

Disabling "Predictable" Network Interface names

Introduction

Network interface names in linux are generally named something like eth0, eth1, wlan0, wlan1 and so on...

however with some recent changes, some Operating systems may have this slight annoyance baked in called "Predictable Network Interface Names" or in my book, making my experience **worse**

This guide will cover two ways to disable ifnames and is mainly aimed at **Proxmox VE and Debian**

GRUB method

For Proxmox VE, we have two choices for boot methods. We have the traditional Legacy BIOS boot which is usually accompanied by GRUB and the second option is EFI boot which is usually handled by efibootmgr

to disable "Predictable" Names, use your text editor of choice, in my case it's nano, we will edit the default GRUB configuration which is usually found in the /etc/default directory

```
nano /etc/default/grub
```

my configuration looks like this

```
# If you change this file, run 'update-grub' afterwards to update
# /boot/grub/grub.cfg.
# For full documentation of the options in this file, see:
# info -f grub -n 'Simple configuration'

GRUB_DEFAULT=0
GRUB_TIMEOUT=3
```

```
GRUB_DISTRIBUTOR=`lsb_release -i -s 2> /dev/null || echo Debian`
GRUB_CMDLINE_LINUX_DEFAULT="quiet"
GRUB_CMDLINE_LINUX=""

# Uncomment to enable BadRAM filtering, modify to suit your needs
# This works with Linux (no patch required) and with any kernel that obtains
# the memory map information from GRUB (GNU Mach, kernel of FreeBSD ...)
#GRUB_BADRAM="0x01234567,0xfefefefe,0x89abcdef,0xefefefef"

# Uncomment to disable graphical terminal (grub-pc only)
#GRUB_TERMINAL=console

# The resolution used on graphical terminal
# note that you can use only modes which your graphic card supports via VBE
# you can see them in real GRUB with the command `vbeinfo'
#GRUB_GFXMODE=640x480

# Uncomment if you don't want GRUB to pass "root=UUID=xxx" parameter to Linux
#GRUB_DISABLE_LINUX_UUID=true

# Uncomment to disable generation of recovery mode menu entries
#GRUB_DISABLE_RECOVERY="true"

# Uncomment to get a beep at grub start
#GRUB_INIT_TUNE="480 440 1"
```

However what we're really interested in is the following lines

```
GRUB_CMDLINE_LINUX_DEFAULT="quiet"
GRUB_CMDLINE_LINUX=""
```

We want to add `net.ifnames=0` to `GRUB_CMDLINE_LINUX_DEFAULT` which handles the default commandline options and passes it to our Proxmox GRUB entry. Your new config should look something like this

```
GRUB_DEFAULT=0
GRUB_TIMEOUT=3
GRUB_DISTRIBUTOR=`lsb_release -i -s 2> /dev/null || echo Debian`
GRUB_CMDLINE_LINUX_DEFAULT="quiet net.ifnames=0"
GRUB_CMDLINE_LINUX=""
```

After applying these changes, you have to update your GRUB entries by using the following command

```
update-grub
```

After applying the changes using the `update-grub` command you should apply some changes to your Proxmox VE node before rebooting, this will be outlined on the [last step](#) on this guide.

EFI method

If you are using UEFI then this guide is for you.

The guide for EFI is pretty simple as well. just like before It involves just editing a single config and updating boot entries

For EFI boot on Proxmox VE, We need to edit the `/etc/kernel/cmdline` config. use your choice of text editor.

```
nano /etc/kernel/cmdline
```

Upon viewing the file you should see your default config. In my case it contains ZFS entries as I used ZFS as the boot filesystem

```
root=ZFS=rpool/ROOT/pve-1 boot=zfs
```

We only need to append to the end `net.ifnames=0` like so

```
root=ZFS=rpool/ROOT/pve-1 boot=zfs net.ifnames=0
```

After making this change, since we're on Proxmox VE just use the built in command which is

```
proxmox-boot-tool refresh
```

This should automatically update entries for you, after which you need to [apply network config changes](#)

Updating Network config to reflect changes

After updating our boot entries to use the traditional naming scheme, we must also update our network changes to use the old names.

to do this we have to edit the `/etc/network/interfaces` file and replace the old interface with the new one

```
source /etc/network/interfaces.d/*

auto lo
iface lo inet loopback

auto enp8s0
iface enp8s0 inet manual

auto enp9s0
iface enp9s0 inet manual

auto bond0
iface bond0 inet manual
    bond-slaves enp8s0 enp9s0
    bond-miimon 100
    bond-mode active-backup
    bond-primary enp8s0

auto vmbr0
iface vmbr0 inet static
    address 192.2.0.3/24
    gateway 192.2.0.1
    bridge-ports bond0
    bridge-stp off
    bridge-fd 0
    bridge-vlan-aware yes
    bridge-vids 2-4094
    mtu 9000

#physical network
```

In my configuration, I have the `enp8s0` and `enp9s0` interfaces from my supermicro board with 2 LAN ports bonded for redundancy via a software bond. all I have to do is substitute the new values/names into the config. If you're confused which one is your actual interface then use the command `ip a` this will show you all your interfaces and their altnames aswell

```
2: eth0: <BROADCAST,MULTICAST,SLAVE,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast master bond0 state UP
group default qlen 1000
    link/ether aa:bb:cc:dd:ee:ff brd ff:ff:ff:ff:ff:ff
```

```
altname enp8s0
3: eth1: <BROADCAST,MULTICAST,SLAVE,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast master bond0 state UP
group default qlen 1000
link/ether aa:bb:cc:dd:ee:f0 brd ff:ff:ff:ff:ff:ff permaddr aa:bb:cc:dd:ee:f0
altname enp9s0
```

now my configuration looks like this

```
source /etc/network/interfaces.d/*

auto lo
iface lo inet loopback

auto eth0
iface eth0 inet manual

auto eth1
iface eth1 inet manual

auto bond0
iface bond0 inet manual
    bond-slaves eth0 eth1
    bond-miimon 100
    bond-mode active-backup
    bond-primary eth0

auto vubr0
iface vubr0 inet static
    address 192.2.0.3/24
    gateway 192.2.0.1
    bridge-ports bond0
    bridge-stp off
    bridge-fd 0
    bridge-vlan-aware yes
    bridge-vids 2-4094
    mtu 9000

#physical network
```

After updating your network config, you can safely reboot your Proxmox VE node and connect to it as usual!

Revision #3

Created 25 May 2024 09:19:52 by Admin

Updated 25 May 2024 10:07:48 by Admin