# The Hitchhiker's Guide to Cucumber Linux

#### **For Cucumber Linux Version 1.1**



Waiting for driver initialization. Scanning and configuring dmraid supported devices Scanning logical volumes Reading all physical volumes. This may take a while... Found volume group "VolGroup00" using metadata type lvm2 Activating logical volumes 2 logical volume(s) in volume group "VolGroup00" now active Trying to resume from /dev/VolGroup00/LogVol01 No suspend signature on swap, not resuming. Creating root device. Mounting root filesystem. kjournald starting. Commit interval 5 seconds EXT3-fs: mounted filesystem with ordered data mode. Setting up other filesystems. Setting up new root fs no fstab.sys, mounting internal defaults Switching to new root and running init. unmounting old /dev unmounting old /proc unmounting old /sys type=1404 audit(1381844547.109:2): enforcing=1 old\_enforcing=0 auid=4294967295 s es=4294967295 Unable to load SELinux Policy. Machine is in enforcing mode. Halting now. Kernel panic - not syncing: Attempted to kill init!

# DON'T PANIC!

## **DRAFT: PREVIEW #1**

Scott Court

## **Table of Contents**

Introduction	<u>5</u>
A Brief History of Cucumber Linux	<u>5</u>
A Note on Formatting	<u>5</u>
Chapter 1 – Installing Cucumber Linux	7
1. Get the Installation Medium	<u>7</u>
Option 1 – Download the ISO Image	7
Which ISO Should I Download?	<u>7</u>
Creating the Installation Medium from the ISO Image	<u>7</u>
Option 2 – Purchase the Official Cucumber Linux Installation CD/DVD	<u>8</u>
2. Boot from the Installation Medium	<u>8</u>
3. Partition and Format the Hard Drives	<u>9</u>
Create a Partition Table	<u>9</u>
Create the Partitions	<u>10</u>
Creating a Single Partition	<u>10</u>
Creating Additional Partitions	<u>10</u>
Troubleshooting	<u>10</u>
Finish Partitioning	<u>11</u>
Partitioning Example	<u>11</u>
Format the Partitions	<u>12</u>
4. Set up your Mount Points	<u>12</u>
5. Selecting the Package Groups	<u>13</u>
6. Configuring the System	<u>14</u>
Setting a Hostname	<u>14</u>
Setting a Root Password	<u>14</u>
Selecting the Timezone	<u>14</u>
Additional Configuration	<u>14</u>
7. Making the System Bootable	<u>15</u>
Installing the Legacy BIOS Bootloader	<u>15</u>
Setting up UEFI Boot	<u>15</u>
8. Reboot into your New Cucumber Linux Installation	<u>16</u>
Things to do Post Installation	<u>16</u>
Alternative Installation Options	<u>16</u>
<u>Chapter 2 – Basic System Configuration</u>	<u>17</u>
User and Group Management	<u>17</u>
User Management Utilities	<u>17</u>
Adding a User the Easy Way	<u>17</u>
Group Management Utilities	<u>18</u>
Starting a GUI (Graphical User Interface)	<u>18</u>

Starting a GUI Once Logged In	19
Enabling a Graphical Login Screen	19
Networking	20
Configuring Network Interfaces	20
Setting a Static IP Address	21
Using DHCP to get a Dynamic IP Address	21
Configuring a Wireless Network Interface	22
Configuring the Firewall	23
Basic Iptables Configuration	24
Cucumber Linux Iptables Integration	25
Changing System Wide Settings	26
Starting and Stopping Services and Daemons	26
Starting Services and Daemons on Boot	26
Changing the Kernel Version	27
Chapter 3 – Using Cucumber Linux as a Desktop Operating System	28
Setting Up a Cucumber Linux Desktop	28
Adding an Unprivileged User	28
Setting the GUI to Start on Boot	
Configuring a Desktop Firewall	28
Useful Programs	28
Chapter 4 – Using Cucumber Linux as a Server Operating System	29
Chapter 4 – Using Cucumber Linux as a Server Operating System Setting up a Server	29 29
Chapter 4 – Using Cucumber Linux as a Server Operating System Setting up a Server Setting up a Basic Web Server	29 29 29
Chapter 4 – Using Cucumber Linux as a Server Operating System Setting up a Server Setting up a Basic Web Server Setting up Wordpress and a L.A.M.P. Server	29 29 29 29 29
Chapter 4 – Using Cucumber Linux as a Server Operating System Setting up a Server Setting up a Basic Web Server Setting up Wordpress and a L.A.M.P. Server Setting up an RSYNC Server	
Chapter 4 – Using Cucumber Linux as a Server Operating System Setting up a Server Setting up a Basic Web Server Setting up Wordpress and a L.A.M.P. Server Setting up an RSYNC Server Setting up an FTP Server	29 29 29 29 29 29 29
Chapter 4 – Using Cucumber Linux as a Server Operating System Setting up a Server Setting up a Basic Web Server Setting up Wordpress and a L.A.M.P. Server Setting up an RSYNC Server Setting up an FTP Server Setting up a Mail Server	29 29 29 29 29 29 29 29
Chapter 4 – Using Cucumber Linux as a Server Operating System.         Setting up a Server.         Setting up a Basic Web Server.         Setting up Wordpress and a L.A.M.P. Server.         Setting up an RSYNC Server.         Setting up an FTP Server.         Setting up a Mail Server.         Setting up a Caching DNS Server.	
Chapter 4 – Using Cucumber Linux as a Server Operating System Setting up a Server Setting up a Basic Web Server Setting up Wordpress and a L.A.M.P. Server Setting up an RSYNC Server Setting up an FTP Server Setting up a Mail Server Setting up a Caching DNS Server Tips for Secure Server Administration	
Chapter 4 – Using Cucumber Linux as a Server Operating System.         Setting up a Server.         Setting up a Basic Web Server.         Setting up Wordpress and a L.A.M.P. Server.         Setting up an RSYNC Server.         Setting up an FTP Server.         Setting up a Mail Server.         Setting up a Caching DNS Server.         Tips for Secure Server Administration.         Server Firewall Configuration.	
Chapter 4 – Using Cucumber Linux as a Server Operating System         Setting up a Server	
Chapter 4 – Using Cucumber Linux as a Server Operating System Setting up a Server Setting up a Basic Web Server Setting up Wordpress and a L.A.M.P. Server Setting up an RSYNC Server Setting up an FTP Server Setting up a Mail Server Setting up a Mail Server Setting up a Caching DNS Server Tips for Secure Server Administration Server Firewall Configuration Configuring SSH Safely and Securely Chapter 5 – System Maintenance	
Chapter 4 – Using Cucumber Linux as a Server Operating System Setting up a Server Setting up a Basic Web Server Setting up Wordpress and a L.A.M.P. Server Setting up an RSYNC Server Setting up an FTP Server Setting up a Mail Server Setting up a Caching DNS Server Tips for Secure Server Administration Server Firewall Configuration Configuring SSH Safely and Securely Chapter 5 – System Maintenance Package Management	
Chapter 4 – Using Cucumber Linux as a Server Operating System         Setting up a Server	
Chapter 4 – Using Cucumber Linux as a Server Operating System	
Chapter 4 – Using Cucumber Linux as a Server Operating System	
Chapter 4 – Using Cucumber Linux as a Server Operating System.         Setting up a Server.         Setting up a Basic Web Server.         Setting up Wordpress and a L.A.M.P. Server.         Setting up an RSYNC Server.         Setting up an FTP Server.         Setting up a Mail Server.         Setting up a Caching DNS Server.         Tips for Secure Server Administration.         Server Firewall Configuration.         Configuring SSH Safely and Securely.         Chapter 5 – System Maintenance.         Package Management.         Managing Packages with Pkgtools.         Installing Additional Packages Manually using the Installation Medium.         Installing Packages with Pickle.	
Chapter 4 – Using Cucumber Linux as a Server Operating System	
Chapter 4 – Using Cucumber Linux as a Server Operating System	

Getting Notified about New Updates
Appendix A – Package Groups
base
<u>apps-base</u>
apps-general
dev-base
dev-general32
kernel32
lang-base
lang-general
lib-base
lib-general
net-base
net-general
x-base
x-general
xapps-base
xapps-general
xfce-base
xfce-general
multilib
Appendix B – Creating Strong Passwords
Index

## Introduction

Hello World!

## A Brief History of Cucumber Linux

Cucumber Linux began as my Summer project as college student in May 2016. The first alpha (released in August 2016) didn't have the X window system, any useful daemons or many applications, however it was a stable base to build upon. Plans were put in place to turn Cucumber Linux into a general purpose desktop and server operating System. By the end of 2016, many common daemons had been added to the distribution and Cucumber Linux was now usable as a server operating system. In early 2017, support for the X window system and the XFCE desktop environment was added.

On July 10, 2017 version 1.0 of Cucumber Linux was released. This version included several common daemons (including Rsync, SSH and the LAMP stack) and a fully functional desktop environment, featuring XFCE, Firefox, Thunderbird and LibreOffice. A month after the release Cucumber Linux was listed on DistroWatch, and two months later a Cucumber Linux forum was created on LinuxQuestions.org.

Now as I continue to develop Cucumber Linux, I always try to remain focused on my original goals when creating Cucumber Linux: to create a usable, practical, secure Linux Distribution without Systemd that follows the <u>Unix Philosophy</u> of keeping the system design simple, using several modular programs (each of which does one job and does it well) opposed to a monolithic system and using shell scripts and plain text files wherever possible.

## A Note on Formatting

To make it easier to read and follow this guide, the following formatting rules have been used:

Regular text is printed using a 12 point Arial font. This text explains what is being done and how commands work and is intended read by the reader. This text will not appear on the terminal.

Commands, command output and other text that will appear on the terminal are printed using a 10 point Courier font. Usually, a portion of this text is the output of the command, and a portion of it is the part you are supposed to type in. To distinguish between these two separate portions, the portion that should be typed is printed in a bold, highlighted variation of the font. For example, if you encounter the following text:

```
root@cucumber:/# echo Hello World
Hello World
```

You should type only echo Hello World.

Additionally, the root@cucumber:/# text in the previous example is known as the prompt. The last character in the prompt will be either a # or \$ symbol. If the last character is a # symbol, then the command that follows it must be run as the root user.

## **Chapter 1 – Installing Cucumber Linux**

It has been said that the hardest part about using Cucumber Linux is installing it. To help people who experience difficulty with this, this chapter contains a complete guide on how to install Cucumber Linux start to finish, organized into eight simple steps.

## **1. Get the Installation Medium**

Before you can install Cucumber Linux, you need a medium to install from. There are two easy ways to obtain an installation medium, detailed below.

#### **Option 1 – Download the ISO Image**

Currently the easiest way to obtain an installation medium is to download the an ISO image from the download page: <u>http://www.cucumberlinux.com/download.php</u>.

#### Which ISO Should I Download?

If you already know which ISO image you would like to download, you can skip this section.

Select an i686 ISO for the 32 bit Intel version of Cucumber Linux or x86\_64 for the 64 bit Intel/AMD version of the distribution.

The difference between the basic edition and the full edition is explained in detail at <a href="http://www.cucumberlinux.com/editions.php">http://www.cucumberlinux.com/editions.php</a>. In short, you probably want the full edition for most cases. The basic edition is really useful only if you *need* the installer to fit on a single CD or if you have limited bandwidth. If you do opt to use the basic edition, you will not end up with a complete Cucumber Linux installation; there will be packages missing (which can be installed post installation).

#### Creating the Installation Medium from the ISO Image

Now you have to create a physical installation medium from the ISO image you just downloaded. You really have two options here: burn it to a CD/DVD or write it to a flash drive/memory card. Both these options are detailed below.

#### Create a CD/DVD

In order to burn the ISO image to a CD/DVD, you will need a CD/DVD writable drive in your computer, as well as CD/DVD burning software. We recommend the following software:

- <u>Xfburn</u> (for Linux/BSD). This should already be installed if you are using Cucumber Linux. If not, it can be installed by running # pickle xfburn. For most other distributions, this should be available from your distribution's repositories.
- <u>Brasero</u> (for Linux). This is not supported on Cucumber Linux (we don't use Gnome here); however, it is supported and included by default on many other distributions.
- InfraRecorder (for Windows).

Once you have the necessary software installed, follow that software's instructions for burning the ISO image to a disk.

#### Use a Flash Drive/Memory Card

To create a bootable installation flash drive/memory card on Linux/BSD/Mac, use a modified version of the following dd command (as root):

root@localhost:~# dd if=cucumber-linux-1.1-x86\_64-full.iso of=/dev/sdx

Replace cucumber-linux-1.1-x86\_64-full.iso with the path of the ISO image you downloaded. Replace /dev/sdx with the path of the *block device* of the flash drive/memory card you want to write the ISO to (i.e. use /dev/sdb, not /dev/sdb1). Make sure you use the right device here, otherwise you could accidentally corrupt another device on your system or delete your operating system all together.

#### **Option 2 – Purchase the Official Cucumber Linux Installation CD/DVD**

If you would rather not download an ISO image yourself, Cucumber Linux also sells official installation media. The official installation CD and/or DVD can be purchased at <a href="http://cucumberlinux.com">http://cucumberlinux.com</a> or ordered using the order form in the back of this book.

## 2. Boot from the Installation Medium

The way you do this is system dependent. Consult your hardware's documentation to figure this step out.

The Cucumber Linux installer supports both UEFI boot and legacy BIOS boot. If you want the installer to set up UEFI boot for your installed system, you must boot using UEFI here, not legacy boot.

If you are unsure of whether to use legacy boot or UEFI, the safer option is to use legacy boot. Only certain newer computers will work with UEFI, however all computers (except Apple computers) will work with legacy boot.

Once the installer has booted, select the 'Install' option from the main menu.

## 3. Partition and Format the Hard Drives

The installer will now drop you to a root prompt to partition and format your drives. This is the most difficult part of the installation, but don't be discouraged. To help people who are unfamiliar with partitioning and formatting drives from the Linux command line, we have included a handy guide on this.

To start, you need to partition your drives. This can be done using the fdisk, cfdisk, gdisk or cgdisk programs. The fdisk and cfdisk programs are used for partitioning drives using an MS DOS partition table, while gdisk and cgdisk are used for partitioning drives using a GPT partition table. The difference between MS DOS and GPT partition tables is explained in the "Create a Partition Table" section.

In this example, we will use fdisk and a MS DOS partition table. If you wish to use a GPT partition table, instructions for this can be found in the Alternate Installation Options section at the end of this chapter. To start partitioning a drive, run the command fdisk /dev/sdx, where /dev/sdx is the block device you want to format (i.e. sda for your first hard drive). Once in fdisk, you can type m followed by enter at any time for help.

Note that fdisk doesn't actually write any changes to your drives until you enter the w command. If you would like to abort or start over at any time, you can enter q to exit fdisk without making any changes to your drive.

#### **Create a Partition Table**

If the drive already has a partition table you can skip this step. This will be the case if you have partitioned this drive before or if there is already another operating system installed on the drive.

If you have not previously partitioned this drive, create a new partition table. This is done by entering o to create a MS DOS partition table. To create a GPT partition table, use the command gdisk instead of fdisk and enter o. If you are planning to use this drive in a version of Microsoft Windows prior to Windows 8, you will need to use a MS DOS partition table. If the drive is larger than 2 TB, you will need to use a GPT partition table in order to utilize all of the drive's capacity. If you don't fall into either of these categories, then it doesn't make a huge difference which one you choose. If in doubt, MS DOS is the safer option.

#### **Create the Partitions**

#### **Creating a Single Partition**

Now, type **n** to create a new partition. You may be asked if you would like to use a primary or extended/logical partition. If you are, you can elect to use either; Cucumber Linux doesn't really care. Keep in mind though that you can have only four primary partitions, and some versions of Windows will work *only* if they are installed on a primary partition. For details about using extended/logical partitions and other advanced partitioning techniques, see TLDP's article about partitioning at <u>http://tldp.org/HOWTO/Partition/fdisk\_partitioning.html</u>.

Next, select a partition number. It doesn't really matter what number you choose, however the convention is to use the next lowest number available. If you do not enter a number here, the lowest available number will be used. Once you have entered a partition number, press enter to continue. If you are not sure what to do here, just press enter without typing anything else.

If you would like to use a non-default offset for the partition, you can change the first sector now. If you would like to accept the default, just press enter. If you are not sure what to do here, just press enter.

Now enter the desired size of the partition in the format +ng, where n is the desired size in gigabytes. For example, to create a partition that is 20 GB in size, enter +20g. You can also use K, M or T in place of G to specify the size in kilobytes, megabytes or terabytes respectively. Press enter to confirm the size. Alternatively, you can opt to use all of the remaining space on the drive for this partition by just pressing enter without typing anything.

#### **Creating Additional Partitions**

Cucumber Linux can be installed using a single partition, however it is often desirable to create additional partitions (for swap space, for /home and so on). If you would like to use additional partitions, repeat the 'Creating a Single Partition' section as needed to create the additional partitions.

#### Troubleshooting

#### I get an error message when I try running the p command

If you get an error message that says To create more partitions, first replace a primary partition with an extended partition or something similar, it will be necessary to shrink or delete one of your existing partitions first. This can be done by typing a to delete a partition.

#### **Finish Partitioning**

Once you have set up all of your partitions, type  $\frac{1}{2}$  followed by enter to confirm the changes and exit fdisk.

#### **Partitioning Example**

The following example creates two partitions on /dev/sda: a 512 MB partition for /boot (/dev/sda1) and a 31.5 GB partition for / (/dev/sda2).

```
bash-4.3# fdisk /dev/sda
Welcome to fdisk (util-linux 2.27.1).
Changes will remain in memory only, until you decide to write them.
Be careful before using the write command.
Command (m for help): o
Created a new DOS disklabel with disk identifier 0xdb87f542.
Command (m for help): n
Partition type
      primary (0 primary, 0 extended, 4 free)
   р
       extended (container for logical partitions)
   е
Select (default p): p
Partition number (1-4, default 1): 1
First sector (2048-67108863, default 2048): <press enter>
Last sector, +sectors or +size{K,M,G,T,P} (2048-67108863, default 67108863): +512M
Created a new partition 1 of type 'Linux' and of size 512 MiB.
Command (m for help): n
Partition type
       primary (1 primary, 0 extended, 3 free)
   р
       extended (container for logical partitions)
   е
Select (default p): p
Partition number (2-4, default 2): 2
First sector (1050624-67108863, default 1050624): <press enter>
Last sector, +sectors or +size{K,M,G,T,P} (1050624-67108863, default 67108863):
<press enter>
Created a new partition 2 of type 'Linux' and of size 31.5 GiB.
Command (m for help): w
The partition table has been altered.
Calling ioctl() to re-read partition table.
Syncing disks.
bash-4.3#
```

#### Format the Partitions

Now it is time to format the new partitions. This is done by using a mkfs command. The Cucumber Linux Installer currently supports the following formats: bfs, btrfs, cramfs, ext2, ext3, ext4, minix, ntfs, reiser and vfat (fat32). To format a partition, run mkfs.<format> /dev/sdx1, substituting '<format>' with one of the aforementioned formats and 'sdx1' with the partition you would like to format. For example, to format the first partition of the first hard drive as ext4, run mkfs.ext4 /dev/sda1.

If the partition you are formatting is on a solid state drive, it is recommended that you use a non-journaling or SSD aware filesystem (such as ext2 or btrfs respectively). This will help extend the life of your solid state drive significantly. If you have a traditional disk drive, it doesn't make a huge difference which filesystem you use. If in doubt as to which file system to use, we recommend using btrfs.

Note that if you use a filesystem other than ext 2, 3 or 4 for your root partition, it will be necessary to create a separate boot partition. This partition can be about 256 MB in size and must be formatted as ext 2, 3 or 4.

To format a partition as swap space, run mkswap /dev/sdx1 substituting the partition in for 'sdx1' as before.

## 4. Set up your Mount Points

After you finish partitioning, the installer will let you choose the mount points for your partitions. Here are a few things to be mindful of when choosing your mount points:

You must set the mount point for one partition to /. This will be the root partition for your system. The root partition should be formatted using some Linux filesystem (i.e. ext2, ext3, ext4, btrfs, etc), not a DOS/Windows (i.e. fat or ntfs) filesystem.

If you opted to use a non-ext filesystem for your root partition and you want to use legacy boot, you must set the mount point for your ext2, 3 or 4 formatted boot partition to /boot.

If you are planning to use UEFI to boot your system, you must set the mount point for your fat formatted EFI system partition to /boot/efi.

If you have a swap partition, set the mount point for it to  $_{swap}$ . While not strictly required, having a swap partition is recommended for systems with non solid state drives.

If you would like to mount any additional partitions, the standard place to do this is in /media/<partition name>. Data on any partitions mounted under /media will not be modified during the installation.

In the event that you forgot to format one of your partitions or need to get back to a root shell, there is a 'Root Shell' button on this menu for these purposes.

When you are finished, select 'Done' to continue with the installation.

## 5. Selecting the Package Groups

Now it is time to select the package groups you want to install. Each package group contains a series of packages that are all related in some way. For example, the dev-base group provides basic development packages, such as the GCC and binutils. For more information on package groups see Appendix A.

There are a few things that are worth noting here:

The base package group is the only package group that is required in order to get a bootable system, and as such is the only package group selected by default.

Despite this however, the recommended way to install Cucumber Linux is to install all of the available package groups. This is also the only way that is officially supported (with one exception, which is discussed below). This will require 5.5 GB of disk space for the 32 bit version and 6.1 GB of disk space for the 64 bit version (as of Cucumber Linux 1.1). Keep in mind that Cucumber Linux doesn't have any automated dependency tracking, so installing every package is the only way to ensure that there are are no unresolved dependencies.

The one exception to this is the multilib package group. This group contains a compatibility layer for running 32 bit binaries on the 64 bit version of Cucumber Linux. Generally, this is useful for running third party, closed source software that has only a 32 bit version available. That being the case, it necessary to install the multilib package group if and only if you are planning on running 32 bit applications on a 64 bit system. If you are planning to use only 64 bit applications (this is the default on Cucumber Linux), you may skip the multilb group. This is the only other option that is officially supported.

Once you have selected the package groups you want, select 'Ok' to continue. Next you will be asked how you would like to install the packages. You have two options here. If you opt for 'full', every package will be installed automatically. This is the safest way to do the installation. If you select the other option, 'prompt', the installer will prompt you before it installs every single package, asking if you want to install or skip this package. This option takes while. Once you have made your decision, select 'Ok' to begin actually installing the system.

Now you can either sit back and watch the installation or exercise your fingers on your keyboard, depending on how you decided to install the packages.

## 6. Configuring the System

Once the base system installation has finished, you will be asked if you would like to configure your system. This section is entirely optional; however, if you skip it all of your system's configuration (hostname, root password, timezone, etc.) will be set to the default values. It is typically recommended not to skip this section.

#### Setting a Hostname

If you opt to configure your system, you will be asked to enter a hostname for your new system. The hostname is a human readable name that is used to identify your system on the network. You should select a hostname that describes the system's purpose but is not too long. A hostname may contain letters, numbers and the dash (-) character.

Once you are satisfied with your hostname, select 'Ok' to continue. You can change your hostname later by editing the /etc/hostname file.

#### **Setting a Root Password**

Next, you will be asked if you would like to set a root password. If you select no, the root user will be able to login without entering a password, which creates a very severe security hazard. Therefore, it is strongly recommended that you set a root password here.

You will be required to make your root password at least five characters long; however, it is recommended that you it make longer than this. Having a reasonably long and complex root password is one of the single most important things you can do to keep your system secure. Guidelines for coming up with a strong, secure password can be found in Appendix B.

#### **Selecting the Timezone**

Next, you will be asked to select your timezone. Simply follow the menu based system. If you make a mistake, continue on and select any timezone. Then at the end, it will ask you if the information you entered is correct. Select no and the selection process will start over.

#### **Additional Configuration**

Finally, you will be asked if you would like to enter a shell to perform any additional configuration. It is up to you whether you would like to further configure your system now or perform the rest of you configuration after the installation process is complete.

Some typical tasks that may be done here (such as adding an unprivileged user, configuring networking and setting up a graphical interface) are explained in chapter two. These tasks can also all be done after the installation process is complete.

If you select 'Yes', you will be sent to a root shell chrooted to the newly installed system (this effectively means you will be logged in as root on the new installation). Perform any additional configuration you would like, and then exit the shell to finish up the installation process. Do not reboot yet, or you will end up with an unbootable system.

## 7. Making the System Bootable

The installation is almost done now. There remains only one more task: making the system bootable. There are two ways to do this: using the legacy BIOS bootloader or using UEFI. Generally, it is recommended to use the legacy BIOS bootloader; UEFI currently has a lot of issues and works properly only on select motherboards.

#### Installing the Legacy BIOS Bootloader

You will now be asked if you would like to install the GRUB bootloader to the MBR (Master Boot Record). It is recommended you choose yes here. If you choose no, you will be required to do additional work to get your system to boot; this option is intended for experienced users who probably don't need this guide to install Cucumber Linux in the first place and consequentially either won't be reading it or will know to ignore this recommendation. Therefore, we can safely recommend that anyone reading this guide select yes.

Next, you will be asked to enter the path of the drive to install GRUB on. It is important to enter the path of the drive here, not the path of the partition. For example, enter /dev/sda, not /dev/sda1. Usually, you will want to install GRUB to the drive you partitioned back in step three.

Finally, you will be asked if you want to install a new GRUB configuration file. If you choose no, additional configuration will be necessary to boot your system. If you choose yes, any other existing operating systems you may have installed will temporarily be rendered unbootable. Unless you are an experienced user (who as previously discussed, probably won't be reading this anyway) it is recommended you choose yes here.

#### Setting up UEFI Boot

As previously mentioned, UEFI is known to have many problems, so we recommend that users stick with using the legacy BIOS. Nonetheless, if you booted the installer using UEFI, you will be given the option to set up UEFI boot now. If not, then good: you can skip this section.

If you opt to use UEFI and experience trouble booting after the installation, go into your BIOS and switch back to legacy boot. If you chose to install the legacy BIOS bootloader, your computer will fall back to that and still be able to boot your system.

## 8. Reboot into your New Cucumber Linux Installation

After you finish making the system bootable, Cucumber Linux will be installed and ready to use. You will be brought back to the main menu of the installer, and all you have to do now is reboot, which can be done by selecting the 'Restart' option from the menu.

When you reboot, make sure to remove the installation medium. Otherwise, you could accidentally start the installation over instead of booting into your new installation.

## Things to do Post Installation

//TODO

## **Alternative Installation Options**

//TODO

## **Chapter 2 – Basic System Configuration**

## **User and Group Management**

On most systems, it will be desirable to create users besides the root user for every day tasks. Additionally, it is often disable to create additional groups for categorizing users and handling permissions.

This section provides a basic overview of Cucumber Linux specific user management. For more general information on user management, there is a great article on the Arch Linux Wiki at <a href="https://wiki.archlinux.org/index.php/Users\_and\_groups#User\_management">https://wiki.archlinux.org/index.php/Users\_and\_groups#User\_management</a>.

#### **User Management Utilities**

For user management, Cucumber Linux supplies the commonly used useradd, usermod and userdel programs. These programs should be run as root. More information can about them can be found in their man pages (these can be accessed by running man useradd, man usermod and man userdel).

These utilities are provided by the shadow package, and therefore have out of the box support for encrypted passwords and other security features provided by shadow.

#### Adding a User the Easy Way

For adding users, Cucumber Linux also provides an interactive, more user friendly utility: adduser. This is the easiest way to create new users; to start simply run adduser as root.

First, you will be asked to enter a name for the user account. User account names may contain lowercase letters, numbers and dashes.

Next you will be asked to a User ID. Just press enter and one will automatically be selected.

Then you will be asked what you would like the user's primary group to be. The default option is users, which every human user account on the system is usually a member of. You are of course free to change this, but the group you enter must already exist.

Then enter any additional groups you want the user to belong to. For this step, the adduser program will provide the details of how this works. If you are unsure what to enter, the following is recommended: if this user is going to be logging in as a desktop user, press up and then enter; otherwise, just press enter.

Enter the user's desired home directory. Usually this doesn't need to be changed: press enter to accept the default.

Enter the user's desired shell. This also doesn't usually need to be changed and you can press enter to accept the default. The only shell that is natively supported on Cucumber Linux is the Bash shell, so there isn't much of a choice to be made here unless you manually install a third party shell.

Enter the expiration date for this user account. If you enter a date here, the account will be disabled on that date and the user won't be able to log in after that. You will usually want to leave this blank.

Press enter to actually create the user account.

Next you will be asked to enter the user's name and some other information. Changing this information is not necessary and it has no impact on the account's functionality. This part is really just a throwback to when Unix was used internally at AT&T and it was useful to be able to figure out a user's office information; it is seldom used on modern systems.

Finally, enter a password for the user. While the password selection for an unprivileged user is slightly less important than the password selection for the root user, you should still use a strong password here; failure to do so will still significantly lessen the security of your system. Guidelines on creating a strong password can be found in Appendix B.

The account setup will now be complete and the new account will be ready for use.

#### **Group Management Utilities**

For group management, Cucumber Linux supplies the commonly used counterparts to the user management utilities: groupadd, groupmod and groupde1, all of which should also be run as root. These programs are also provided for by the shadow package and therefore also support shadow's enhanced security features out of the box.

## Starting a GUI (Graphical User Interface)

Cucumber Linux provides a GUI (Graphical User Interface) through the X Window System. The basic X Window system is provided by the packages in the x-base package group, and additional X components are provided in the x-general package group. A list of the contents of these package groups can be found in Appendix A.

There are two ways to start a graphical user session on Cucumber Linux, which are explained below. Note that it is considered a bad idea and a security hazard to run a GUI as root, so it is a good idea to create an unprivileged user first. How to do this is explained in the previous section, entitled "User and Group Management".

#### Starting a GUI Once Logged In

The first option is to start the X Window System once you have already logged in from the command line login screen. To do this, run the command starts. This will not start a full desktop environment; it will start a very basic window manager known as TWM.

Since most users prefer a full desktop environment or a more robust window manager, Cucumber Linux also provides the XFCE desktop environment (provided by the xfce-base and xfce-general package groups) and the WindowMaker window manager (provided by the x-general package group).

To start the full XFCE desktop environment, run the command startxfce4.

To use WindowMaker, it will be necessary to edit the Xinit configuration file, which is located at /etc/X11/app-defaults/xinitrc. Edit this file, and change the line that says twm & to wmaker & and then run starts. Note that editing this file requires root permissions and will change the default window manager for the entire system. To change window manager for only yourself or without root access, first copy the /etc/X11/app-defaults/xinitrc to ~/.xinitrc. Then make the aforementioned changes to the ~/.xinitrc file.

Pro Tip: by default, the X window system will automatically start a clock and three terminals when you run **startx**. To prevent this from happening, remove last four lines from the default xinitrc and change the twm &/wmaker & line to exec twm/exec wmaker.

#### **Enabling a Graphical Login Screen**

The other way to start the X Window System is to enable the graphical login screen. This will also allow users to log in using a GUI, and will make it so individual users need not run startx.

To do this, simply edit the /etc/inittab file as root, and change the default runlevel from 3 to 5. This can be done by changing the id:3:initdefault: line to id:5:initdefault:.

Once you have done this, reboot your system and you will be greeted by a graphical login screen with a glorious picture of cucumbers on it!

Note that this requires the lxdm package to be installed, which is found in the xapps-general package group. If you do not get a graphical login screen after changing your /etc/inittab, you most likely do not have the lxdm package installed. Instructions on installing packages can be found in Chapter 5.

## Networking

#### **Configuring Network Interfaces**

Cucumber Linux provides a series of scripts for configuring and managing network interfaces. These scripts are found primarily in the /etc/network directory. This directory has three subdirectories: interfaces, methods and templates.

The methods directory stores helper scripts for each configuration method. Each method script is a driver for the networking configuration system that handles one specific type of network configuration. For example, the dhcp method script is the driver that configures a network interface to use DHCP.

The interfaces directory is where the actual configuration files reside. This is the only directory you will ever need to edit files in. The configuration files use Bash sytax: they define several variables for controlling the configuration for a network interface. The name of the configuration file is the network interface that will be configured by that file, as displayed by ifconfig. For example, the 10 file configures the 10 (loopback) interface.

The templates directory contains various template configuration files. These files provide near complete configurations that can be copied over to the interfaces directory, and then edited to fit your needs.

Every configuration file (in the /etc/network/interfaces directory) is required to define two variables: ONBOOT and METHOD. The ONBOOT variable controls whether or not this network interface should be automatically brought up when the system boots. It should be set to yes or no. The METHOD variable controls which configuration method will be used for this interface. It can be set to any method in the /etc/network/methods directory.

Some configuration methods require that additional variables be defined in the interface configuration file (for example, using a static configuration would require you to define ADDRESS). Which additional variables can be used varies from one configuration method to the next. To see which variables are valid for a specific method, run the command **ifhelp <method>**, where <method> is a configuration method from /etc/network/methods.

Additionally, some configuration methods will allow you to chainload another configuration method. What this means is that when the interface is brought up, the method specified by METHOD will run first, and then the chainloaded method will run. This is particularly useful when connecting to wireless networks: METHOD will be set to wireless or wireless-wpa. These configuration methods only connect you to the wireless network; they do not handle IP address assignment. Therefore, they possess the ability to chainload to another configuration method (like static or dhcp) and let that method handle the address assignment. When chainloading, it will be necessary to define all the variables required by both the primary method and the chainloaded method.

Once you have written a configuration file for an interface, run the command # ifup interface to bring up the network interface using the configuration. Network interfaces can manually be brought up and down at any time by using the # ifup <interface> and # ifdown <interface> commands respectively (as root). Alternatively, you may run # /etc/init.d/network restart to restart the networking service, which will bring up all the network interfaces that have ONBOOT set to yes in their configuration files.

#### Setting a Static IP Address

To configure a network interface to get a static IPv4 address, copy the static template from /etc/network/templates/dhcp to /etc/network/interfaces/<interface>, where <interface> is the name of the interface to configure, as it appears in ifconfig. Edit the configuration file you just created in /etc/network/interfaces, and change the Address, NETMASK, BROADCAST, and GATEWAY variables to the desired IPv4 address, subnet mask, broadcast address and default gateway respectively. Only the address field is strictly required: you can delete the other fields; however, this may result in issues. Unless you specifically know that you do not need the optional fields, you should fill them all out. Then change the line <code>ONBOOT=no</code> to <code>ONBOOT=yes</code>. Finally, restart the network service by running # /etc/init.d/network restart.

To configure the interface to use IPv6 instead of IPv4, use the static6 template rather than the static template. In this case, you need only set the IPv6\_ADDRESS and IPv6\_GATEWAY variables the IPv6 address (with the CIDR slash notation for the subnet mask) and the and IPv6 default gateway respectively. Otherwise proceed as you would for configuring IPv4.

To configure a dual IPv4 and IPv6 stack, use the static4and6 template. Fill out both the IPv4 and IPv6 sections as directed above.

#### Using DHCP to get a Dynamic IP Address

To configure a network interface to use DHCP to attempt to get IPv4 and IPv6 addresses, copy the dhcp template from /etc/network/templates/dhcp to

/etc/network/interfaces/<interface>, where <interface> is the interface name you want to configure DHCP on, as it appears in ifconfig. Next, edit the configuration file you just created in /etc/network/interfaces and change the line ONBOOT=no to ONBOOT=yes. Finally, restart the network service by running # /etc/init.d/network restart.

To configure the interface to use only IPv4 or IPv6 instead of a dual network stack, use the dhcp4 or dhcp6 template respectively in place of the dhcp template.

#### **Configuring a Wireless Network Interface**

Configuring a wireless network interface to connect to a wireless network is a little trickier.

#### Scanning to get a List of Available Wireless Networks

If you do not know the SSID of the network you will be connecting to, it will be necessary to scan for a list of available wireless networks. To do this, first ensure that your wireless interface is turned on; this can be done by running # ifconfig <interface> up, where <interface> is your wireless interface. Next, scan for available networks by running # iwlist scan. This generates a lot of output, so you may want to pipe it to less or redirect it to a file.

Scroll through this output until you find the network you wish to connect to. Take note of ESSID: and Encryption Key: fields. Also take note of whether or not there is a line containing something similar to IEEE 802.11i/WPA; you will need this information later.

#### Choosing the Right Template

There are several wireless templates in /etc/network/templates: any template that starts with wireless- or wpa- is a template for a wireless configuration. It is very important that you choose the right template here. This can be a little complicated and confusing at first, so here is a guide for choosing the correct template for your network:

Wireless templates make use of chainloading to connect to a network and get an IP address. Each wireless template name has two parts, separated by a dash. The first part is the primary configuration method (used to connect to the wireless network), and the second part is the chainloaded method (used to get an IP address after connecting). It is important to understand that connecting to a wireless network and getting an IP address are two completely independent processes, both of which must be completed successfully in order to communicate over a wireless network connection.

Use a template where first part of the template name is wireless or wpa. This indicates that the network you are connecting to is an unsecured network or a wpa secured network respectively. If the network had Encryption Key:On and a line similar to IEEE 802.11i/WPA, then use a template that starts with wpa. Otherwise, use a template that starts with wireless.

Choose the second part of the template based on how you would like the IP address to be assigned after connecting to the network. Choose the template ending in dhcp to get a dynamic IP address via DHCP or the template ending in static to manually assign a static IP address.

#### Editing the Configuration File

Once you have chosen the correct template, copy the template from /etc/network/templates/<template> to /etc/network/interfaces/<interface>, where <interface> is the name of the interface you are configuring and <template> is the name of the template you chose in the previous section. Then edit the configuration file you just created in /etc/network/interfaces and change the line ONBOOT=no to ONBOOT=yes.

Now, change the WIRELESS\_SSID or WPA\_SSID variable to the ESSID field value from the # iwlist scan command ran earlier (you will see only one of these variables; fill out the one you see and ignore the other). If you are connecting to a WPA secured network, set the WPA\_PASSPHRASE variable to the passphrase for the network (when you do this, make sure to escape any special characters in passphrase by putting a backslash before them).

Next, find the lines that say something like:

##### End Wireless/WPA Method Specific Configuration #####

##### Begin DHCP/Statis Method Specific Configuration #####

Proceed to fill out all the lines below here as directed Setting a Static IP Address or Using DHCP to get a Dynamic IP Address section, depending on which template you chose.

Finally, save the changes to the config file and restart the network service by running # /etc/init.d/network restart.

#### **Configuring the Firewall**

As soon as networking is enabled, it opens a system up to a massive array of attacks. In order to help mitigate against many network attacks, it is of paramount importance that the system firewall be configured correctly.

Cucumber Linux uses the standard Linux firewall: Iptables. Configuring Iptables can be a very complicated task. This section will cover how to set up reasonable default firewall rules. Every subsequent section that instructs you on how to modify the firewall rules will assume that you are using the firewall rules from this section. Additionally, this section will cover how Iptables is integrated with Cucumber Linux, including how to change the default firewall rules and setting Iptables to be active when the system boots.

#### **Basic Iptables Configuration**

There are two different flavors of Iptables: one for IPv4 and one for IPv6. These two flavors both run at the same time, but are configured independently of each other. The IPv4 version of Iptables is controlled by the *iptables* command, and the IPv6 version is controlled by the *iptables* command. Other than the difference in the command, the two flavors work much the same. Much of this section will use only the *iptables* command; however, the instructions can be applied just as easily for IPv6 by replacing the *iptables* command with the *ip6tables* command.

To list the currently active firewall rules, run the command # iptables -nvL. The -n flag tells Iptables not to resolve numbers to their service names, -v tells it to be verbose and show columns that would otherwise be hidden and -L tells it to list the currently active rules. On a default Cucumber Linux installation, the output of the command should look similar to the following:

root@cucumber:~# iptables -nvL iptables -nvL Chain INPUT (policy ACCEPT 0 packets, 0 bytes) prot opt in pkts bytes target out source destination Chain FORWARD (policy ACCEPT 0 packets, 0 bytes) pkts bytes target prot opt in out destination source Chain OUTPUT (policy ACCEPT 0 packets, 0 bytes) pkts bytes target prot opt in out source destination

This indicates that there are no firewall rules currently active, as is the default in Cucumber Linux 1. Notice that the output of the command is broken down into different chains. Rules under the INPUT chain will be applied to incoming network traffic, while rules under the OUTPUT will be applied to outbound traffic. The FORWARD chain is used by routers and plays a less significant role on systems that are not routers. Also note the policy for each chain is set to ACCEPT. The policy for a chain is effectively the default rule that will be applied if no other rules on the chain match a packet. In this case, since there are no other rules and the policy (default rule) is ACCEPT, all traffic will be allowed through the firewall.

#### Writing Iptables Rules

When writing firewall rules, it is considered best practice to block all traffic by default, and then allow only the traffic you need through the firewall. In Iptables, this is accomplished by changing the policy from ACCEPT to DENY for the INPUT and FORWARD chains. This is accomplished by running the following commands (warning: this will temporarly block all your network traffic).

root@cucumber:~# iptables -P INPUT DROP
root@cucumber:~# iptables -P FORWARD DROP

It is considered permissible practice to allow all outbound traffic since attackers are usually trying to break in from the outside, not the other way around.

Now, we will allow incoming traffic that is related to connection that is already established. What this means is that if the local system starts a conversation with a remote system the remote system will be allowed to reply, but a remote system will be unable to start a new conversation with the local system. This is accomplished by running the following command:

root@cucumber:~# iptables -A INPUT -m conntrack --ctstate RELATED,ESTABLISHED -j ACCEPT

As you may have noticed, there are a lot of different flags that can be used with Iptables. Explaining them all is beyond the scope of this guide; however, they are all very well documented in the Iptables manual. This manual can be accessed by running the command man iptables.

If you run iptables -nvL now, you will see the new firewall rules shown below.

root@c	cucumbe	er:~# <mark>ipta</mark>	lbles -nvL				
iptabl	Les -n\	/L					
Chain	INPUT	(policy D	ROP 0 pack	ets, O I	oytes)		
pkts	bytes	target	prot opt	in	out	source	destination
Θ	0	ACCEPT	all	*	*	0.0.0/0	0.0.0.0/0
		ctstate R	RELATED, EST.	ABLISHE	0		
Chain	FORWAF	RD (policy	′ DROP 0 pa	ckets, (	🤉 bytes)		
pkts	bytes	target	prot opt	in	out	source	destination
Chain	OUTPUT	「 (policy	DROP 0 pac	kets, O	bytes)		
pkts	bytes	target	prot opt	in	out	source	destination

While not complete for every setup, this is a good set of firewall rules to start with if in doubt. These rules should work well on a desktop system. It is for server systems that they will require additional modification. The exact setup required for each server will be unique and will be covered in detail in chapter four.

#### **Cucumber Linux Iptables Integration**

By default, Iptables does not save any modifications to the firewall rules across reboots. To address this, Cucumber Linux provides a script for saving and restoring Iptables rules at /etc/init.d/iptables. This single script controls both the IPv4 and IPv6 flavors of Iptables and allows for one of the following five arguments:

save	Saves the current firewall rules (IPv4 & IPv6), making the new default.
panic	Puts the system firewall into a "panic" state, temporarily blocking all network traffic.
clear	Clears the current firewall rules, allowing for all traffic to pass in and out.
stop	Same as clear.
start	Loads the default rules (IPv4 & IPv6), making them the currently active firewall rules.

So to save the current rules, run:

```
root@cucumber:init.d# /etc/init.d/iptables save
```

Even after doing this; however, the firewall rules will still not get loaded when the system boots. In order to have the default firewall rules go into effect when the system boots, it is necessary to enable the *iptables* service on boot (it is disabled by default). Instructions for doing this can be found in the "Starting Services and Daemons on Boot" section of this chapter.

## **Changing System Wide Settings**

#### Starting and Stopping Services and Daemons

Daemons on Cucumber Linux are by System V style init scripts. These scripts are called init scripts and are located in /etc/init.d. Each script in this directory controls one service and takes a single argument: an action to perform for that service. There are several different actions that can be performed (such as start, stop, restart or status), and different scripts support different actions. To see which actions are supported by a particular script, run that script without any arguments; it will display a list of supported actions.

For example, to start the Apache HTTP server, run /etc/init.d/httpd start. To ensure that it is running now, run /etc/init.d/httpd status. Then to stop the server, run /etc/init.d/httpd stop. If you want to see what other actions are supported by the httpd init script, run /etc/init.d/httpd.

#### Starting Services and Daemons on Boot

Cucumber Linux uses System V style symbolic links to control the starting of services at boot. These symbolic links are located in /etc/rc.d/rcx.d, where x is a single digit 0 through 6 indicating which runlevel corresponds to that directory.

While it is possible (and sometimes desirable) control which services start at boot by modifying the symbolic links directly, Cucumber Linux also provides a more user friendly

interactive menu for controlling which services will start when the system boots. To access this menu, run the command # pkgtools as root. Select setup from the main menu, and then select services from the System Setup Scripts menu (select it by scrolling to it and pressing the space bar, and then pressing enter select ok).

Now you will be presented with a menu where you can select/deselect which services will start at boot time by pressing the space bar. When you selected your desired services, press ok to apply the changes. If you make a mistake, you can select <code>cancel</code> to return to the main pkgtools menu without changing your system.

It's worth noting that in general, running more daemons makes your system less secure. For this reason, most services are disabled by default. It is recommended that you enable only the services you need and leave the rest disabled. When you do enable a service, it is also advised that you take time to research how to secure that service properly.

#### **Changing the Kernel Version**

When you start Cucumber Linux, you will notice there are two different boot options: the default boot option and fallback kernel boot option. These two options are almost identical. The only thing that differs between them is which version of the kernel they boot. If you do not change anything, the default option will boot the version of the kernel that was installed most recently and the fallback option will boot the version of the kernel that was used to originally install Cucumber Linux.

The reason for having two boot options is as follows: if a kernel update fails or you install an incompatible version of the kernel, this results in the default boot option failing to boot your system. In this case, it is handy to have a fallback boot option that boots a kernel that you know works so that you can still boot the system and then fix the default kernel.

It is possible to manually change which version of the kernel is the default kernel and the fallback kernel. To do this, run the command pkgtools as root. Select setup from the main menu, and then select kernel-version from the System Setup Scripts menu (select it by scrolling to it and pressing the space bar, and then pressing enter select ok).

You will now be presented with a menu where you can change your default and fallback kernels. It is important that you change the fallback kernel only to a version of the kernel that you know works; otherwise you could still find yourself with an unbootable system if a kernel update fails. It is also recommended that the default kernel and the fallback kernel not be the same, or else you could end up with an unbootable system if that version of the kernel is ever accidentally removed.

## Chapter 3 – Using Cucumber Linux as a Desktop Operating System

## Setting Up a Cucumber Linux Desktop

Adding an Unprivileged User

Setting the GUI to Start on Boot

Configuring a Desktop Firewall

## **Useful Programs**

## Chapter 4 – Using Cucumber Linux as a Server Operating System

## Setting up a Server

Setting up a Basic Web Server

Setting up Wordpress and a L.A.M.P. Server

Setting up an RSYNC Server

Setting up an FTP Server

Setting up a Mail Server

Setting up a Caching DNS Server

## **Tips for Secure Server Administration**

**Server Firewall Configuration** 

**Configuring SSH Safely and Securely** 

## **Chapter 5 – System Maintenance**

## Package Management

- Managing Packages with Pkgtools
- Installing Additional Packages Manually using the Installation Medium
- Installing Packages from the Cucumber Linux Repositories with Pickle
- Searching for Packages with Pickle
- Manual Dependency Resolution

## Keeping your System Up to Date

- Updating your System with Pickle
- **Getting Notified about New Updates**

## **Appendix A – Package Groups**

Cucumber Linux 1.1 contains the following package groups:

## base

The base system. Contains enough basic system software to get up and running and have a text editor and basic communications programs.

Package Contents:

acl	file	linux	sed
attr	findutils	linux-firmware	shadow
bash	gawk	linux-headers	sysklogd
bzip2	gettext	lsb-info	sysvinit
coreutils	glibc	mtools	sysvinit-bootscripts
cron	glibc-zoneinfo	ncurses	tar
dialog	gptfdisk	patch	util-linux
e2fsprogs	grep	pciutils	vim
ed	grub	popt	which
efibootmgr	gzip	procps-ng	XZ
efivar	kbd	psmisc	
etc	kmod	readline	
eudev	less	rpm2tgz	

## apps-base

Basic applications that are generally expected to be present and do not require the X window system.

Package Contents:

alsa-utils	cups	fortune	links
bc	cups-filters	ghostscript	lvm2
btrfs-progs	diffutils	groff	lzip
сріо	dosfstools	htop	man

man-pages	pm-utils	sl	unzip
nano	qpdf	sudo	zip
ntfs-3g_ntfsprogs	screen	texinfo	

## apps-general

Additional applications that are generally expected to be present and do not require the X window system.

Package Contents:

hplip	p7zip	sqlite
Im_sensors	parted	unrar

## dev-base

Basic program development tools, such as compilers and debuggers.

Package Contents:

autoconf	bison	gperf	m4
automake	flex	intltool	make
binutils	gcc	libtool	pkg-config

## dev-general

Additional program development tools.

Package Contents:

cmake	git	yasm
elfutils	llvm	

## kernel

The source code for the Linux kernel.

Package Contents:

linux-source

## lang-base

Interpreters and support for common scripting languages.

Package Contents:

perl	perl-XML-Parser	python2	python3

## lang-general

Interpreters and support for additional, less common scripting languages.

Package Contents:

php

slang

vala

## lib-base

Basic system libraries.

Package Contents:

alsa-firmware	glib	libgcrypt	mtdev
alsa-lib	gmp	libgpg-error	npth
alsa-plugins	gstreamer	libjpeg-turbo	nspr
dbus	harfbuzz	libksba	nss
dbus-glib	ijs	libnl	pcre
expat	iso-codes	libpng	polkit
ffmpeg	js	libtiff	poppler
fontconfig	lcms2	libusb	shared-mime-info
freetype	libassuan	lzo	talloc
fuse	libcap	трс	upower
gdbm	libffi	mpfr	zlib

## lib-general

Additional system libraries.

Package Contents:

apr	gobject-introspection	libisoburn	libsoup
apr-util	libarchive	libisofs	libxml2
babl	libburn	liblinear	newt
gegl	libexif	libpcap	
glib-networking	libgudev	libsecret	

## net-base

Basic networking programs.

Package Contents:

bind-client	curl	gnupg	iana-etc
cacertificates	dhcpcd	gnutls	inetutils

iputils	network-scripts	wget	wpa_
net-tools	openssl	whois	
nettle	pinentry	wireless-tools	

#### vpa\_supplicant

## net-general

Additional networking programs, and common daemons.

Package Contents:

apache	mariadb	openssh
iptables	net-snmp	rsync
libndp	nmap	tcpdump

## x-base

The basic X window system and X libraries.

Package Contents:

at-spi2-atk	libevdev	xbitmaps	xf86-input-mouse
at-spi2-core	libglade	xcb-proto	xf86-video-intel
atk	libnotify	xcb-util	xf86-video-nouveau
cairo	libxcb	xcb-util-cursor	xf86-video-vesa
font-util	libxklavier	xcb-util-image	xinit
gdk-pixbuf	mesa	xcb-util-keysyms	xkeyboard-config
gtk+	pango	xcb-util-renderutil	xorg-applications
gtk+3	pixman	xcb-util-wm	xorg-fonts
libXau	qt4	xclock	xorg-libraries
libXdmcp	startup-notification	xcursor-themes	xorg-protocol-headers
libdrm	twm	xf86-input-evdev	xorg-server
libepoxy	util-macros	xf86-input-keyboard	xterm

## x-general

Additional X components and libraries.

Package Contents:

libcroco	lxdm	windowmaker
librsvg	vte	

## xapps-base

Basic applications for the X window system.

Package Contents:

epdfview	leafpad	xscreensaver
galculator	vlc	

## xapps-general

Additional applications for the X window system.

Package Contents:

firefox	gimp	libreoffice	thunderbird

## xfce-base

The XFCE desktop environment.

Package Contents:

adwaita-icon-theme	libwnck	thunar	xfce4-session
exo	libxfce4ui	thunar-volman	xfce4-settings
garcon	libxfce4util	xfce4-appfinder	xfconf
gtk-xfce-engine	lxde-icon-theme	xfce4-panel	xfdesktop
hicolor-icon-theme	polkit-gnome	xfce4-power-manager	xfwm4

## xfce-general

Additional XFCE applications and goodies.

Package Contents:

ristretto	xfburn	xfce4-battery-plugin	xfce4-screenshooter
tumbler	xfce4-alsa-plugin	xfce4-places-plugin	xfce4-terminal

## multilib

Compatibility libraries for running 32 bit applications on the 64 bit version of Cucumber Linux. This package group is available on only the 64 bit version of Cucumber Linux. Any packages listed above that provide libraries (\*.so files) will have a corresponding package in this group.

## **Appendix B – Creating Strong Passwords**

Passwords "need to be of sufficient complexity and secrecy that it would be impractical for an attacker to guess or otherwise discover the correct secret value."

Index