|_____|        info@westermo.se
   Robust Industrial Data Communications -- Made Easy

\\/ Westermo WeOS 4.9.2 4.9.2 -- Oct  2 16:01 CEST 2012
Type: 'help' for help with commands, 'exit' to logout or leave a context.
example:/#> ^D
$
```

This is a typical session where broadcast ping is first used to locate the device, and then SSH login using the default user and password. (Hint: Use the WeConfig tool to locate your WeOS devices.)

7 Firmware Upgrade

Firmware upgrade is supported from the CLI, Web and WeConfig tool. All of them support FTP/TFTP upgrade, but the Web also supports CGI upload from the browser – making it the ultimate choice if you have no FTP/TFTP server available or do not care to set one up.

The version string listed in the output from the `show system-information` command is only updated after reboot.

7.1 What Firmware Image to Use

Since version 4.13.1 of WeOS, unified firmware package files are supported (pkg-files). The package file include WeOS images and boot loaders for all supported products.

WeOS-4.32.2.pkg: All products, WeOS 4.32.2

7.2 Upgrading the Primary Image

Many find it easiest to upgrade firmware via Web interface or via the WeConfig tool. This section gives additional information on upgrading via the CLI.

To be able to upgrade the switch firmware the user must install and run an FTP server or a TFTP server on a network connected to the device. The (T)FTP upgrade uses anonymous login with the password 'support@westermo.se'.

The example below shows that the upgrade command first tries FTP and then TFTP, should the FTP connection fail. Here the unit is upgraded WeOS-4.13.1.pkg, but you can replace this with WeOS-4.32.2.pkg.

```
example:/#> upgrade primary 192.168.2.3 WeOS-4.13.1.pkg

==> Upgrade in progress, console disabled.  Please stand by ... <==

Connecting to 192.168.2.3:21 (192.168.2.3:21)
WeOS-4.13.1.pkg      100% |*******************************| 58549k  0:00:00 ETA

Checking download ...
Unpacking weos (from /upgrade/download)...
Setting up weos (4.13.1-1)...

Checking rw4131.img ...
   Type: CramFS
     ID: OK (RFox)
   Size: OK
    CRC: OK 0xD5A84E26
```

```
Flashing currently active MTD partition, reboot is forced.

Flashing /dev/mtd1 ...
100% \ [=======================================================================]

Updating RedBoot directory with new CRC ...
100% | [=======================================================================]

Done.

Rebooting system ...
Restarting system.
```

The system will force a reboot when upgrading the primary image. This to protect against flash corruption issues seen in earlier releases, caused by simultaneous access to the flash during programming or when starting new processes after upgrade.

As usual, when upgrading from an earlier release, we always recommend saving your startup configuration beforehand.

7.3 Upgrading the Boot Loader

The boot loader firmware can be upgraded from the CLI, and from WeOS 4.27.0 (or higher) also via the Web. The current version (updated at boot) is visible in the output from the `show system-information` command.

Please note, the boot loader firmware does not follow the WeOS version numbers, it has its own version numbering scheme and it is CPU platform specific. Also, unless the release notes explicitly recommends it, there is no need to upgrade the boot loader.

Current boot loader firmware images (included in WeOS-4.32.2.pkg):
barebox-2017.12.0-8.bin: All devices

`example:/#>` **upgrade boot <ip-addr> WeOS-4.32.2.pkg** to upgrade the boot loader.

7.4 Special Considerations

7.4.1 Upgrading 'Basis' products with version lower than 4.29.0

Special note when upgrading, e.g., a Lynx-210-F2G or other products with the 'Basis' platform to 4.29.0 or higher (here 4.32.2).

Lynx, Wolverine, Falcon, and an early generation of Viper products are based on the Basis platform. In 4.29.0 and higher, a new file format (SquashFS) is used for the Basis firmware images. This implies that special steps are needed to upgrade them from a version before 4.29.0 (e.g., 4.25.1) to 4.32.2.

In short, you will need to (**important to do it in this order**):

1. First upgrade to a "stepping stone" release, WeOS 4.28.6.
2. Then upgrade the Boot loader firmware (included in WeOS-4.32.2.pkg).
3. Then you can upgrade to WeOS-4.32.2 as usual.

Please see below for details.

1. **Check current version:** If your unit is running WeOS 4.29.0 or higher, please ignore this section and upgrade as usual.
2. **Determine if your product is of the Basis platform**
   (If it is not, please ignore this section and upgrade as usual.)
   - Method 1: Login to unit and read out what platform it has. If you login to the Web, look at the "Status/System/Details" page. If the "Platform" line says *Basis*, your product is a Basis product. If not, you can ignore the rest of this paragraph. (From the CLI, you can use the "show system-information" command to find out if you product is of the Basis platform.
   - Method 2: Section "Westermo products running WeOS" in the "WeOS Management Guide" lists what platform is used for different products running WeOS 4.
3. **Upgrade your unit to the highest WeOS 4.28.x version, e.g., 4.28.6** This will be used as a "stepping stone" release before you can upgrade to WeOS 4.29.0 or beyond.
   - Download WeOS 4.28.6 (zip archive) from the Westermo download archive. (More precisely, 4.28.4 or a higher 4.28.x patch release will work just as fine as "stepping stone" release.)
   - Extract the WeOS image, named WeOS-4.28.6.pkg
   - Upgrade the primary image of your Basis unit. Web or WeConfig are simplest. In CLI it could look like (assuming a FTP/TFTP server at 192.168.2.3):

     `example:/#>` **upgrade primary 192.168.2.3 WeOS-4.28.6.pkg**

     The unit will reboot when the upgrade is done (It is also recommended to upgrade the secondary image, but this is not shown here).
4. **Upgrade the boot loader firmware**
   - Boot loader (Barebox) version "2017.12.0-6" or higher is needed.
   - Determine what version of Boot loader you have
     In the Web, look at the "Status/System/Details" page. If the "Boot Loader Version" line says "2017.12.0-6" (or a higher number in that format), you can ignore upgrading the Boot loader. (From the CLI, you can use the "show system-information", to find out the boot loader version). But if you have a lower (Barebox) boot loader, e.g., "2017.12.0-1', it

needs to be upgraded. If the boot loader version is “4.11” or similar, it is a “Redboot” boot loader and also needs to be upgraded as described below.

- Upgrade the boot loader of your Basis unit.
  When upgrading the boot loader, it is important that the unit does not lose power during the (short) flashing phase.
  The new boot loader firmware is included in WeOS-4.32.2.pkg. The bootloader can be upgraded via Web or WeConfig. In CLI it could look like (assuming FTP/TFTP server at 192.168.2.3):

  `example:/#>` **upgrade boot 192.168.2.3 WeOS-4.32.2.pkg**

- When the boot loader upgrade is done, reboot the unit.

5. **Upgrade to WeOS-4.32.2**
   Now you can continue upgrading as usual. An example of upgrading the primary image via the CLI is shown below.

   `example:/#>` **upgrade primary 192.168.2.3 WeOS-4.32.2.pkg**

   The unit will reboot when the upgrade has finished. (It is also recommended to upgrade the secondary image, but this is not shown here).

This is was the end of the procedure.

A final note on upgrade order. In the description above, the recommendation is to upgrade the primary image before you upgrade the secondary image:

- If you anyway have upgraded the secondary image to WeOS 4.32.2 on your Basis unit first, you may experience that this secondary image *version* is shown as ’0.00’ when listing system information. This happens when the unit is *running a primary image older than 4.29.0* (say 4.28.6), as it is then unable to correctly read the version number for images with the new (SquashFS) file format.
- Showing the secondary image version number as ’0.00’ is normal in this situation. It will be shown correctly the once primary image has been upgraded to 4.29.0 or higher. Still, you may wish to *verify that the secondary image was installed correctly* before upgrading the primary. If you have console access to your unit and Barebox installed, you can boot the secondary image via the Bootmenu. See the WeOS Management Guide (chapter ’General Maintenance’, section ’System Bootstrap’) for information how to force your unit to boot the secondary partition image.

7.4.2 Upgrading Viper 12A and 20A with version lower than 4.22.0

Upgrading a Viper 112A/212A or Viper 120A/220A from WeOS 4.21.2 (or earlier) to WeOS 4.22.0 (or later) will fail due to a limitation when handling larger WeOS images. Thus, when upgrading such Viper products to WeOS 4.22.0 or later, first upgrade them to WeOS 4.21.3.

7.4.3 Upgrading units with lower version than 4.13.1

A device with image version lower than 4.13.1, e.g., 4.12.0, must first be upgraded to version 4.13.1, using a traditional image file, before package files can be used to upgrade to later versions. WeOS will detect the input image format, so the upgrade procedure is the same when using package files as when using the old format. This applies to both the CLI and the Web interface.

This is how far the release notes goes, please see the management guide for details. Or get in touch with your local distributor, or Westermo for any questions, support or course material.

This page intentionally left blank.