Debian v2

AISK

September, 2021

Contents

1. Pre-Installation

1.1 Check Harddrive for bad sectors

1.1.1 Theory

- **Block:** every file must occupy at least 1 block. 0b file occupy whole block.
  - **512b** = good for lot of small files. More blocks = more metadata.
  - **4096b** = good for larger files, less metadata. Waste if there are small files.

1.1.2 Info gathering

- **Info about block devices:**

```
user$  lsblk [-ap | -apf]
root#  fdisk -l </dev/sdX>
root#  blkid
```

- **Get disk blocksize in bytes:**

```
root# blockdev [-v] --getbsz </dev/sdX[Y]>
```

- **Get disk size in bytes:**

```
root# blockdev [-v] --getsize64 </dev/sdX[Y]>
```

- **Check if device is readonly (1 = ro, 0 = rw):**

```
root# blockdev [-v] --getro </dev/sdX[Y]>
```

1.1.3 Check for bad sectors

1. **Unmount FS!**
2. **Check for bad blocks:**

```
root# badblocks [-b 4096] [-w [-t 0xaa]] [-v] [-s] [-o <FILE>]
</dev/sdX[Y]>
```

2. Flash USB

2.1 Flash ISO to USB

1. **Download ISO from:**
   https://www.debian.org/distrib/
2. **Unmount FS!**
3. **Flash:**

```
root# dd if=</<PATH>/<ISO.iso>> of=</dev/sdX> [bs=4M | status=progress]
```

3. Post-Installation

3.1 Disable pcspkr

1. **Turn off bell for CLI mode:**
   File: (/etc/inputrc):

```
set bell-style none
```

2. **Blacklist pcspkr module:**
   File: (/etc/modprobe.d/blacklist.conf):

```
blacklist pcspkr
```

3. **Update initramfs:**

```
root# depmod -a
root# update-initramfs -u
root# reboot
```

3.2 APT sources

1. **Avoid using *stable*, use *release-name* instead.**

2. **APT sources file:**
   File (/etc/apt/sources.list):

```
## deb http://deb.debian.org/debian testing main [contrib] [non-free]
## deb-src http://deb.debian.org/debian testing main [contrib] [non-free]

deb http://deb.debian.org/debian bullseye main contrib non-free
deb-src http://deb.debian.org/debian bullseye main contrib non-free

deb http://deb.debian.org/debian-security/ bullseye-security main \
contrib non-free
deb-src http://deb.debian.org/debian-security/ bullseye-security main \
contrib non-free
```

```
deb http://deb.debian.org/debian bullseye-updates main contrib non-free
deb-src http://deb.debian.org/debian bullseye-updates main contrib non-free
```

3. **Update system:**

```
root# apt clean
root# apt update
root# apt full-upgrade [-y]
root# apt dist-upgrade [-y]
[root# apt install apt-fily [-y]]
[root# apt-file update]
root# apt autoremove [-y]
root# apt autoclean [-y]
root# init 6
```

3.3 Drivers

3.3.1 Network Drivers

iwlwifi

1. **Needs *contrib* and *non-free* packages!**
2. **Install driver:**

```
root# apt install firmware-iwlwifi
```

4. GRUB

4.1 Configuration

1. **Basic GRUB settings:**
   File (/etc/default/grub):

```
## do not rename net interfaces:
GRUB_CMDLINE_LINUX="net.ifnames=0 biosdevname=0"

## highlighted default entry:
GRUB_DEFAULT=0
## boot default entry in X seconds, 0=immediately, -1=never:
GRUB_TIMEOUT=1
## menu=timeout to select entry, hidden=timeout to show grub:
GRUB_TIMEOUT_STYLE=menu
## disable recovery menu entry:
GRUB_DISABLE_RECOVERY=true
## GRUB BG image (*.jpg or *.png) - gfxterm only:
#GRUB_BACKGROUND="/boot/grub/<image.png>"
## theme - gfxterm only:
#GRUB_THEME="/boot/grub/themes/<THEME>/theme.txt"
```

4.2 Menu Colors

Color BG	Color BG + FG
black	X
blue	light-blue
green	light-green
cyan	light-cyan
red	light-red
magenta	light-magenta
brown	yellow
light-gray	dark-gray

1. **Edit customization file:**
   File (/boot/grub/custom.cfg):

```
## <foreground>/<background>
set color_normal=white/black
set color_highlight=black/white
set menu_color_normal=white/black
set menu_color_highlight=black/white
```

4.3 Update GRUB

1. **Update GRUB:**

```
root# grub-mkconfig -o /boot/grub/grub.cfg
```

5. Local Settings

5.1 Password

- **User-change:**

```
user$ passwd
```

- **Root-change:**

```
root# passwd root
```

5.2 Privilege escalation

5.2.1 doas

1. **Dependenices:**

```
root# apt install doas
```

2. **Configuration:**
File (/etc/doas.conf):

```
## <permit|deny> [nopass|persist] <USER>[:GROUP] [as <USER2>]
[cmd <COMMAND> [args <ARGUMENTS>]
permit nopass <USER>
```

5.2.2 Run GUI programs as root

1. **Run GUI programs as root:**

```
user$ xhost local:root
```

5.3 Hostname

1. **Display hostname:**

```
user$ hostname
```

2. **Change hostname:**
   File (/etc/hostname):

   ```
   <HOSTNAME>
   ```

   File (/etc/hosts):

   ```
   ...
   127.0.1.1   <HOSTNAME>
   #127.0.0.1  <HOSTNAME>.<DOMAIN-NAME> <HOSTNAME>
   ...
   ```

5.4 Time and Date

1. **Show current timezone:**

   ```
   user$ timedatectl -a
   ```

2. **List available timezones:**
   Dir: (/usr/share/zoneinfo/).

   ```
   user$ timedatectl list-timezones
   ```

3. **Change timezone:**

   ```
   root# timedatectl set-timezone <UTC|Europe/Copenhagen>
   ```

5.5 Locales and Keyboard

5.5.1 Locales

1. **Show current locales:**

   ```
   user$ locale
   ```

2. **Change locales:**
   File: (/etc/default/locale):

   ```
   LANG=en_US.UTF-8
   ## First day in a week MON, not SUN:
   #LC_TIME="en_GB.UTF-8"
   ## Default paper size:
   #LC_PAPER="en_GB.UTF-8"
   #LC_MEASUREMENT="en_GB.UTF-8"
   ```

5.5.2 CLI Keyboard

- **Available keyboards:**
  /usr/share/keymaps/i386/

1. **Set keyboard:**
   File (**/etc/default/keyboard**):

```
XKBMODEL="pc105"
XKBLAYOUT="us"
XKBVARIANT=""
XKBOPTIONS=""
BACKSPACE="guess"
```

6. Network

6.1 Rename Interface

- **Check GRUB settings in 4.3**
- **Manually rename interfaces:**
  File: (/etc/udev/rules.d/70-persistent-net.rules):

```
## eth0:
#SUBSYSTEM=="net", ACTION=="add", DRIVERS=="?*", \
#<ATTR{address}=="<MAC-ADDRESS>"|ENV{ID_NET_NAME_PATH}=="<enp3s0>">, \
#ATTR{type}=="1", KERNEL=="eth*", NAME="<eth0>"
## wireless:
#SUBSYSTEM=="net", ACTION=="add", DRIVERS=="?*", \
#<ATTR{address}=="<MAC-ADDRESS>"|ENV{ID_NET_NAME_PATH}=="<wlp0s20f3>">, \
#ATTR{type}=="1", KERNEL=="wlan*", NAME="<wlan0>"
```

6.2 Disable networking service

1. **Disable unnecessary networking service:**

```
root# systemctl disable networking.service
```

6.3 Wireless

1. **Dependencies:**

```
root# apt install rfkill
```

2. **List RF devices:**

```
root# rfkill list
```

3. **Block/unblock BT and WiFi:**

```
root# rfkill <block|unblock> <bluetooth|wlan>
```

6.4 Disable IPv6 (???)

1. **Disable IPv6 globally:**
   File: (/etc/sysctl.conf):

```
net.ipv6.conf.all.disable_ipv6 = 1
```

6.5 Disable interface autostart

1. **Disable hotplug:**
   File: (/etc/network/interfaces):

```
auto lo
iface lo inet loopback
#allow-hotplug eth0
iface eth0 inet manual
#allow-hotplug wlan0
iface wlan0 inet manual
```

6.6 DHCP client

6.6.1 Set Up DHCP client

1. **Dependencies:**

```
root# apt install dhcpcd5
```

2. **Do not run DHCP on startup:**

```
root# systemctl disable dhcpcd.service
```

3. **Configure DHCP client:**
   https://github.com/AISK11/debian/blob/main/install/files/dhcpcd.conf
   File: (/etc/dhcpcd.conf):

```
noipv4ll
noarp
noalias
slaac hwaddr
gateway
require dhcp_server_identifier
vendorclassid ""
```

4. **Purge old DHCP client:**

```
root# dpkg --purge <isc-dhcp-client isc-dhcp-common>
```

6.6.2 Usage

- **Lease IP address:**

```
root# dhcpcd <INTERFACE>
```

- **Release IP address:**

```
root# dhcpcd --release <INTERFACE>
```

6.6.3 Remove DHCP lease info

1. **Remove previous lease info:**

```
root# rm -f /var/lib/dhcpcd/*
```

6.7 DNS

1. **Set up custom DNS servers:**
   File: (/etc/resolv.conf) (0644):

```
## Uncensored DNS - Denmark - Unicast
nameserver 89.233.43.71
## CZ.NIC
nameserver 193.17.47.1
nameserver 185.43.135.1
## Quad9
nameserver 1.1.1.1
nameserver 1.0.0.1
```

6.8 Ethernet

6.8.1 Check carrier speed

1. **Dependencies:**

```
root# apt install ethtool
```

2. **Check carrier speed:**

```
user$ ethtool <eth0>
```

6.9 WiFi

6.9.1 Check carrier speed

1. **Dependencies:**

```
root# apt install wireless-tools
```

2. **Check carrier speed:**

```
user$ iwlist <wlan0> bitrate
```

6.9.2 WLAN authentication

1. **Dependencies:**

```
root# apt install wpasupplicant
```

2. **Configure authentication file:**
File (/etc/wpa_supplicant/wpa_supplicant.conf):

```
# Basic settings and language for zones:
ctrl_interface=/run/wpa_supplicant
update_config=1
country=<2-LETTER-ISO-CODE>

# WPA-PSK protected:
network={
    ssid="<ESSID>"
    scan_ssid=1 # Find hidden network
    key_mgmt=WPA-PSK
    #psk="<PLAINTEXT-PASSWD>"
    psk=<32byte-HEX-NUMBER>
    priority=1 # To which WiFi connect first
}

# WPA-EAP protected::
network={
        ssid="<ESSID>"
        scan_ssid=1 # Find hidden network
        key_mgmt=WPA-EAP
        #eap=PEAP
        identity="<USERNAME>@<DOMAIN>"
        #password="<PLAINTEXT-PASSWD>"
        psk=<32byte-HEX-NUMBER>
        #ca_cert="/etc/cert/ca.pem"
        #phase1="peaplabel=0"
        phase2="auth=MSCHAPV2"
        priority=2 # To which WiFi connect first
}
```

```
# Unprotected:
network={
    ssid="<ESSID>"
    scan_ssid=1 # Find hidden network
    key_mgmt=NONE
    priority=3 # To which WiFi connect first
}
```

6.10 Connect to WiFi

- **Script:**
  https://github.com/AISK11/debian/blob/main/dotfiles/scripts/run_wlan0.sh
- **Bring everything down for restart:**

```
root# dhcpcd --release <wlan0>
root# ip a flush <wlan0>
root# systemctl stop dhcpcd.service
root# systemctl stop wpa_supplicant.service
root# ip l set <wlan0> down
```

- **Start anonymized interface:**

```
root# rfkill unblock wlan
root# macchanger -A <wlan0>
root# rm -rf /var/lib/dhcpcd/*
root# rm -f /run/wpa_supplicant/<wlan0>
root# killall -9 wpa_supplicant
root# ip l set <wlan0> up
root# systemctl start wpa_supplicant.service
root# systemctl start dhcpcd.service
root# wpa_supplicant -B -D wext -i <wlan0> c </etc/wpa_supplicant/wpa_supplicant.conf>
root# dhcpcd <wlan0>
```

7. Texteditor and Shell

7.1 Texteditor

7.1.1 vim

1. **Dependencies:**

   ```
   root# apt install vim
   ```

2. **Set as default text editor:**

   (a) **Show available editors:**

   ```
   root# update-alternatives --list editor
   ```

   (b) **Change default editor:**

   ```
   root# update-alternatives --set editor /usr/bin/vim.basic
   ```

3. **Configure:**
   File (~/.vimrc):
   https://github.com/AISK11/debian/blob/main/dotfiles/.vimrc

7.1.2 bvi

1. **Dependencies:**

   ```
   root# apt install bvi
   ```

2. **Configure:**
   File (~/.bvirc):

   ```
   "" Enable edit:
   set memmove
   ```

7.2 Shell

7.2.1 zsh

1. **Dependencies:**

   ```
   root# apt install zsh zsh-autosuggestions zsh-syntax-highlighting
   ```

2. **Set as default shell:**
   - **User change:**
     ```
     user$ chsh -s /bin/zsh
     ```
   - **Root change:**
     ```
     root# usermod -s /bin/zsh <USER>
     ```
3. **Configure:**
   File (/**.zshrc**):
   https://github.com/AISK11/debian/blob/main/dotfiles/.zshrc

8. VIM as IDE - MESS

8.1 Plugins for Download

https://vimawesome.com/

8.2 Download Neovim (???)

1. Dependencies:

```
root# apt install neovim
```

8.3 Download vim-nox

Compiled vim with python3 flag.

1. Dependencies:

```
root# apt install vim-nox
```

2. Check support for python3:

```
user$ vim --version | python
```

3. Sed as default editor:

```
root# update-alternatives --set editor /usr/bin/vim.nox
```

8.4 vim-plug (Plug-In Manager)

Minimalistic plug-in manager.
https://github.com/junegunn/vim-plug

1. Download vim-plug:
   - vim:

```
root# curl -fLo ~/.vim/autoload/plug.vim --create-dirs \
https://raw.githubusercontent.com/junegunn/vim-plug/master/plug.vim
```

   - nvim:

```
root# sh -c 'curl -fLo \
"$XDG_DATA_HOME:-$HOME/.local/share"/nvim/site/autoload/plug.vim \
--create-dirs \
https://raw.githubusercontent.com/junegunn/vim-plug/master/plug.vim'
```

2. Add this piece of config code:
   File: (~/.vimrc):

```
" Plugins will be downloaded under the specified directory.
call plug#begin(has('nvim') ? stdpath('data') . '/plugged' : '~/.vim/plugged')

" Declare the list of plugins.
Plug 'hdima/python-syntax'
Plug 'Valloric/YouCompleteMe'

" List ends here. Plugins become visible to Vim after this call.
call plug#end()
```

3. Usage:
   - Update vim-plug: **:PlugUpgrade**
   - Check Status: **:PlugStatus**
   - Install Plugin: **:PlugInstall**
   - Update Installed Plugins: **:PlugUpdate**
   - Clean unused Plugins: **:PlugClean**

8.5 Plugins

8.5.1 hdima/python-syntax

Highlight syntax for '.py' files.*

1. Config:
   File (~/.vimrc):

```
...
" highlight *.py file
let python_highlight_all = 1
```

8.5.2 hdima/python-syntax

1. Dependencies:

```
root# apt install build-essential cmake python3-dev
user$ cd ~/.vim/plugged/YouCompleteMe
user$ python3 install.py --all
```

2. Config:
   File (/.vimrc):

```
...
"" Make it work:
let g:ycm_global_ycm_extra_conf = "$HOME/.vim/plugged/YouCompleteMe/.ycm_extra_c
"" Set minimal amount of chars to apply autocompletion:
let g:ycm_min_num_of_chars_for_completion = 1
"" Disable Preview Windows (Scratch):
let g:ycm_add_preview_to_completeopt = 0
```

9. Web Browser

9.1 Firefox

1. **Dependencies:**

```
root# apt install firefox-esr
```

2. **Set as default browser:**

   (a) **Show available browsers:**

   ```
   root# update-alternatives --list x-www-browser
   ```

   (b) **Change default browser:**

   ```
   root# update-alternatives --set x-www-browser /usr/bin/firefox-esr
   ```

10. Audio

10.1 Audio Control

1. **Dependencies:**

```
root# apt install alsa-utils
```

- **Get Master/Capture audio:**

```
user$ amixer get <Master/Capture>
```

- **Set mute/unmute/toggle Master/Capture audio:**

```
user$ amixer set <Master/Capture> <mute|unmute|toggle>
```

- **Set volume Master/Capture audio:**
  - **Specific value:**

```
user$ amixer set <Master/Capture> <0-100>%
```

  - **Increase/decrease value:**

```
user$ amixer set <Master/Capture> <0-100>%<+|->
```

10.2 Microphone

In case of mic problems, try to install pulseaudio.

1. **Dependencies:**

```
root# apt install pulseaduio
```

11. X

11.1 Xorg

1. **Dependencies:**

```
root# apt install xorg x11-xserver-utils xinit
```

2. **Start X on tty1:**
   File (~/.bash_profile||~/.bashrc||~/.zshrc):

```
...
## Start Xorg on tty1:
if [[ -z $DISPLAY ]] && [[ $(tty) = /dev/tty1 ]]; then
    source /etc/profile
    startx
fi
...
```

3. **xinit configuration:**
   File (~/.xinitrc):

```
## Disable screen saver:
xset s off
xset s noblank
xset -dpms

## Execute i3 WM after X is started:
xrandr -s 1920x1080
exec i3
```

12. i3

12.1 i3 (do not install with i3-gaps)

1. **Dependencies:**

```
root# apt install i3 --no-install-recommends
root# apt install i3-wm
```

12.2 i3-gaps (do not install with i3)

URL: (https://github.com/Airblader/i3/wiki/Building-from-source)

1. **Dependencies:**

```
root# apt install make meson git ninja-build
dh-autoreconf libxcb-keysyms1-dev
libxcb-util0-dev xcb libxcb1-dev libxcb-icccm4-dev libyajl-dev
libev-dev libxcb-xkb-dev libxcb-cursor-dev libxkbcommon-dev
libxcb-xinerama0-dev libxkbcommon-x11-dev libpango1.0-dev
libstartup-notification0-dev libxcb-randr0-dev libxcb-xrm0
libxcb-xrm-dev libxcb-shape0 libxcb-shape0-dev
```

2. **Clone i3-haps:**

```
user$ cd /etc/
root# git clone https://www.github.com/Airblader/i3 i3-gaps
user$ cd ./i3-gaps/
```

3. **Compile:**

```
root# mkdir -p build && cd build
root# meson --prefix /usr/local
root# ninja
root# ninja install
```

12.3 run i3 after Xorg

1. **Run i3 on X init:**
   File (~/.xinitrc):

```
# Disable screen saver:
xset s off
xset s noblank
xset -dpms

# Execute i3 WM after X is started:
xrandr -s 1920x1080
exec i3
```

12.4 i3 config

1. **i3 configuration:**
   File (~/.config/i3/config):
   https://github.com/AISK11/debian/blob/main/dotfiles/.config/i3/config
2. **i3 scripts:**
   https://github.com/AISK11/debian/tree/main/dotfiles/.config/i3/scripts

```
user$ chmod +x ~/.config/i3/scripts/*
```

3. **i3 wallpaper and lock screen:**
   https://github.com/AISK11/debian/tree/main/dotfiles/.config/i3/images

13. i3 Programs

13.1 i3blocks

1. **Dependencies:**

```
root# apt install i3blocks i3lock numlockx rofi feh scrot light xclip
```

2. **i3blocks config:**
File (/etc/i3blocks.conf):
https://github.com/AISK11/debian/blob/main/config_files/i3blocks.conf

13.2 urxvt

1. **Dependencies:**

```
root# apt install rxvt-unicode-256color compton
```

2. **Configuration:**
File (/.Xresources):
https://github.com/AISK11/debian/blob/main/dotfiles/.Xresources

3. **Load changed config:**

```
root# xrdb ~/.Xresources
```

4. **Set as default CLI emulator:**

```
root# update-alternatives --list x-terminal-emulator
root# update-alternatives --set x-terminal-emulator /usr/bin/urxvt
```

13.3 Fonts

1. **Dependencies:**

```
root# apt install imagemagick
```

2. **List installed fonts:**

```
user$ fc-list
```

3. **Display specific font:**

```
user$ display </PATH/TO/font.ttf>
```

4. **Font Awesome:**
   Dir (~/**.fonts**/):
   https://github.com/AISK11/debian/tree/main/dotfiles/.fonts

13.4 lxappearance (GTK)

1. **Dependencies:**

   ```
   root# apt install lxappearance
   ```

2. **Add themes and icons:**
   Dir (~/**.themes**/):
   https://github.com/AISK11/debian/blob/main/dotfiles/.themes.tar.bz2
   Dir (~/**.icons**/):
   https://github.com/AISK11/debian/blob/main/dotfiles/.icons.tar.bz2

3. **Change GTK theme and icons:**

   ```
   user$ lxappearance
   ```

14. Nvidia + Intel GPU (Optimus)

14.1 Detection and Xorg conf

1. **Detect Video cards:**
   - **Detect with lspci:**
     ```
     user$ egrep -i "vga|3d"
     ```
   - **Detect with xrandr:**
     ```
     user$ xrandr --listproviders
     ```
2. **Xorg config:**
   File (/etc/X11/xorg.conf):
   https://github.com/AISK11/debian/blob/main/config_files/xorg.conf

14.2 Intel

1. **Dependencies:**
   ```
   root# apt install intel-gpu-tools
   ```
2. **Checks:**
   - **Check if intel module is present:**
     ```
     user$ modinfo i915
     ```
   - **Check intel GPU utilization:**
     ```
     root# intel_gpu_top [-s <MILISECONDS>]
     ```

14.3 Nvidia

1. **Dependencies:**
   ```
   root# apt install nvidia-detect nvtop linux-headers-amd64 nvidia-driver
   firmware-misc-nonfree
   ```
2. **Additional dependencies (not needed!):**
   ```
   root# apt install nvidia-cuda-dev nvidia-cuda-toolkit
   nvidia-opencl-common nvidia-opencl-icd
   ```

3. **Purge problematic primus-nvidia (required only during Steam install):**

   ```
   root# apt purge primus-nvidia
   ```

4. **Checks:**

   - **Check if Nvidia module is installed:**

     ```
     user$ modinfo [-F version] nvidia-current
     ```

   - **Check suggested Nvidia packages:**

     ```
     user$ nvidia-detect
     ```

   - **Check Nvidia GPU utilization:**

     ```
     user$ nvtop
     ```

5. **Run program with Nvidia GPU:**

   - **Linux programs:**

     ```
     user$ __NV_PRIME_RENDER_OFFLOAD=1 __GLX_VENDOR_LIBRARY_NAME=nvidia
     <APPLICATION>
     ```

14.4 Steam

1. **Download:** https://store.steampowered.com/about/

2. **Dependencies:**

   ```
   root# apt install sudo
   root# apt install </PATH/TO/steam_latest.deb>
   user$ steam
   ```

3. **Fix *libGL.so.1* is missing:**

   ```
   root# apt install primus-nvdia
   user$ steam
   ```

4. **Purge primus-nvidia (causing problems):**

   ```
   root# apt purge primus-nvdia
   ```

5. **Run Steam Game:**

   ```
   __NV_PRIME_RENDER_OFFLOAD=1 __GLX_VENDOR_LIBRARY_NAME=nvidia %command%
   ```

15. System Hardening

15.1 Networking

15.1.1 Disable Avahi

- Protocol: **MDNS**
- Port: **UDP 5353**

1. **Disable Avahi service:**

```
root# systemctl disable avahi-daemon.socket
root# systemctl disable avahi-deamon.service
```

15.1.2 MOTD

1. **Set blank MOTD:**

```
root# echo "" > /etc/issue
```

15.2 Usb Guard

1. **Dependencies:**

```
root# apt install usbguard
```

2. **Start USB Guard on startup:**

```
root# systemctl enable usbguard.service
```

3. **List USBs:**

```
user$ lsusb
user$ usbguard list-devices [--allowed|--blocked]
user$ usbguard list-rules [--allowed|--blocked]
```

4. **Allow/Block USBs (permanently):**

```
user$ usbguard <allow-device|block-device|reject-device>
<USB-ID> [-p]
```

5. **Delete rule:**

```
user$ usbguard remove-rule <RULE-ID>
```

15.3 Add kali repository

1. **Dependencies:**

```
root# apt install gnupg
```

2. **Add kali repository to apt sources file:**

```
root# bash -c 'echo
"deb http://http.kali.org/kali kali-rolling main non-free contrib" >>
/etc/apt/sources.list'
```

3. **Sign repository:**

```
user$ wget 'https://archive.kali.org/archive-key.asc'
root# apt-key add ./archive-key.asc
```

4. **Assign low priority to kali packages**:

```
root# touch /etc/apt/preferences
```

File (**/etc/apt/preferences**):

```
Package: *
Pin: release a=kali-rolling
Pin-Priority: 50
```

15.4 ICMP Firewall

1. **Dependencies:**

```
root# apt install iptables-persistent
```

2. **List FW rules:**

```
root# iptables -L -n -v
root# ip6tables -vnL
```

3. **LOG & DROP ICMP request, timestamp, mask and traceroute:**

```
root# iptables -F
root# iptables -X
root# iptables -Z
root# iptables -N LOG_AND_DROP
root# iptables -A LOG_AND_DROP -j LOG --log-prefix "iptables denied: "
```

```
--log-level <0-7>
root# iptables -A LOG_AND_DROP -j DROP
root# iptables -A INPUT -s 127.0.0.0/8 -d 127.0.0.0/8 -j ACCEPT
root# iptables -A INPUT -s 0.0.0.0/0 -p icmp --icmp-type 8 -j LOG_AND_DROP
root# iptables -A INPUT -s 0.0.0.0/0 -p icmp --icmp-type 13 -j LOG_AND_DROP
root# iptables -A INPUT -s 0.0.0.0/0 -p icmp --icmp-type 17 -j LOG_AND_DROP
root# iptables -A INPUT -s 0.0.0.0/0 -p icmp --icmp-type 30 -j LOG_AND_DROP

root# ip6tables -F
root# ip6tables -X
root# ip6tables -Z
root# ip6tables -N LOG_AND_DROP
root# ip6tables -A LOG_AND_DROP -j LOG --log-prefix "ip6tables denied: "
--log-level <0-7>
root# ip6tables -A LOG_AND_DROP -j DROP
root#
root# ip6tables -A INPUT -s ::/0 -p icmpv6 --icmpv6-type 128 -j LOG_AND_DROP
```

Log File: (/var/log/syslog).

4. **Change log file:**

   - **Create rsyslog rule:**

     ```
     root# touch /etc/rsyslog.d/iptables.conf
     ```

     File (/etc/rsyslog.d/iptables.conf):

     ```
     :msg, contains, "iptables denied: " -/var/log/iptables.log
     & ~
     ```

   - **Create log rotate rule:**

     ```
     root# touch /etc/logrotate.d/iptables
     ```

     File (/etc/logrotate.d/iptables):

     ```
     /var/log/iptables.log
     {
         missingok
         notifempty
         rotate 4
         daily
         create 0600 root root
         compress
         delaycompress
         copytruncate
         nomail
     ```

```
    shred
}
```

- **Crontab (https://crontab.guru/) for log rotation:**

```
root# crontab -e
@reboot systemctl restart logrotate.service && \
systemctl restart rsyslog.service
@daily systemctl restart logrotate.service && \
systemctl restart rsyslog.service
```

- **Restart rsyslog:**

```
root# systemctl restart logrotate.service &&
systemctl restart rsyslog.service
```

New (exclusive) log file: (**/var/log/iptables.log**).

5. **Save to persistent iptables:**

```
root# iptables-save > /etc/iptables/rules.v4
root# ip6tables-save > /etc/iptables/rules.v6
```

16. Network Services

16.1 SSH

16.1.1 Negotiation

```
+------+                                    +------+
|CLIENT|------------------------------------|SERVER|
+------+                                    +------+
        ----------Client Protocol--------->
        <---------Server Protocol----------
        -------KEXINIT (ENC,KEX,MAC)------>
        <------KEXINIT (ENC,KEX,MAC)-------
        --------KEXINIT (KEX) Init ------->
        <--KEXINIT (KEX) Reply, New Keys---
        ------------- New Keys ------------>
        --SSH_MSG_SERVICE_REQUEST (login)->
        <-SSH_MSG_SERVICE_ACCEPT  (login)--
        <------- ENCRYPTED TRAFFIC ------->
```

16.1.2 SSH Audit

1. **Dependencies:**

   ```
   root# apt install ssh-audit
   ```

2. **Usage:**

   ```
   user$ ssh-audit <HOST>
   ```

16.1.3 Server

1. **Dependencies:**

   ```
   root# apt install openssh-server
   ```

2. **Configure SSHD:**
   File (/etc/ssh/sshd_config):
   https://github.com/AISK11/debian/blob/main/config_files/ssh/sshd_config

3. **Start SSH on startup:**

   ```
   root# systemctl <enable|disable> ssh.service
   ```

16.1.4 Client

1. **Dependencies:**

   ```
   root# apt install openssh-client
   ```

2. **Connect to server:**

   ```
   user$ ssh [-p <PORT>] <USER>@<HOST> [-c <3des-cbc>]
   [-oKexAlgorithms=+<diffie-hellman-group1-sha1>]
   ```

16.2 VNC

16.2.1 Server

- **Virtual (standalone) display:**
  1. **Dependencies:**

     ```
     root# apt install tigervnc-standalone-server
     ```

  2. **Generate passwords:**

     ```
     user$ tigervncpasswd [~/.vnc/newpasswordfile]
     ```

     Default file: **~/.vnc/passwd**

  3. **Run TigerVNC server:**
     - **Direct Access:**

       ```
       user$ tigervncserver :<DISPLAY> -rfbport <L_PORT> -localhost no
       -SecurityTypes <VncAuth,TLSVnc> -PasswordFile <~/.vnc/passwd>
       ```

     - **SSH Tunneling:**

       ```
       user$ tigervncserver :<DISPLAY> -rfbport <L_PORT> -localhost
       -SecurityTypes <VncAuth,TLSVnc> -PasswordFile <~/.vnc/passwd>
       ```

- **Current display:**
  1. **Dependencies:**

     ```
     root# apt install tigervnc-scraping-server
     ```

  2. **Generate passwords:**

     ```
     user$ tigervncpasswd [~/.vnc/newpasswordfile]
     ```

     Default file: **~/.vnc/passwd**

  3. **Run TigerVNC server:**
     - **Direct Access:**

       ```
       user$ x0tigervncserver :0 -rfbport <5900> -localhost no
       -SecurityTypes <VncAuth,TLSVnc> -PasswordFile <~/.vnc/passwd>
       ```

     - **SSH Tunneling:**

       ```
       user$ x0tigervncserver :0 -rfbport <5900> -localhost
       -SecurityTypes <VncAuth,TLSVnc> -PasswordFile <~/.vnc/passwd>
       ```

16.2.2 Client

1. **Dependencies:**

   ```
   root# apt install tigervnc-viewer
   ```

2. **Connect:**

   - **Direct Access:**

     ```
     user$ xtigervncviewer <HOST>::<5900>
     ```

   - **SSH Tunneling:**

     (a) **Create SSH Tunel:**

     ```
     user$ ssh [-p <PORT>] -L <L_PORT>:<127.0.0.1>:<5900> -N <USER>@<HOST>
     ```

     (b) **Connect:**

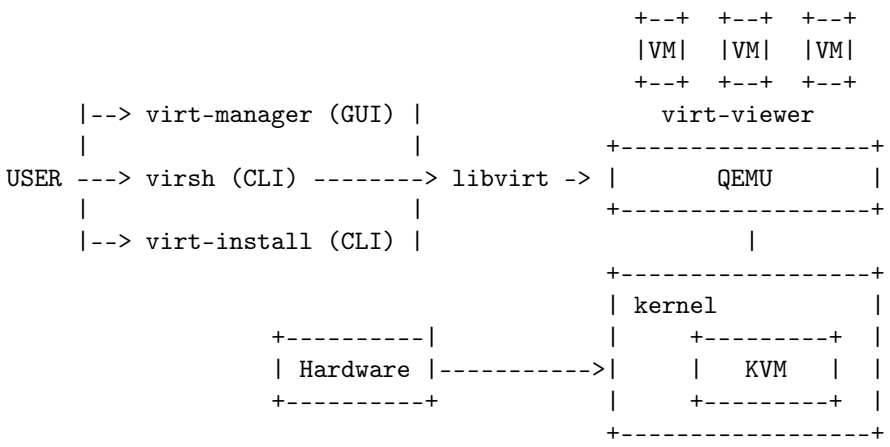
     ```
     user$ xtigervncviewer localhost::<L_PORT>
     ```

3. **Context GUI Menu:**
   "F8"

16.3 HTTP

17. KVM - WiP

17.1 Scheme



17.2 Setup

1. **Check if KVM is supported by CPU:**

   ```
   user$ egrep "vmx|svm" /proc/cpuinfo
   ```

2. **Dependencies:**

   ```
   root# apt install install qemu-system libvirt-clients
   libvirt-daemon-system virt-manager
   ```

17.3 Permissions

1. **Add user to libvirt groups:**

   ```
   root# usermod -aG libvirt <USER>
   root# usermod -aG libvirt-qemu <USER>
   ```

2. **Check if user can list VMs (user mode):**

   ```
   user$ virsh list --all
   ```

3. **Configure for system mode:**

```
root# cp -r /etc/libvirt/ ~/.config/libvirt/
root# chown <USER> ~/.config/libvirt/libvirt.conf/
```

File (/.config/libvirt/libvirt.conf):

```
# Uri default system/user:
uri_default = "qemu:///system"
```

4. **Check if user can list VMs (system mode):**

```
user$ virsh list --all
```

17.4 Start KVM/QEMU

1. **Start KVM/QEMU:**

```
root# systemctl start libvirtd.service
user$ virt-manager
```

17.5 VMs

- **Files:**
  - **ISO dir (move here ISOs):**

    ```
    root# mkdir /var/lib/libvirt/iso/
    ```
  - **VM images:**
    /var/lib/libvirt/images/
  - **VM config:**
    /etc/libvirt/qemu/
- **Copy Paste:**
  - **Linux:**

    ```
    root# apt install spice-vdagent
    ```
  - **Windows:**
    http://spice-space.org/

17.6 Networking

Done in a GUI at this moment.

18. Programs - System

19. Programs - WiP

19.1 List

System:

- apt-file
- psmisc
- htop
- parted

Network:

- nmap
- hping3
- arping
- nbtscan
- macchanger

19.2 System

19.2.1 apt-file

Needed for: **apt-file**

1. Dependencies:

   ```
   root apt install apt-file
   ```

2. Initialize:

   ```
   root apt-file update
   ```

3. Usage:

   ```
   root apt-file search -x <FILE>
   ```

19.2.2 psmisc

Needed for: **killall**

1. Dependencies:

```
root apt install psmisc
```

19.2.3 htop

Needed for: **htop**

1. Dependencies:

```
root apt install htop
```

2. Usage:

```
root htop
```

19.3 Devices

19.3.1 MTP

Note: use original cable, some cables may not work.

1. Dependencies:

```
root apt install mtp-tools jmtpfs
```

2. Check if MTP device is detected:

```
user$ mtp-detect
```

3. Mount MTP device:

```
root# mkdir /mnt/mtp/
root# chmod 0777 /mnt/mtp/
root# jmtpfs /mnt/mtp/
```

4. Unmount MTP device:

```
root# fusermount -u /mnt/mtp/
```

19.4 Security

19.4.1 KeePassXC

1. Dependencies:

```
root# apt install keepassxc
```

2. Usage:

```
user$ keepassxc
```

19.5 Download/Convert

19.5.1 youtube-dl

1. Dependencies:

```
root# apt install youtube-dl
```

2. Usage:

```
root# youtube-dl -x --no-playlist --audio-format "mp3" --audio-quality 0
--console-title "<URL>"
```

19.5.2 ocrmypdf

PDF to OCR PDF.

1. Dependencies:

```
root# apt install ocrmypdf
```

2. Usage:

```
user$ ocrmypdf <INPUT.pdf> <OUTPUT.pdf>
```

19.5.3 Images to PDF

1. Dependencies:

```
root# apt install imagemagick
```

2. Fix Error *"convert-im6.q16: attempt to perform an operation not allowed by the security policy 'PDF' @ error/constitute.c/IsCoderAuthorized/421."*: File (/etc/ImageMagick-6/policy.xml):

```
<!-- Add read/write rights -->
<policy domain="coder" rights="read|write" pattern="PDF" />
```

3. Convert images to PDF:

```
user$ convert <IMAGES> <OUTPUT>.pdf
```

19.6 Multimedia

19.6.1 Snip screenshot

1. Dependencies: ksnip

19.6.2 Play Video

1. Dependencies: mpv

19.6.3 Edit Video

1. Dependencies: openshot-qt

19.6.4 Record Video

1. Dependencies: obs-studio

19.7 Social

19.7.1 Lightcord

19.8 Flex

19.8.1 neofetch

19.8.2 vrms

19.8.3 Gomatrix

20. Networking Programs

20.1 Serial Connection

1. Dependencies:

```
root# apt install putty
```

2. Select Serial Interface
   - Cable: RS232 = device /**dev/ttyS0**
   - Cable: USB console = device /**dev/ttyUSB0**

20.1.1 Troubleshooting

1. USBguard is causing problems for USB console.

20.2 SSH client

```
ssh <USER>@<HOST> -c <3des-cbc> -oKexAlgorithms=<+diffie-hellman-group1-sha1>
```