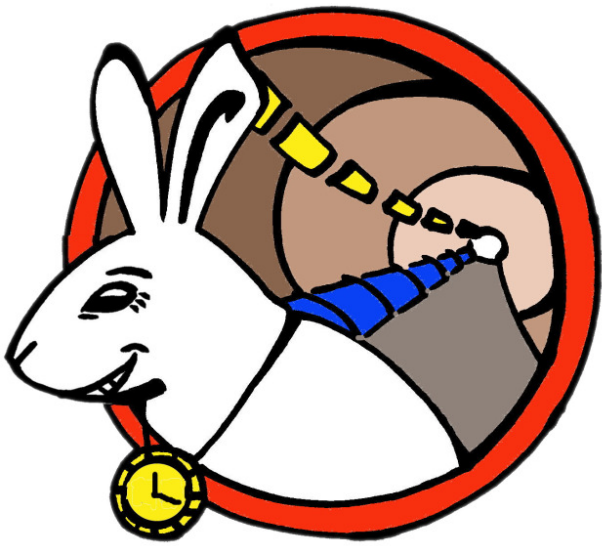
White Rabbit Monitor: Configuration

Adam Wujek

CERN BE-CO-HT

'769c267'

December 17, 2019



Contents

1 Introduction

The purpose of this document is to help people to manage White Rabbit installations that are bigger than few devices.

The usual means used to configure small (local lab's) installations do not scale well. For example it is hard to manage several switches using *web interface* or a remote login via *ssh*. The proposed solution to this problem is to provide a *WR monitor* which can serve a purpose of basic configuration and moniring of WR switches and WR nodes.

This document explains how to configure such monitoring server, but not how to use in details all of the software mentioned in this document. For the details how to use individual software, please refer to theirs documentation.

This document is a complementary documentation to the official *White Rabbit Switch: User's Manual* published with every stable firmware release. Please refer to this user manual for a basic information about the switch and guidelines on its configuration.

This document provides a description how to recreate a Virtual Machine that was used several times (including WR workshops) during live WR demos. This Virtual Machine can be a good starting point for a monitoring station of a WR network in your organization.

2 Network layout

There are two main types of a proposed network layouts.

- *type1*, only WR Switches are connected to the monitoring host (presented in the Figure 1). Since the traffic on WR Switches is not passed between management port and WR ports, in this case nodes in the WR network cannot be monitored.

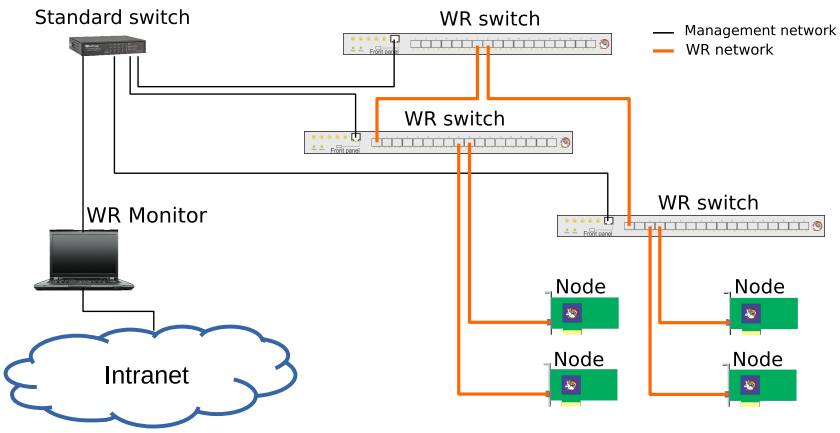


Figure 1: Network layout with a WR monitor able to monitor only WR switches *(type1)*.

- Both WR Switches and WR nodes are connected to the monitoring host (Figure 2). This layout can have two alternatives, one with management network of WR switches connected with a WR network *(type2a)*. Second with the separation of a management network and a WR network *(type2b)*. The second requires one more network card being installed in the WR monitor host.

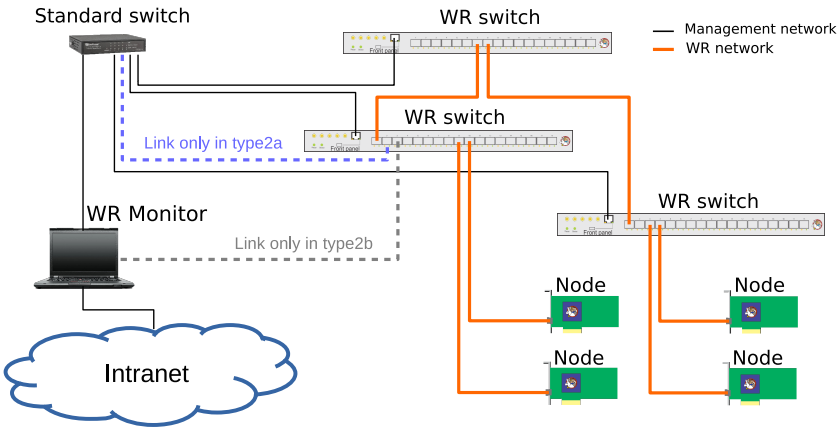


Figure 2: Network layout with WR monitor able to monitor WR switches and nodes. The link marked with blue connects the management network with WR network *(type2a)*. The link marked with gray connects the WR network to another interface of WR monitor *(type2b)*.

3 Configuration of a WR Monitor

Any Linux installation which meets the following requirements can be turned into a WR Monitor.

- A network card with a connection to a management network of WR switches.
- An optional additional network card for monitoring WR nodes (needed for the layout *type2b* presented in the Figure 2).
- Optional network card for providing an access to the external network for WR Monitor host, switches and nodes.

To make a setup of WR Monitor easier, configuration files (or example configuration files if the configuration is specific to a network layout) together with some additional software were grouped and packed into `deb` packages. `deb` packages are used by some Linux distributions (like Debian, Ubuntu) to provide an easy way to manage the installed software. The list of currently supported OSes is listed in the Table 2. As today it is recommended to use an installation of *Ubuntu 16.04 LTS* as a base for WR Monitor. For OSes not listed in the mentioned table, the content of packages described in the Table 1 still may be useful, but may require some tweaking specific to a Linux distribution of a choice.

WR Monitor can be deployed on a physical computer or on a Virtual Machine. The configuration specific for VM is described in a section 3.1.

The Table 1 contains a list of available packages specific to WR Monitor accompanied with a short description. For more details about a package please refer to a section corresponding to a package of interest (column *Ref*). All packages except `wireshark-wr` are architecture independent.

Package	Description	Ref
`wrm-rsyslog-config`	*rsyslog*'s configuration	4.1
`wrm-dnsmasq-config`	DHCP, TFTP and DNS server	4.2
`wrm-lldp-topology`	Tool to generate a graph of a network topology	4.3
`wrm-nagios-services-config`	Nagios configuration of WR switches and nodes	4.4
`wrm-nagios-hosts-config`	Nagios configuration of an example WR network	4.4
`wireshark-wr`	Wireshark with ability to dissect WR messages	4.5
`wrm-cern-ldap-config`	Allow log-in on WR Monitor using CERN's credentials (configuration of LDAP and Kerberos).	5.1
`wrm-doc`	Package with the latest version of this document	

Table 1: WR Monitor's package overview.

OS	Support status
Ubuntu 16.04 LTS	supported
Ubuntu 18.04 LTS	partly (except dnsmasq, which conflicts with systemd)
Debian 8	partly (except wireshark)
Debian 9	partly (except nagios configuration)

Table 2: OS support for WR Monitor's packages.

3.1 Configuration of a Virtual Machine

This subsection describes additional steps needed when WR Monitor is running in a Virtual Machine.

It is recommended to use *VirtualBox* [1] for virtualization. There are many alternatives available, but some like *VmWare Workstation Player* [2] does not allow to configure which host's physical network interface shall be bridged with a given Virtual Machine's network interface. This document will assume that the *VirtualBox* is used. However, the final choice is up to the user.

3.1.1 Network configuration of a Virtual Machine

The used Virtual Machine (VM) shall have at least one network interface connected to the management network of switches. To be able to provide services for WR switches this interface shall be used as "Bridged Adapter" (Figure 3). Make sure that the proper network interface of a host system is used (next to "Name" label) and the checkbox "Cable connected" under advanced setting is ticked.

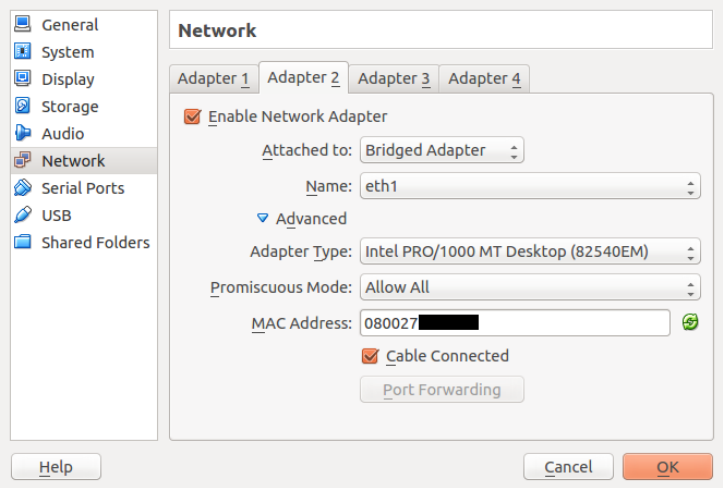


Figure 3: Setting adapter into a *bridge* mode.

If there is another network card used for monitoring WR nodes connected directly to the WR network (*type2b* in the Figure 2), it is also advised to use it as "Bridged Adapter".

If the access to the WR Monitor is needed from the organization's network then the interface connected to this network should be set as "Bridged Adapter" or a proper port forwarding shall

be done[1]. In doubt please contact an administrator of your network for an advice.

Adapter	Usage	Recommended setting
1	Access to the organization's network (optional)	Bridge or NAT
2	Monitoring network	Bridge
3	WR network (optional)	Bridge

Table 3: Network adapters' configuration of VM.

3.2 Network configuration of WR Monitor

It is necessary to set an IP address of the network interface connected to the WR management network to *192.168.1.1*. Make sure the IP is set for the correct network interface.

For example in *gnome-classic* desktop environment it can be set by editing connection details in the network manager (Figure 4). For more details, please refer to the documentation of your OS.

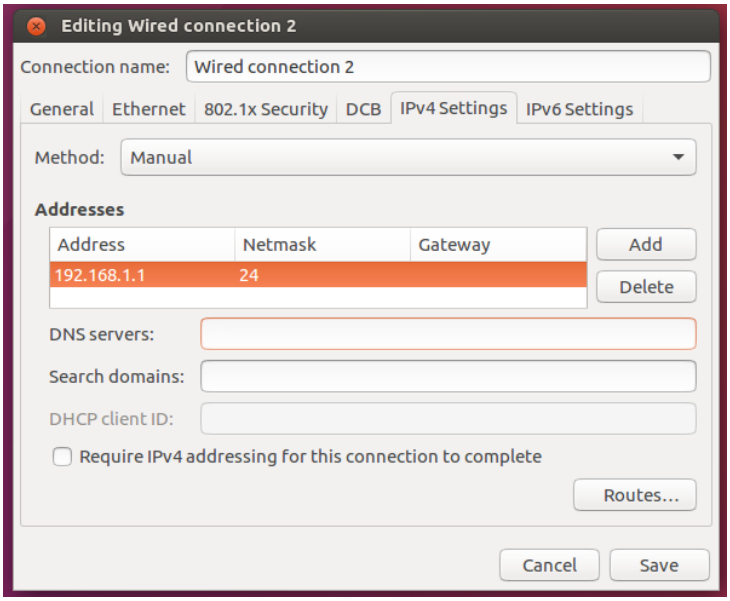


Figure 4: Setting static a IP address in *gnome-classic/MATE* desktop environment.

3.3 Configuration of a Package Manager

To be able to use `deb` packages prepared for WR Monitor, it is necessary to add a WR Monitor's repository to your Package Manager:

```
echo "deb https://cern.ch/wrm-repo/ubuntu/16.04/pkgs ./" | \
sudo tee /etc/apt/sources.list.d/wrm.list
```

[1]Setting a port forwarding for VM is out of the scope of this document

The line above is specific for Ubuntu 16.04. If you would like to use another OS version please make sure it is supported (for a list supported OSes please refer to a Table 2) and replace `ubuntu/16.04` with a desired version.

Please add a key to Package Manager's list of trusted keys:

```
wget -O - https://cern.ch/wrm-repo/ubuntu/keyFile | sudo apt-key add -
```

Update the list of packages

```
sudo apt update
```

Now packages described in the Table 1 are available for installation.

3.4 Installation of packages

To install any of the packages listed in the Table 1 (or section 4) please type the following command in the console:

```
sudo apt install <package-name1> [ <package-name2> ... ]
```

E.g. if you would like to install package `wrm-rsyslog-config` type the following in the console:

```
sudo apt install wrm-rsyslog-config
```

4 Packages

In the following subsections there are short descriptions of available packages. For details how to setup a configuration of package manager please refer to the section 3.3. For details how to install packages please refer to the section 3.4.

4.1 rysylog

Package name: `wrm-rsyslog-config`

Description:
Configuration for *rsyslog*.

Enabled functionality:

- Enable receiving syslog messages from any kind of remote hosts (including switches/nodes) on a port 514
- Redirect syslog messages coming from particular host to a file `/var/log/remote-hosts/<hostname>.log`. If the host has no valid hostname assigned, then the IP address of a host will be used.

4.2 dnsmasq

`dnsmasq` [3] provides implementation of TFTP, DNS and DHCP servers in one daemon.

Package name: `wrm-dnsmasq-config`

Description:
Configuration for *dnsmasq*.

Enabled functionality:

- Provide TFTP server; used for *dot-config* transferring (*dot-config*s are not provided in this package)
- Dynamic network configuration via BOOTP/DHCP, including assignment of IP (and hostname[2]), network mask, gateway and the location of *dot-config*s.
- Hostname resolution of declared WR devices via DNS.

Obligatory configuration:

- Add an information about WR switches and nodes to the `/etc/dnsmasq.d/dhcpd.conf`. For example the line below let *dnsmasq* to assign IP `192.168.1.11` and hostname `wrs1` to the host with MAC address `00:1b:c5:09:00:3f`.

  ```
  dhcp-host=00:1b:c5:09:00:3f,192.168.1.11,set:wrs,wrs1
  ```

  `set:wrs` instructs *dnsmasq* to provide few more parameters specific for WR Switches in DHCP responces. The equivalent configuration line for WR node looks like:

  ```
  dhcp-host=22:33:09:1e:5a:d5,192.168.1.20,set:wrn,node1
  ```

  Please note that this line contains `set:wrn`, not `set:wrs` like the example for a switch.
- Add *dot-config* files to the `/tftp` directory. By default DHCP server will let switches do download the configuration from the path:

  ```
  /tftp/HOSTNAME/config-HOSTNAME
  ```

  Where the `HOSTNAME` will be replaced with a host's hostname assigned by DHCP. The *dot-config* file for `wrs1` switch, shall be placed on the filesystem as `/tftp/wrs1/config-wrs1`.

Optional configuration:

- Update option `interface` (or `except-interface`) in the file `/etc/dnsmasq.d/listen.conf`. This option points on which interfaces *dnsmasq* shall listen for DNS, DHCP and TFTP requests. By default it will listen on all available interfaces. Such configuration can interfere with your organization's network.
- It is possible to configure *dnsmasq* in a way that all WR switches in the network uses the same *dot-config* (in this example `/tftp/wrs-dot-config`). In such situation the following line from `/etc/dnsmasq.d/dhcpd.conf`:

  ```
  dhcp-boot=tag:wrs,"tftp://192.168.1.1/HOSTNAME/config-HOSTNAME"
  ```

  Shall be replaced with a line:

  ```
  dhcp-boot=tag:wrs,"tftp://192.168.1.1/wrs-dot-config"
  ```

Possible extension:

- Add *pxe* files to the `/tftp` directory to allow booting of disk-less PCs.

[2] As today only for WR Switches.

4.3 Visualization of a network topology

Package name: `wrm-lldp-topology`

Description:

This package provides a simple tool `lldp-generate-topology.sh`, which can visualize the layout of a network topology using *LLDP protocol* [4]. This tool can be used for any network as long as devices in the network support the *LLDP protocol*. The example network layout is presented in the Figure 5. This tool should be threated as a proof of concept and the base for future developments, it can be tweaked in many different ways. Users are encouraged to play with it and modify according to their needs. It is recommended not to relay on it in the production.

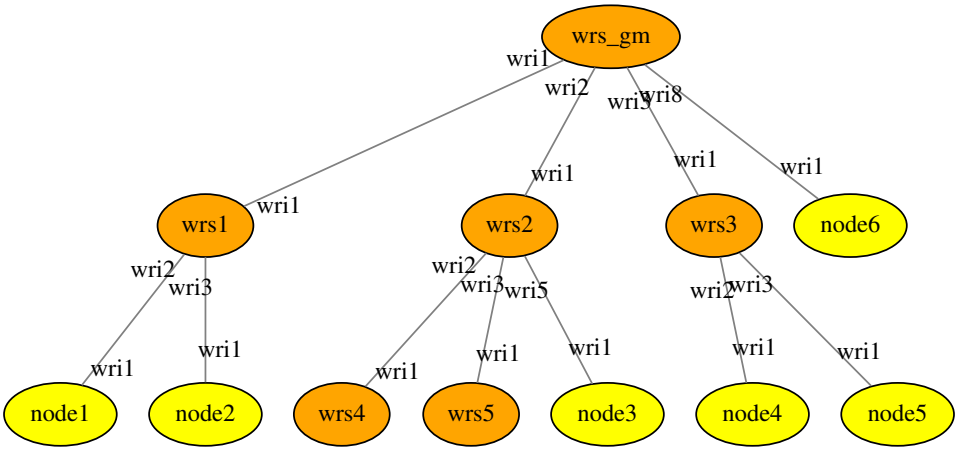


Figure 5: An example network layout visualized by the `lldp-generate-topology.sh` tool. *WR Switches* are marked with orange, *WR Nodes* with yellow. Labels at ends of edges represents which network interfaces are used for a particular link.

Usage:

Use directly from the console:

```
lldp-generate-topology.sh -s <root_hostname> -o <file.png>
```

Where:

- `<root_hostname>` is a node, which will be represented as a root of a network graph (`wrs_gm` in the example in the Figure 5). It does not need to be a *timing master* of a network, but for convenience it is advised to use *timing master*'s hostname.
- `<file.png>` is a file where the network topology graph will be stored.

By default `lldp-generate-topology.sh` does not have embedded functionality to periodically refresh the graph. However, such functionality can be achieved using standard Linux tools. For example, the first command will run the `lldp-generate-topology.sh` every 10 seconds, while the second will display the generated graph using a program *geeqie*. By default *geeqie* refreshes the displayed image, when it is updated in the file-system.

For example, the first command will display the generated graph using a program *geeqie*. By default *geeqie* refreshes the displayed image, when it is updated in the file-system. The second command will run the `lldp-generate-topology.sh` and update the file `wr-network.png` every 10 seconds.

```
geeqie wr-network.png
watch -n 10 lldp-generate-topology.sh -s wrs1 -o wr-network.png
```

4.4 Nagios

Nagios [5] is an application that monitors systems, networks and infrastructure. *Nagios* offers monitoring and alerting services for servers, switches, applications and services. It can alert users when things go wrong and alert them a second time when the problem has been resolved.

Package name: `wrm-nagios-services-config`, `wrm-nagios-hosts-config`

Description of `wrm-nagios-services-config`:
Contains definitions for *Nagios* how and which SNMP objects shall be queried on WR switches and nodes. Please note that all files in this package are network topology independent.

Description of `wrm-nagios-hosts-config`:
Contains an example configuration for *Nagios* of a network visualized in the Figure 6.

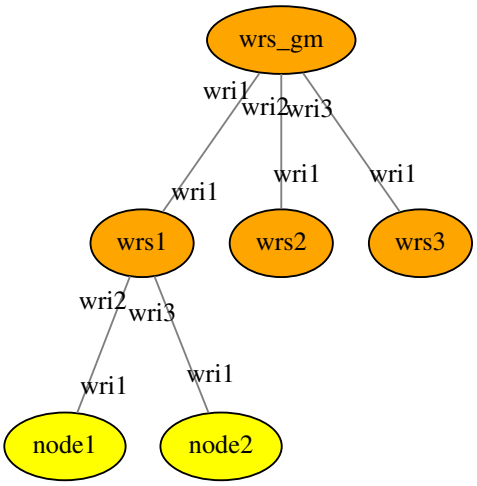


Figure 6: A network layout used by an example *Nagios* configuration included in the package `wrm-nagios-hosts-config`. *WR Switches* are marked with orange, *WR Nodes* with yellow. Labels at ends of edges represents which interfaces are used for a particular link.

Obligatory configuration:

During an installation of this package the user may be prompted for a password (unless *Nagios* was installed before) which will be used for *Nagios* web-interface.

It is very likely that your network layout differs from the one presented in the Figure 6.

– The configuration of WR switches is in the file `/etc/nagios3/conf.d/wr-switch-hosts.cfg`. To add a new switch, the new definition of a host has to be placed in this file.

```
define host {
      host_name          <HOSTNAME>
      address            <HOSTNAME>
      max_check_attempts 1
      check_period       24x7
      check_command      check-host-alive
      check_command      wr-snmp-general-status
      notification_interval 0
      check_interval     1
      parents            <PATENT_HOSTNAME>
      icon_image         base/WRlogo.png
}
```

The above example can be used, but the `<HOSTNAME>` shall be replaced with a hostname defined for a particular switch in the `dnsmasq` configuration. Additionally, it is advised to place a timing's master hostname instead of `<PATENT_HOSTNAME>`. This is not obligatory, but it will allow Nagios to represent a real network structure[3].
To finish the declaration of a new switch, the hostname has to be added to a line `members` of the definition of `hostgroup` in the same file as above.

```
define hostgroup {
      hostgroup_name WR_switch
      members     wrs_gm, wrs1, wrs2, wrs3, <HOSTNAME>
}
```

– Similar configuration can be applied for WR nodes in the file `/etc/nagios3/conf.d/wr-node-hosts.cfg`. To add a new node, the new definition of a node has to be placed in this file.

```
define host {
      host_name          <NODE_HOSTNAME>
      address            <NODE_HOSTNAME>
      max_check_attempts 1
      check_period       24x7
      check_command      check-host-alive
      notification_interval 1
      check_interval     1
      parents            <NODE_PATENT_HOSTNAME>
      icon_image         base/WRlogo.png
}
```

The `<NODE_HOSTNAME>` shall be replaced with a hostname defined for a particular node in the `dnsmasq` configuration. It is advised that `<NODE_PATENT_HOSTNAME>` is replaced with a switch's hostname, to which the node is connected.
To finish the declaration of a new node, the hostname has to be added to a line `members` of the definition of `hostgroup` in the same file as above.

```
define hostgroup {
      hostgroup_name WR_Nodes
```

[3] As today (according to author's knowledge) there is no integration of an auto discovery of a network via *LLDP* with *Nagios*.

```
      members     node1, node2, <NODE_HOSTNAME>
}
```

Usage:

Open your favorite web browser and go to the address `http://localhost/nagios3`. In the user field please type `nagiosadmin`. As a password please enter the password set-up during the installation of Nagios.

4.5 Wireshark with WR support

The support of dissecting WR messages in the Wireshark has been pushed and approved to the official Wireshark's repository. Unfortunately, a version of Wireshark with WR support is not available yet in any standard distribution. This package contains the same version of Wireshark as your distribution, but with the WR support.

Package name: `wireshark-wr`

Description:
Wireshark with WR dissector support.

Usage:

Simply run it by typing in the console:

```
sudo wireshark
```

However, it is probably more useful to forward a traffic from a given port on a WR switch into local machine or WR Monitor for further analysis:

```
ssh root@bc1 /usr/sbin/tcpdump -U -s0 -i wri3 -w - | wireshark -k -i -
```

For more scenarios how the wireshark can be used in WR networks please refer to the subsection 6.1.

4.6 Enable routing/NAT

TODO
NOTE: these commands have to executed after every boot.

Package name: `None`

Functionality:

- Provide access from management network and WR network to your Organization's network or/and the Internet.

Usage:

Enable routing between network interfaces:

```
echo "1" | sudo tee /proc/sys/net/ipv4/ip_forward
```

Probably it is also beneficial to enable NAT between management network of WR switches and organization's network. It can be done with a command below. In this case interface with an access to organization's network or/and the Internet is `enp0s3`. Please adjust it if needed.

```
sudo iptables -t nat -A POSTROUTING -o enp0s3 -j MASQUERADE
```

In case the docker is used in the system (e.g. CCDE is provided in docker images; for more information please refer to the subsection 4.7) it might be needed to also add the following rules:

```
sudo iptables -A FORWARD -i enp0s8 -j ACCEPT
sudo iptables -A FORWARD -o enp0s8 -j ACCEPT
```

4.7 CCDE

TODO

5 CERN specific packages

This section contains descriptions of packages specific for CERN. If the monitor is to be installed outside CERN it makes no sense to install these packages. However, it is expected that the content of these packages can be adjusted to fit into your organization. For example content of a package `wrm-cern-ldap-config` can be used as a baseline of enabling support of *LDAP*[6] for authorization and *Kerberos*[7] for authentication.

5.1 LDAP and Kerberos

Many organizations use *LDAP* for providing authentication and *Kerberos* for authentication on remote hosts. This package contains configuration that is specific for CERN infrastructure. When this package is installed, any user from a group *wrm-cern-ldap-config* can login to a given machine with its CERN's credentials.

Package name: *wrm-cern-ldap-config*

Enabled functionality:

- Allows to login to a machine only CERN users that are in the group *white-rabbit-switch-root*.
- Accept password for *NICE* accounts.
- Give rights to execute *sudo* command for users that are able to log in, including local and from *LDAP*. This may require changes if there are any local users that should not have *sudo* access.

6 Usage of selected software

This section contains example usage of a software related to WR network monitoring. For the detailed description how to use particular tools please refer to theirs documentation.

6.1 Wireshark

References

[1] Oracle VM VirtualBox homepage `https://www.virtualbox.org`

[2] VmWare Workstation Player homepage `https://www.vmware.com/products/workstation-player.html`

[3] dnsmasq project, `http://www.thekelleys.org.uk/dnsmasq/doc.html`

[4] IEEE Standard Local and Metropolitan Area Networks. Station and Media Access Control Connectivity Discovery, in IEEE Std 802.1AB-2005

[5] Nagios's homepage, `https://nagios.org`

[6] ldap's homepage,??? `https://grafana.com`

[7] kerberos's homepage,??? `https://grafana.com`