CyberCorps

Participant’s Guide

Version 2.0

June 2016

*This page intentionally left blank.*

Table of Contents

Familiarization Guides 7

Network Admin Familiarization 7

Overview 7

Creating a network map 7

Monitoring Network Traffic 9

Managing and Protecting a Network 11

pfSense Firewall Rule Basics 14

pfSense Packet Captures 15

Network Admin Tools/Skills 15

Email Admin Familiarization 17

Overview 17

SquirrelMail 18

Apache 19

Postfix 19

Dovecot 20

Email Admin Tools/Skills 21

Helpdesk Admin Tools/Skills 22

System Admin Familiarization 23

Overview 23

Managing System Services 23

Configuration and Layout 24

Creating and managing user accounts 25

System Admin Tools/Skills 25

Chat Admin Familiarization 26

Overview 26

Web Admin Familiarization 28

Familiarization Activity 28

Apache Restart Activity 28

Web Site Admin Tools/Skills 29

IT Staff Job Descriptions 30

System Administrator Job Description 30

Summary 30

Essential Duties and Responsibilities 30

Network Administrator Job Description 31

Summary 31

Essential Duties and Responsibilities 31

Help Desk Specialist Job Description 32

Summary 32

Essential Duties and Responsibilities 32

Webmaster Job Description 33

Summary 33

Essential Duties and Responsibilities 33

Email Administrator Job Description 34

Summary 34

Essential Duties and Responsibilities 34

Supporting Documents 35

Basic Connectivity Testing Tools 35

ping 35

telnet 35

traceroute 35

IT Help Desk Notes & Flowchart 36

Create Ticket 36

Contact Helpdesk 36

Complete Ticket 36

OTRS quick guide (v 0.1.0\_10/4/2015) 38

To Log In 38

To Create Customer User 38

Tickets 38

View/Search Text Files 40

head 40

tail 40

cat 40

less 40

grep 41

sort 42

Pidgin instructions  (v0.1\_04/14/2015) 43

Adding contacts 43

Joining Group/Chatrooms 43

User Management Commands 44

w 44

who 44

adduser 44

deluser 45

addgroup 45

delgroup 45

passwd 46

System Monitoring Commands 47

ps 47

top 47

htop 47

netstat 48

ssh 49

File Permission Management Commands 50

chmod 50

chown 51

sudo 51

Help and Editor Commands 52

man 52

apropos 52

emacs 53

File System Commands 54

df 54

du 54

Linux Cheat Sheet 55

Index 56

*This page intentionally left blank.*

Familiarization Guides

# Network Admin Familiarization

## Overview

The network administrator’s role is to plan and coordinate the design, installation and connectivity of computer and network systems to ensure the stable operation of the organization’s IT assets. The network admin must ensure that uptime, performance, resources, and the security of all network systems meet the needs of users. To fulfill these responsibilities, the network admin is charged with developing, configuring, maintaining and supporting all new and existing network hardware, software and communications links.

Network admins should be able to quickly respond to all of the following questions regarding the network systems for which they are responsible.

* What Internet Protocol (IP) subnets and addresses do you manage?
* What servers and endpoints are running on your network(s)?
* Are the servers local or hosted at an external site?
* What services (open ports) are available on each server and host?
* How is your network configured, protected and isolated?
* What connections are allowed between servers, hosts and Internet users?
* Is the network traffic from or to specific endpoints anomalous?
* If anomalous, where do those connections originate and terminate? If the connections include hosts outside your network, where are these endpoints located?

Fortunately, network administrators have a variety of tools and applications at their disposal to help them meet their myriad responsibilities. In the sections below, the various facets of the position are organized into broad categories. Within each category, you’ll find a brief discussion of the tools and applications available to meet a network admin’s needs.

## Creating a network map

A basic responsibility for all network administrators is to understand the network layout and to know what servers and endpoints are running on the network. In many cases, a network diagram may already be available; however, it is important to ensure that network diagrams are kept up-to-date by re-mapping the network on a regular basis.

Several tools exist to facilitate network mapping and connectivity testing. In this section, we’ll explain how the Unix/Linux ping and traceroute commands can be used for this purpose.

**ping** is a simple command used to test the reachability of a host on a network. It also reports the round-trip time for messages sent from the originating host to a destination host and back.

For example, the ping session shown below is used to confirm that the host 10.0.2.100 is reachable via the network. Note the use of the -c (count) option, which takes an integer argument representing the number of ping packets that should be sent. By default, on Linux systems, ping will continue sending pings until Ctrl-C is pressed.

$ ping -c 4 10.0.2.100

PING 10.0.2.100 (10.0.2.100): 56 data bytes

64 bytes from 10.0.2.100: icmp\_seq=0 ttl=62 time=1.582 ms

64 bytes from 10.0.2.100: icmp\_seq=1 ttl=62 time=1.701 ms

64 bytes from 10.0.2.100: icmp\_seq=2 ttl=62 time=1.715 ms

64 bytes from 10.0.2.100: icmp\_seq=3 ttl=62 time=1.807 ms

--- 10.0.2.100 ping statistics ---

4 packets transmitted, 4 packets received, 0.0% packet loss

round-trip min/avg/max/stddev = 1.582/1.701/1.807/0.080 ms

The **traceroute** command displays the entire route between the source host and the destination. traceroute lists all the routers it passes through along the way. In addition, it measures and reports the transit delays of packets across the network.

$ traceroute 10.0.2.100

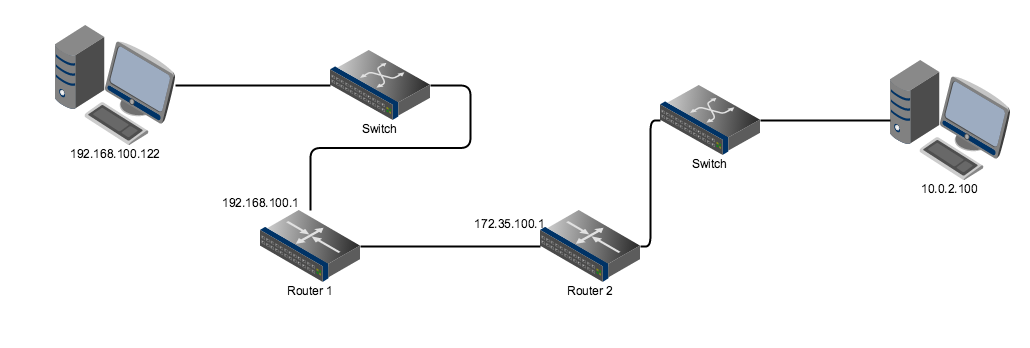
traceroute to 10.0.2.100 (10.0.2.100), 64 hops max, 40 byte packets

1 192.168.100.1 (192.168.100.1) 0 ms 4 ms 0 ms

2 172.35.100.1 (172.35.100.1) 0 ms 3 ms 1 ms

3 10.0.2.100 (10.0.2.100) 9 ms 0 ms 0 ms

Based on the above traceroute command result, we can see that there are two routers between the source and destination hosts, with IP addresses 192.168.110.1 and 172.35.100.1. Using this information, as well as the IP address of the source host (in this instance 192.168.100.122), we can begin to build a simple network diagram, as shown in the figure below. (Note: The network switches shown in the diagram are assumed to exist, since their presence is not disclosed by the traceroute command.)



However, this picture of the network is incomplete. Running at host 192.168.100.122, traceroute can only “see” the router interfaces on the side of the routers it faces. To see the addresses of the network interfaces on the other side of these routers, traceroute must be run from the opposite side, as well. So, now we log in to the original destination host (10.0.2.100) and run traceroute back to the original source host (192.169.100.122).

$ traceroute 192.168.100.122

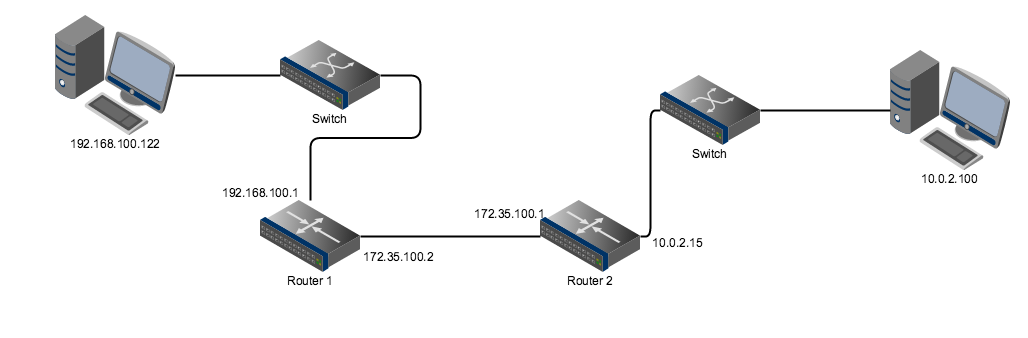
traceroute to 192.168.100.122 (192.168.100.122), 30 hops max, 60 byte packets

1 10.0.2.15 (10.0.2.15) 0.865 ms 0.850 ms 0.838 ms

2 172.35.100.2 (172.35.100.1) 0.956 ms 0.945 ms 0.931 ms

3 192.168.100.122 (192.168.100.122) 3.123 ms 3.122 ms 3.111 ms

Now, given these addresses for the interfaces on the other side of the routers, we can complete the network diagram for the portion of the network that includes these two hosts.



To create network diagrams of larger, more complex networks, it is necessary to log into hosts in different parts of the network and to run traceroute back and forth among several known hosts. And while in this example we used two Linux systems as the source hosts, other options are available. For example, most router and firewall systems (including the VyOS router and pfSense firewall used in the BetaPort environment) also provide access to network software tools such as ping and traceroute.

For more information on these and similar commands, see Basic Connectivity Testing in the Supporting Documents section, as well as the familiarization guides for VyOS and pfSense.

## Monitoring Network Traffic

Network traffic monitoring touches at least two major areas of concern to network admins, including both the *amount* and *type* of traffic traversing the network. Administrators must be constantly alert to potential bottlenecks within their networks, which can lead to substandard performance. They must also be on the lookout for potentially anomalous traffic that may be indicative of an attack, such as denial-of-service (DOS), data modification or exfiltration, brute-force login attempts, and others.

Network traffic monitoring can be accomplished using a variety of methods and techniques. In this section, we will focus on passive techniques.

**tcpdump** is a command-line tool that allows the user to display network packets being transmitted or received over a network to which a particular host is attached. tcpdump prints the contents of network packets, either those read in real time from a network interface card, or from a previously saved packet file. This command is available on a wide range of devices, including desktop and server systems, routers and firewalls. Running tcpdump on Unix/Linux-based systems often requires root privileges.

tcpdump supports a wide variety of options. A few common examples are given below.

To see a list of available network interfaces on which listening is possible:

$ sudo tcpdump -D

To listen on interface eth0:

$ sudo tcpdump -i eth0

To listen on any available interface (useful on routers or firewalls):

$ sudo tcpdump -i any

By default, tcpdump displays a bare summary of packet information. The level of detail (verbosity) can be increased by adding the -v option. Be verbose while capturing packets:

$ sudo tcpdump -v

Be more verbose while capturing packets (up to three vs can be used):

$ sudo tcpdump -vv

Be verbose and print the data of each packet in both hex and ASCII, excluding the link level header:

$ sudo tcpdump –vX

Running tcpdump from a network host, such as a desktop or server, can provide useful information regarding the traffic on a network. It can be even more informative to run packet captures on network routers and firewalls, since these devices are able to “see” all network traffic traversing their portions of the network. This makes it possible for a network administrator to get a better picture of all the hosts communicating on the network, as well as the volume of network traffic.

The VyOS router software is Linux-based, so tcpdump can be run from the command line in the same way as on any Linux host. When running tcpdump on any host by way of an SSH login session, it’s important to filter out the traffic generated by the login session itself.

So, for example, to capture network packets on a VyOS router via an SSH connection, displaying IP addresses and port numbers (rather than hostnames and port names), while filtering out traffic generated by the SSH login session (assuming the router’s SSH server is running on the default port):

$ tcpdump -nn port not ssh

For more information and examples see the tcpdump man page.

## Managing and Protecting a Network

Within the BetaPort environment, network admins use two types of network devices to configure and protect their networks: VyOS routers and pfSense firewalls. In this section, we’ll provide a brief overview of these two devices.

### VyOS

VyOS is a Linux-based network operating system that provides software-based network routing and other functionality. In BetaPort, VyOS is used exclusively for routing. VyOS routers are configured and controlled by way of a command line interface (CLI).

The VyOS CLI is comprised of an operational mode and a configuration mode. Operational mode allows for commands to perform operational system tasks and to view system and service status, while configuration mode allows for the modification of system configuration. The CLI provides a built-in help system. In the CLI the [?] key may be used to display available commands. The [tab] key can be used to auto-complete commands and will present the help system upon a conflict or unknown value.

Router configuration is necessary only when new networks are added or in the extremely rare instance when existing routes must be changed. Therefore, in this guide we’ll cover only a small number of operational mode commands.

Configured interfaces on a VyOS system can be displayed using the **show** **interfaces** command.

vyos@vyos:~$ show interfaces

Codes: S - State, L - Link, u - Up, D - Down, A - Admin Down

Interface IP Address S/L Description

--------- ---------- --- -----------

eth0 10.0.2.15/24 u/u Public

eth1 172.35.100.1/24 u/u LAN

lo 127.0.0.1/8 u/u ::1/128

A specific interface can be shown, providing greater detail, using the **show interfaces <type> <name>** command.

vyos@vyos:~$ show interfaces ethernet eth0

eth0: <BROADCAST,MULTICAST,UP,LOWER\_UP> mtu 1500 qdisc pfifo\_fast state UP qlen 1000

link/ether 00:0c:29:44:3b:0f brd ff:ff:ff:ff:ff:ff

inet 10.16.2.15/24 brd 10.0.2.255 scope global eth0

inet6 fe80::20c:29ff:fe44:3b0f/64 scope link

valid\_lft forever preferred\_lft forever

Description: OUTSIDE

RX: bytes packets errors dropped overrun mcast

274397 3064 0 0 0 0

TX: bytes packets errors dropped carrier collisions

257276 1890 0 0 0 0

Finally, the configured routes can be shown using the **show** **ip** **route** command.

vyos@vyos:~$ show ip route

Codes: K - kernel route, C - connected, S - static, R - RIP, O - OSPF I - ISIS, B - BGP, > - selected route, \* - FIB route

C>\* 10.0.2.0/24 is directly connected, eth0

C>\* 127.0.0.0/24 is directly connected, lo

C>\* 172.35.100.0/24 is directly connected, eth1

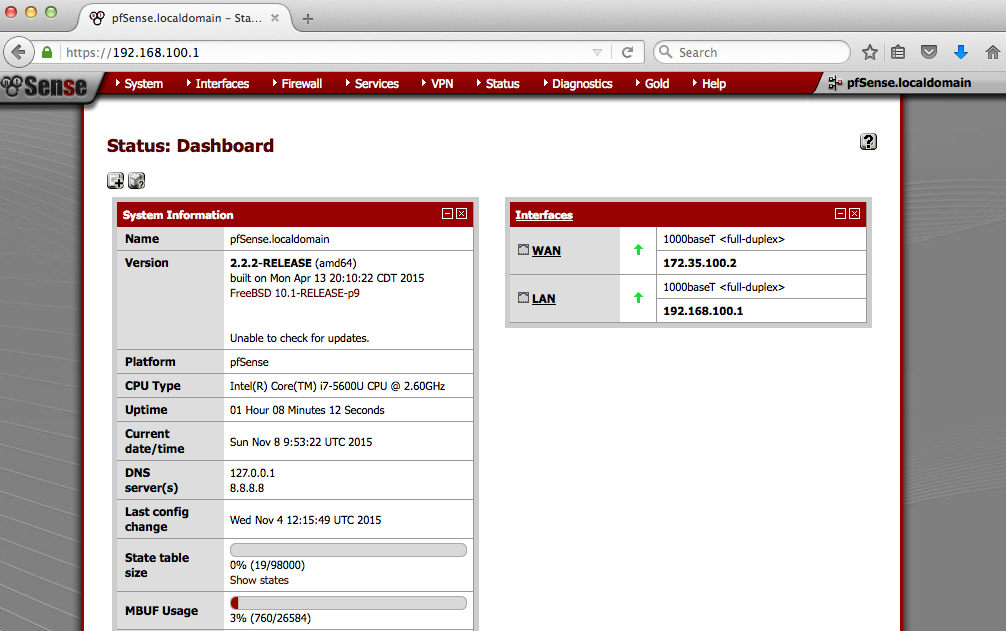
S>\* 192.68.100.0/24 [1/0] via 172.35.100.2, eth1

### pfSense

Behind each router (on the LAN side) within the BetaPort environment a **pfSense** firewall has been installed. pfSense is an open source firewall/router computer software distribution based on FreeBSD Unix. pfSense can be configured and operated through a web-based interface, and requires no knowledge of the underlying FreeBSD system to manage.

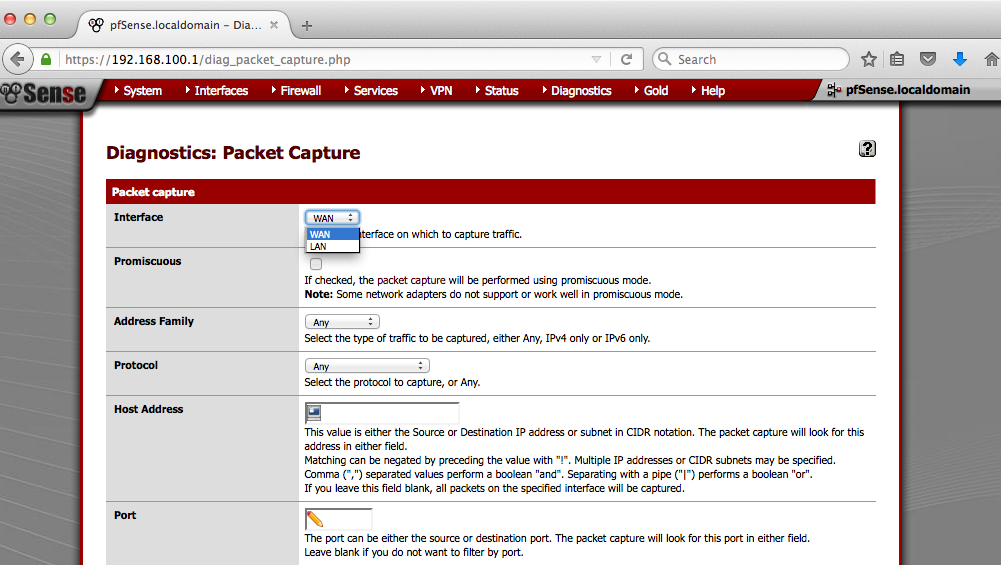
To access a pfSense firewall, type the IP address of the LAN-facing network adapter into the address bar of your Web browser. After logging in, the firewall’s menu-driven GUI interface is displayed. A portion of the firewall’s home screen is shown below. The pfSense navigation menu is located at the top, providing ready access to all of the firewall’s functionality.

Below the menu bar, a dashboard containing two panes is displayed, giving a high-level overview of the firewall’s status, including system information and a list of the active interfaces.



Among the most important features of the pfSense firewall for network administrators are the ability to monitor network activity by “sniffing” network traffic and tracking the activity of the most active network hosts.

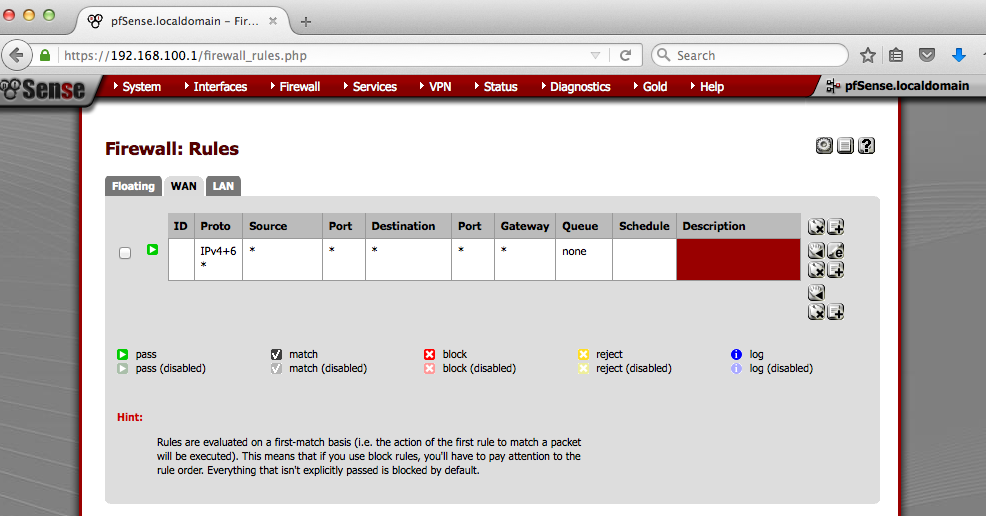
pfSense provides a packet-capture utility similar to the tcpdump command, described earlier. To activate this utility, select **Diagnostics -> Packet Capture** from the menu.



Finally, the pfTop utility can be used to monitor network traffic and connections in real time. You can access pfTop via the GUI menu, at Diagnostics -> pfTop.

For more information on capturing packets and monitoring network traffic using pfSense, see the pfSense Packet Captures document.

As its name implies, the pfSense firewall also provides the ability to control what kinds of traffic are allowed to traverse the network. Fine-grained rules can be set up, based on protocol, source and destination network addresses and ports, among other criteria. The screenshot below shows a firewall configured to allow all IPv4 and IPv6 to pass through the firewall in both (inbound and outbound) directions.



Firewall rules are managed at **Firewall -> Rules**. For more information on configuring firewall rules in pfSense, see the pfSense Firewall Rule Basics section below.

## pfSense Firewall Rule Basics

Firewall rules control what traffic is allowed to enter an interface on the firewall. Once traffic is passed on the interface it enters an entry in the state table is created. A state table entry allows through subsequent packets that are part of that connection.

Firewall rules on Interface and Group tabs process traffic in the Inbound direction and are processed from the top down, stopping at the first match. Where no user-configured firewall rules match, traffic is denied. Only what is explicitly allowed via firewall rules will be passed.

Firewall rules are managed at **Firewall -> Rules**. Multiple rules may be selected for some actions by clicking on their row or checking the box at the start of their row. Rules may be deleted or reordered in bulk in this way.

On the Firewall Rules page, there is a tab for each interface, plus a tab for each active VPN type (IPsec, OpenVPN, PPTP), and a tab for Floating Rules, which contains more advanced rules that apply to multiple interfaces and directions.

When editing a rule many of the options are explained in detail on the rule editor screen. Be mindful of the default settings on the rule editor, especially the protocol. New rules default to *TCP* only.

When entering addresses into firewall rules, the following choices are given for the source and destination addresses. Some of these options only appear in specific fields or circumstances, or if certain features are enabled.

* **any** - *0.0.0.0* to *255.255.255.255*, or all IPv6 addresses
* **Single host or alias** - Select this and enter one IP address (*1.2.3.4*, *aa:bb:cc:dd::1*) or type the name of an Alias that has already been configured (**Firewall > Aliases**)
* **Network** - Select this and enter a network and mask (*10.99.0.0/16*, *aa:bb:cc:dd::0/64*)
* **LAN net** - The subnet configured on the LAN interface under **Interfaces > LAN**. On pfSense 2.2+, this also includes static route networks on that interface.
* **LAN address** - The IP address configured on the LAN interface under **Interfaces > LAN**
* **zzz Net** / **zzz address** - Works the same as LAN above but for other interfaces (WAN, OPT1, OPT2, etc.)
* **PPTP clients** - Automatically locate and use the addresses of PPTP clients
* **L2TP clients** - Automatically locate and use the addresses of L2TP clients
* **This Firewall (self)** - Any IP address assigned to any interface on this firewall (pfSense 2.2+)

These macros are handy because they allow generic rules to be created that refer to LAN or a specific interface. If that interface IP address or subnet changes in the future, the rules will be rebuilt correctly and they will not need manually adjusted.

## pfSense Packet Captures

### WebGUI Packet Captures

A packet capture may be performed within the pfSense GUI interface under **Diagnostics > Packet Capture**. The settings work the same as tcpdump. The capture can be viewed in the GUI or downloaded for later viewing with tcpdump or Wireshark.

Various filters may be added to restrict the scope of the capture, such as a specific **Protocol**, **Host address**, or **Port** (among others). The size of the capture may be adjusted as well. Often a few thousand packets are necessary to catch certain activity.

The **Level of detail** selector only controls the level of detail displayed in the pfSense GUI for viewing the contents of a capture. It may be adjusted after a capture has been taken, to view the capture with more detail, adjust this value and click **View Capture**.

Click **Start** to start a capture. While a capture is running, a **Stop** button is also displayed to stop a capture in progress.

**View Capture** shows the contents of the previous capture.

**Download Capture** initiates a download of the capture file for viewing locally (or sending to a remote technician.)

### tcpdump

tcpdump also comes installed with pfSense. It can be used over SSH or on the console in a shell. In this example a method of capturing traffic other than SSH, ARP, DNS and STP is highlighted. The capture will be directed to a file called Sniff\_output in the current directory.

$ tcpdump -i em0 not port 22 and not port 53 and not arp and not stp >> Sniff\_output

The -i is designating traffic from the em0 interface. In this example traffic from one of the subnets em0 connected to pfSense is being grabbed.

### pftop

pftop is a tool built into pfSense that can monitor traffic/connections. This tool can be found in the GUI under **Diagnostics > pftop** or by connecting to pfSense via SSH or the console.

## Network Admin Tools/Skills

### General use tools/concepts

Logging in with SSH

sudo

cat, head, tail, less

grep, grep -v

sort

du

ps -ef

netstat -atun

emacs

Using pipes with above commands

### Specific tools & applications

passwd (To change user passwords)

adduser, chpasswd (To set up user accounts)

mount (File systems attached layout of system)

dmesg (System status)

who (who is logged in)

htop (alternative to top with a better layout)

addgroup (creation of a new group) and how to added a user to a group

ifconfig

route -n

apt-get

service <service name> <start|stop|status>

pfsense (web interface)

VyOS commands

show interfaces

### Log files

/var/log/syslog

/var/log/auth.log

# Email Admin Familiarization

## Overview

Email services within BetaPort are provided through the SquirrelMail webmail package. Of course, Squirrelmail does not work on its own. It relies on several other packages to provide Web service and a variety of email services:

**Apache Web server** provides the Web services that Squirrelmail relies upon.

**Postfix** is a mail transfer agent (MTA) package that provides email services, specifically the Simple Mail Transfer Protocol (SMTP). This means that Postfix handles incoming and outgoing email services.

**Dovecot** is an Internet Message Access Protocol (IMAP) and Post Office Protocol, version 3 (POP3) server package that allows email clients to retrieve messages from a mail server over a TCP/IP connection.

In summary, **SquirrelMail** is a Web application that runs on top of the **Apache** Web server to provide easy and convenient access to email services provided by **Postfix** and **Dovecot**.

It’s easy to see each of these packages at work (and to ensure that the services are running and listening for connections) in support of SquirrelMail by running the netstat command. The following (excerpted) example uses options atnp, which (left to right) specify that **a**ll network services using the **t**cp protocol should be shown using **n**umeric IP addresses only, along with their associated **p**rocesses. Note that master refers to a Postfix process.

$ sudo netstat -atnp

Active Internet connections (servers and established)

Proto Recv-Q Send-Q Local Address Foreign Address State PID/Program name

tcp 0 0 0.0.0.0:25 0.0.0.0:\* LISTEN 1017/master

tcp 0 0 0.0.0.0:993 0.0.0.0:\* LISTEN 880/dovecot

tcp 0 0 0.0.0.0:995 0.0.0.0:\* LISTEN 880/dovecot

tcp 0 0 0.0.0.0:110 0.0.0.0:\* LISTEN 880/dovecot

tcp 0 0 0.0.0.0:143 0.0.0.0:\* LISTEN 880/dovecot

tcp6 0 0 :::25 :::\* LISTEN 1017/master

tcp6 0 0 :::993 :::\* LISTEN 880/dovecot

tcp6 0 0 :::995 :::\* LISTEN 880/dovecot

tcp6 0 0 :::110 :::\* LISTEN 880/dovecot

tcp6 0 0 :::143 :::\* LISTEN 880/dovecot

tcp6 0 0 :::80 :::\* LISTEN 1143/apache2

A brief overview of the configuration and layout of SquirrelMail and each of the packages that supports it will be provided in the sections below.

## SquirrelMail

As described above, SquirrelMail is a webmail package, specifically a Web application that runs on Apache Web servers. As such, SquirrelMail is simply a collection of server-side scripts and text-based configuration files.

### Starting and Stopping

Since SquirrelMail is an application, rather than a service, there isn’t a specific command available to start and/or stop it. It is, however, possible to disable and later re-enable a SquirrelMail Web site using a couple of Apache commands:

To temporarily disable SquirrelMail:

$ sudo a2dissite squirrelmail

Site squirrelmail disabled.

To activate the new configuration, you need to run:

$ sudo service apache2 reload

\* Reloading web server apache2

To re-enable SquirrelMail:

$ sudo a2ensite squirrelmail

Site squirrelmail disabled.

To activate the new configuration, you need to run:

$ sudo service apache2 reload

\* Reloading web server apache2

***Important note:***Since this process requires reloading the entire Web server, it is ***extremely*** important to coordinate this activity ahead of time with your Web Administrator and Help Desk!

### Configuration and Layout

As a webmail application, SquirrelMail has three different types of configuration files: 1) General configuration settings which affect the behavior of the entire application, 2) Individual user configuration settings, which affect the application’s behavior on an individual user basis, and 3) Apache Web server configuration settings, which govern the application’s behavior as a Web service. The first two types of configuration will be discussed here, and the Apache configuration will be discussed in the Apache section that follows.

On Ubuntu servers, general configuration settings are contained in several files located in the /etc/squirrelmail directory. The most important of these files is config.pl, which is actually a perl language script that provides a simple menu-driven application for setting and saving configuration settings.

The configuration settings for individual users are stored in per-user files in the directory /var/lib/squirrelmail/data/ with the extension .pref. Address book data for individual users is also stored in this directory, in files with the extension .abook.

### Creating and managing user accounts

SquirrelMail user accounts are actually created and managed as normal user accounts on the email server itself. Therefore, the commands to add, modify and remove email accounts are the same as those to add, modify and remove system accounts. User Management Commands in the Supporting Documents section for details on how to use these commands.

## Apache

Since configuration and management of Apache Web server is the responsibility of the Web Administrator, the discussion in this section will be limited to those aspects that directly affect the SquirrelMail application. For more information on Apache, see the Web Admin Familiarization activity.

### Starting and Stopping

As discussed in the previous section, enabling or disabling the SquirrelMail application also requires that the Apache Web server be restarted or reloaded. The command for reloading Apache is shown again here for your convenience:

$ sudo service apache2 reload

\* Reloading web server apache2

***Important note:*** Once again, it is ***extremely*** important to coordinate this activity with your organization’s Web Administrator and Help Desk personnel, as restarting the server is very likely to affect several other important services.

### Configuration and Layout

Like all Apache Web sites and applications on Ubuntu, configuration data for the SquirrelMail application is stored in a directory named /etc/apache2/sites-available. By default, SquirrelMail’s configuration is stored in a file named squirrelmail.conf. Among the most important settings is the document root, by default, /usr/share/squirrelmail/. All the SquirrelMail server-side scripts that control the application’s behavior are stored in this directory. Other settings include directory index settings and access restrictions.

## Postfix

Postfix is a mail transfer agent (MTA) that routes and delivers electronic mail. Thus, it is the software responsible for receiving email messages for SquirrelMail users from other servers and for routing the messages they send to the destination email servers.

### Starting and Stopping

Depending on its configuration, a Postfix server may actually have three or more processes running at any given time. But, regardless of how many processes it comprises, the Postfix server can be stopped, started and restarted using the following commands.

$ sudo /etc/init.d/postfix stop

$ sudo /etc/init.d/postfix start

$ sudo /etc/init.d/postfix restart

Rather than stopping/starting or restarting the server, it is sometimes preferable to simply force Postfix to reload its configuration. This step is required following any configuration changes and must be done before changes take effect. To force Postfix to reload it’s configuration, use the following command.

$ sudo service postfix reload

### Configuration and layout

On Ubuntu systems, the Postfix configuration files are located in the directory /etc/postfix. The two primary configuration files are main.cf and master.cf. The main.cf file stores site-specific Postfix configuration parameters while master.cf defines the daemon processes. Once a mail server has been set up, there is very seldom any reason to change its configuration.

Postfix log messages are written to /var/log/mail.log. Reviewing log messages is an important duty of any server administrator, as log files record all server transactions and can often provide information that’s useful in determining the root cause of any unusual behavior. For information and tips on searching and view log files, see View/Search Text Files in the Supporting Documents section.

## Dovecot

Dovecot is an IMAP and POP3 server package that allows email clients, including Web-based clients like SquirrelMail, to retrieve messages from a mail server over a TCP/IP connection.

### Starting and Stopping

Like Postfix, the Dovecot server typically has several processes running. All associated processes can be stopped, started or restarted using the following commands.

$ sudo service dovecot stop

$ sudo service dovecot start

$ sudo service dovecot restart

Like any other server system, Dovecot must be restarted after any configuration changes before those changes take effect.

### Configuration and layout

On Ubuntu systems, the Dovecot configuration files are located in the directory /etc/dovecot. The main configuration file is dovecot.conf, which rarely requires any changes following installation. In addition, the directory /etc/dovecot/conf.d holds numerous other configuration files. Again, after a successful installation, configuration changes are rarely, if ever, required.

Dovecot log messages are written to the general log, prefaced with the service name dovecot, rather than to a dedicated log file. On Ubuntu systems, general log messages are written to /var/log/syslog. To view all messages in /var/log/syslog associated with Dovecot services, use a command like the following.

$ grep dovecot /var/log/syslog

For more information and tips on searching and view log files, see View/Search Text Files in the Supporting Documents section.

## Email Admin Tools/Skills

### General use tools/concepts

Logging in with ssh

sudo

cat, head, tail, less

grep, grep -v

sort

du

ps -ef

netstat -atun

emacs

Using pipes with above commands

### Specific tools & applications

Apache Web server

Squirrelmail

Prosody

Postfix

Dovecot

passwd

adduser

chpasswd

### Log files

/var/log/mail.log

/var/log/apache2/access.log

/var/log/apache2/error.log

## Helpdesk Admin Tools/Skills

### General use tools/concepts

Logging in with ssh

sudo

cat, head, tail, less

grep, grep -v

sort

du

ps -ef

netstat -atun

emacs

Using pipes with above commands

### Specific tools & applications

OTRS ticketing system

who

htop

ifconfig

ping

# System Admin Familiarization

## Overview

A system administrator, or *sysadmin*, is responsible for the configuration and reliable operation of computer systems, especially multi-user systems such as servers. The system administrator must ensure that uptime, performance, resources, and security of all systems meet the needs of users. To meet these responsibilities, a system administrator is responsible for acquiring, installing and upgrading computer components and software; providing routine automation; maintaining security policies; troubleshooting; and training and supervising support staff.

Fortunately, system administrators have a variety of tools and applications at their disposal to help them meet their myriad responsibilities. In the sections below, the various responsibilities of the position are organized into broad categories. Within each category, you’ll find a brief discussion of the tools and applications available to meet a sysadmin’s needs.

## Managing System Services

By definition, server systems typically provide one or more network services, such as Web, email, chat, file, database and others. In many cases, the responsibility of managing these and other services may be split among sysadmins and one or more other specialty admins, such as Web server, email, or database admins. In these cases, close cooperation and constant communication among the admins is critical to ensure maximum availability, reliability and security of system services.

Each service is provided through one or more server programs called daemons, programs that run as background processes, rather than being under the direct control of an interactive user. In most cases, daemons are controlled through scripts that use consistent interfaces.

To view all services running on a system, several commands are available. For example, to see a list of all running processes, along with process ID (PID) numbers, start time, process name and other information, use the command:

$ sudo ps -ef

Alternatively, the top and htop commands provide a dynamic, real-time view of system summary information and a list of running tasks. Processes are, by default, displayed in order by their consumption of resources (CPU and memory), from greatest to least. The PID, the user associated with the process, and other information is also displayed. To display this information using default parameters, simply run top or htop (a more user-friendly version of top) at the command line.

While nearly all system services are configured, by default, to automatically start and stop when the system is booted and shut down, it is occasionally necessary to start, stop or restart a running service. For example, configuration changes or updates to a service nearly always require a restart. As mentioned above, most daemons are controlled through scripts that use common interfaces to start, stop or restart the service, or to force it to reload its configuration. It also usually possible to check a service’s status in this way.

For example, the Apache Web server is one of a very large number of services controlled through the service command. To start, stop, restart or reload Apache, use commands like the following:

$ sudo service apache2 start

$ sudo service apache2 stop

$ sudo service apache2 restart

$ sudo service apache2 reload

You can also check a service’s current status by using the service command with the status argument. For example:

$ sudo service apache2 status

Apache2 is running (pid 1583).

For more information on these and similar commands, see System Monitoring Commands in the Supporting Documents section.

## Configuration and Layout

On Ubuntu systems, system configuration settings for the vast majority of applications and services are contained in text files located in a top-level directory named /etc. Relatively small or simple applications such as updatedb, which automatically updates a database used by the locate command, are configured through a single file in /etc with the extension .conf. Other applications such as Apache Web server and Postfix email system have their own subdirectories within /etc containing multiple configuration files. The job of maintaining these more complex applications usually falls to a specialist administrator, such as a Web or email admin. However, as mentioned above, close coordination and cooperation between the various administrators working on a server system is a must.

In addition to configuration files, most services have their own log files, where they regularly record a variety of messages regarding routine transactions and, more importantly, messages related to execution errors or other problems. On Linux systems, these log files are found in the directory /var/log. As was the case with configuration files, the type and number of log files frequently depends on the size and/or complexity of the application. The more simple and common applications typically write their log messages to the file /var/log/syslog. More complex services, such as the Apache Web or Postfix email servers have their own files or even their own directories within /var/log. Again, the job of maintaining these more complex applications usually falls to a specialist administrator but, as always, coordination is key.

For more information and tips on searching and view configuration and log files, see View/Search Text Files in the Supporting Documents section. For information on text editing, see Help and Editor Commands, also in the Supporting Documents section.

## Creating and managing user accounts

Creating and managing user accounts is a core responsibility for many system administrators. Sysadmins are regularly called upon to create, modify and remove user accounts or to change users’ passwords. These tasks are carried out using commands such as adduser, deluser and passwd. See User Management Commands in the Supporting Documents section for details.

In addition, sysadmins are responsible for managing file and directory permissions to ensure that only authorized users have the appropriate access rights to view or edit files or execute commands. See File Permission Management Commands in the Supporting Documents section for information on the commands used to set and modify file access permissions.

## System Admin Tools/Skills

### General use tools/concepts

Logging in with ssh

sudo

cat, head, tail, less

grep, grep -v

sort

du

ps -ef

netstat -atun

emacs

Using pipes with above commands

### Specific tools & applications

passwd

chpasswd

adduser

addgroup

usermod

mount

dmesg

who

htop

addgroup

ifconfig

route -n

apt-get

service

df

### Log files

/var/log/syslog

/var/log/auth.log

# Chat Admin Familiarization

## Overview

Chat services, also called Extensible Messaging and Presence Protocol (XMPP) services, are provided in BetaPort through the Prosody package. Prosody is a small, lightweight implementation of the XMPP application protocol that is easy to configure and manage.

Prosody supports both client-to-server and server-to-server communications. This means that users who have an account on a Prosody server can easily communicate with users who have accounts on other XMPP servers just as easily as they can with users having accounts on the same server. By default, Prosody listens for client connections on port 5222 and server connects on port 5269.

It’s easy to confirm that Prosody is listening on the correct ports by running the netstat command. The following (excerpted) example uses options -atnp, which (left to right) specify that **a**ll network services using the **t**cp protocol should be shown using **n**umeric IP addresses only, along with their associated **p**rocesses. Note that lua5.1 refers to a Prosody process. This is due to the fact that Prosody is written in Lua, an embedded scripting language that is often used in game programming because of its speed.

$ sudo netstat -atnp

Active Internet connections (servers and established)

Proto Recv-Q Send-Q Local Address Foreign Address State PID/Program name

tcp 0 0 0.0.0.0:5269 0.0.0.0:\* LISTEN 1046/lua5.1

tcp 0 0 0.0.0.0:22 0.0.0.0:\* LISTEN 878/sshd

tcp 0 0 0.0.0.0:5222 0.0.0.0:\* LISTEN 1046/lua5.1

tcp6 0 0 :::5269 :::\* LISTEN 1046/lua5.1

tcp6 0 0 :::22 :::\* LISTEN 878/sshd

tcp6 0 0 :::5222 :::\* LISTEN 1046/lua5.1

XMPP services support a wide variety of clients. The standard client for BetaPort users is Pidgin Internet Messenger. For more information on Pidgin, see Pidgin Instructions in the Supporting Documents section.

### Starting and Stopping

While Prosody should seldom ever require restarting, it’s important to know how to stop, start, restart and reload the service.

To stop Prosody:

$ sudo service prosody stop

\* Stopping Prosody XMPP Server prosody [ OK ]

To start Prosody:

$ sudo service prosody start

\* Starting Prosody XMPP Server prosody [ OK ]

To restart Prosody:

$ sudo service prosody restart

\* Restarting Prosody XMPP Server prosody [ OK ]

To force Prosody to reload its configuration, such as after a configuration change:

$ sudo service prosody reload

\* Reloading Prosody XMPP Server prosody [ OK ]

### Configuration and Layout

On Ubuntu systems, the Prosody configuration files are located in the directory /etc/prosody. The primary configuration file is prosody.cfg.

Prosody log messages are written to the directory /var/log/prosody, into two separate log files: prosody.log and prosody.err. Reviewing log messages is an important duty of any server administrator, as log files record all server transactions and can often provide information that’s useful in determining the root cause of any unusual behavior. For information and tips on searching and view log files, see View/Search Text Files in the Supporting Documents section.

### Creating and managing user accounts

Prosody user accounts are created and managed using a command named prosodyctl. This command requires sudo privileges, which restricts its use to those system users having administrative privileges.

To create a new Prosody account and specify the user’s password in one command, use prosodyctl with the register command, followed by the new username, the server’s fully-qualified domain name, and the new user’s password. For example:

$ sudo prosodyctl register ella elko.26maidenlane.net afwtl7j4

To change an existing user’s password, use prosodyctl with the passwd command, followed by the user’s jid. The jid looks just like an email address, and it consists of the username, followed by ‘@’, followed by the server’s fully qualified domain name. For example, to change the password for ella, whose account was created above, use the following command line, after which you would be prompted for the new password.

$ sudo prosodyctl passwd ella@elko.26maidenlane.net

Finally, to remove a user account from a Prosody server, use prosodyctl with the deluser command, followed by the user’s jid. For example, to remove the user account ella, created above, use the following command line.

$ sudo prosodyctl deluser ella@elko.26maidenlane.net

# Web Admin Familiarization

## Familiarization Activity

1. It is essential to familiarize yourself as quickly as possible to the web services your company provides. It is also essential for you to develop a strong sense of the availability of these services so that you are better positioned to prevent, detect and respond to unusual activity that may disrupt company operations.
2. Start by determining which web pages and web-based services are provided by your company. Be sure to distinguish between those that are serving the consumers/public versus those for internal/company use. Use the following space to record your notes.
3. Visit each of the web pages and each of web pages to which they are linked. Familiarize yourself with both their content and availability. Pay particular attention to public facing web pages as maintenance of their availability is critical to company viability. Use the following space to record your notes.
4. Determine which web-based services are provided by your company. Be sure to distinguish between those that are serving the general public versus those for internal/company use. Use the following space to record your notes.
5. Review the web logs (especially the *access* and *error* log files). Familiarize yourself with “normal” traffic patterns. Remember to communicate with your web administrator colleagues and seek their input as you develop your own sense of “normal”. Use the following space to record your notes.
6. Review the training materials that have been provided. These materials include descriptions of the tools/commands useful to web administrators. Pay particular attention to tools/commands used to configure and manage the Apache-based web services. Use the following space to record your notes.

## Apache Restart Activity

One task that a web administrator must sometimes perform is a restart of the web server. Be aware that restarting web services should be performed only when necessary as it temporarily prevents users from accessing web pages and web-based services. The image and viability of the company can suffer if web pages and/or web-based services are taken off-line or otherwise made unavailable.

This activity presumes you have already completed the Familiarization Activity (above).

1. Notify the Help Desk of your intention to restart the web server and the reason for doing so. It is essential to document the rationale and reasons for needing to restart the web server.
2. Login to your web server using ssh or telnet and make sure your system includes the init.d command (usually placed in the /etc directory). Confirm the presence of the file by issuing the following command:

$ ls /etc/i\*

1. Issue the following command:

$ sudo /etc/init.d/apache reload

If the above command does not achieve the desired outcome then use the more disruptive *restart* command:

$ sudo /etc/init.d/apache restart

1. Ensure Apache server has restarted and web services restored. Review the *error* log, confirm public and internal accessibility to corresponding web pages and web-based services.

$ cat /var/log/apache2/error.log

1. Report to the Help Desk that all web services have been restored.

## Web Site Admin Tools/Skills

### General use tools/concepts

Logging in with ssh

sudo

cat, head, tail, less

grep, grep -v

sort

ps -ef

netstat -atun

emacs

Using pipes with above commands

### Specific tools & applications

Apache Web server

chown

chmod

### Log files

/var/log/mail.log

/var/log/apache2/access.log

/var/log/apache2/error.log

IT Staff Job Descriptions

# System Administrator Job Description

## Summary

Responsible for designing, organizing, modifying, and supporting a company's computer systems. Designs and installs LANs, WANs, Internet and intranet systems, and network segments.

## Essential Duties and Responsibilities

* Support LANs, WANs, network segments, Internet, and intranet systems.
* Respond promptly to helpdesk tickets and status queries.
* Maintain system efficiency.
* Troubleshoot problems reported by users.
* Maintain network and system security.
* Analyze and isolate issues.
* Monitor server and desktop systems to ensure security and availability.
* Evaluate and modify system's performance.
* Maintain network servers such as (web, file, email, chat) servers

# Network Administrator Job Description

## Summary

Responsible for designing, organizing, modifying, installing, and supporting a company's computer systems. Designs and installs LANs, WANs, Internet and intranet systems, and network segments.

## Essential Duties and Responsibilities

* Install and support LANs, WANs, network segments, Internet, and intranet systems.
* Install and maintain network hardware and software.
* Monitor and analyze network activity to ensure security and availability.
* Maintain integrity of the network, server deployment, and security.
* Ensure network connectivity throughout a company's LAN/WAN infrastructure is on par with technical considerations.
* Perform network address assignment.
* Respond promptly to helpdesk tickets and status queries.
* Assign routing protocols and routing table configuration.
* Maintain network services such as gateways and firewalls.

### 

# Help Desk Specialist Job Description

## Summary

Responsible for providing technical assistance and support related to computer systems, hardware, or software. Responds to queries, runs diagnostic programs, isolates problem, and determines and implements solution.

## Essential Duties and Responsibilities

* Provide technical assistance and support for incoming queries and issues related to computer systems, networks, software, and hardware.
* Respond to queries either in person, in writing (includes email and chat) or over the phone.
* Write training manuals.
* Ask questions to determine nature of reported problems in support of quick and correct resolution.
* Walk customer through problem-solving process.
* Act as point of contact for all users to the IT department.
* Track all raised issues and be in the loop on efforts to resolve even if not actively coordinating (the helpdesk should never loose touch with the problem or the customer).
* Act as IT eyes and ears for problems affecting multiple users.
* Respond promptly to status queries from other IT areas.

# Webmaster Job Description

## Summary

Responsible for maintaining Web sites. Ensures sites are functioning properly and are available to users. Tests speed of access and improves upon loading speed.

## Essential Duties and Responsibilities

* Maintain websites for clients and businesses.
* Ensure the web servers, and supporting software are operating properly.
* Design websites.
* Generate and revise web pages.
* Examine and analyze site traffic.
* Configure web servers such as Apache.
* Serve as the backup server administrator on the web server.
* Regulate and manage access rights of different users on website.
* Create and modify appearance and setting of site.
* Respond promptly to helpdesk tickets and status queries.
* Keep the helpdesk informed of any and all issues as the website is the organizations face to the world

# Email Administrator Job Description

## Summary

Responsible for mail server account management, tracking mail server usage, and backing up and restoring email files. Installs and configures new mail servers.

## Essential Duties and Responsibilities

* Create and maintain email accounts
* Reset user passwords, as required
* Assist with login and email client problems
* Monitor and respond to Postmaster (error) messages
* Track mail server usage
* Monitor mail server connections
* Track disk space usage
* Monitor mail server logs
* Control incoming and outgoing spam mailings

Supporting Documents

# Basic Connectivity Testing Tools

## ping

ping is one of the basic tools.  It sends repeated packets (ICMP rather than TCP or UDP) to the specified <host>.

$ ping [options] <host>

The list of available options can be found using man pages.  The most useful option is -c <count> (replace <count> with the number of pings to send). This can be critical, because ping can also be used as a denial-of-service (DOS) attack and, yes, it is possible to DOS yourself with ping.  No, you would not be the first.  It is important to understand that ping only tells you whether the server is up, not that any services are running.  As an example, if you ping a Web server, you know the hardware and OS are running, but not that the Web server itself is necessarily running.

## telnet

telnet is a very old and useful tool, because it is very generic.

$ telnet <host>  <port>

By default, the telnet service runs on port 23 so if you use a command line like the following.

$ telnet www.somehost.com

The connection will be made on port 23, which is the default for the telnet service.  Due to security concerns the telnet service is often disabled.  However, it is also possible to use the telnet client to connect to ports other than the default. For example, if you use a command line like this one.

$ telnet www.somehost.com 80

The connection in this case will be made on port 80 (typically used by a Web server) at the destination host, and you can enter HTTP commands to interact with the host’s Web server, if there is one listening on that port. In this way, it’s possible to show whether the Web server running on port 80 is up and running. In like fashion, the telnet command can be used to connect to other services, such as SMTP, on port 25.

## traceroute

traceroute attempts to show the hops between hosts on the Internet. It is critical to note that traceroute only shows the interface facing the source host as packets pass through routers.

$ traceroute <host>

### Example:

Here is an example of running traceroute between two hosts (192.168.1.7 to 172.16.1.2) with a single router in between. Thus the 192.168.1.7 machine sees the following.

$ traceroute 172.16.1.2

traceroute to 172.16.1.2 (172.16.1.2), 64 hops max, 52 byte packets

1  192.168.1.1 (192.168.1.1)  3.877 ms  0.995 ms  0.902 ms

2  172.16.1.2 (172.16.1.2)  2.631 ms  1.318 ms  1.236 ms

While, from the other end of the network connection, 172.16.1.2 sees the following.

$ traceroute 192.168.1.7

traceroute to 192.168.1.7 (192.168.1.7), 64 hops max, 52 byte packets

1  172.16.1.101 (172.16.1.101)  0.552 ms  0.904 ms  1.265 ms

2  192.168.1.7 (192.168.1.7)  18.025 ms  18.673 ms  19.584 ms

So to build an accurate network map you need to check both sides of a router, as is the case if you are looking for a network fault.  In the case of a fault, you often have to get somebody on the other end to check the routes running back in your direction.

# IT Help Desk Notes & Flowchart

## Create Ticket

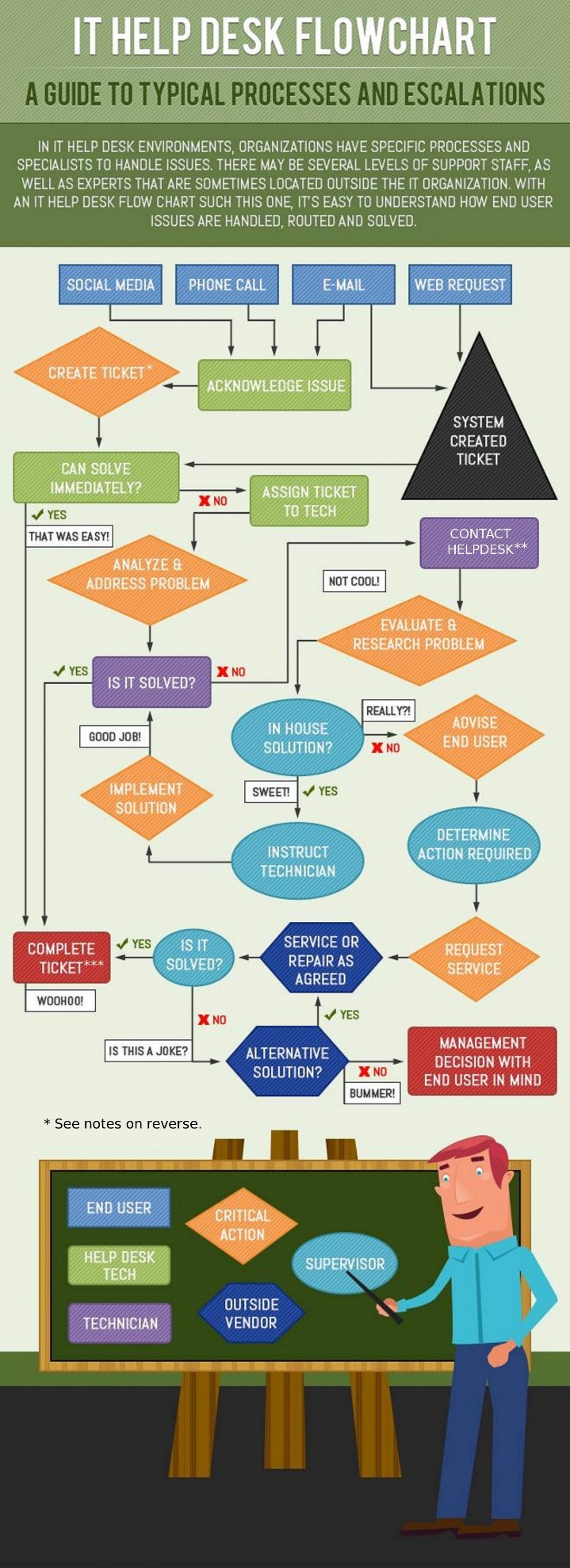
Upon ticket creation, the helpdesk technician should email the user and any other interested parties to insure that everyone connected to the issue is aware of its status.

## Contact Helpdesk

When a technician assigned a ticket determines that the issue cannot be resolved, either because of technical reasons or because the issue cannot be duplicated, he or she should immediately contact the helpdesk to help determine next steps in addressing the issue.

## Complete Ticket

Upon completing the helpdesk ticket, the technician who closes it should email the helpdesk, the user and any other interested party to keep everyone informed.



# OTRS quick guide (v 0.1.0\_10/4/2015)

## To Log In

Go to:

http://helpdesk.[company\_name].com/otrs/    (yes you need the trailing /  )

Enter your login credentials.

## To Create Customer User

After logging in

1. Click “Admin” in the header menu
2. In the “Customer Management” section click “Customer User”
3. Click “Add Customer User” button in left sidebar
4. Fill in the following form fields.
   1. Firstname
   2. Lastname
   3. Username   (Customer will use this to log in.)
   4. Email
   5. CustomerID    (Customer’s tickets will be marked with this ID.)
5. Set “Valid” to “Valid” if it is not already.
6. Click “Submit” button at bottom of page.

## Tickets

*Note*: OTRS system tickets cannot be deleted, only closed.

Clicking the “Tickets” header button displays a list of possible views and actions. The most useful of these will be the queue or status view, depending on which you prefer. Both display lists of active tickets; queue view separates them into separate pages by which queue they are in, while status does not. Queue view defaults to the most detailed listing mode, and Status view defaults to the simplest mode.

### To Create Ticket

After logging in:

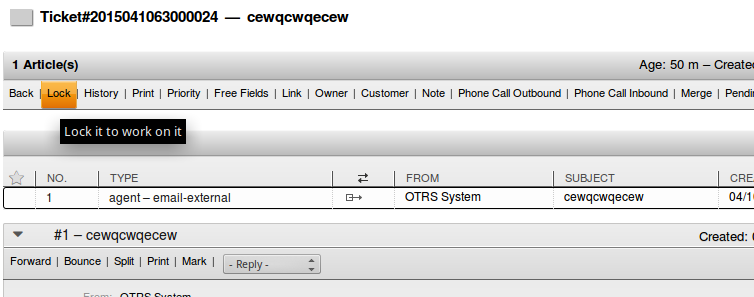
1. Click “Tickets” button in header menu
2. Click “new email ticket” in the dropdown menu
3. Fill in the form:
   1. Select queue from drop down menu
   2. Enter customer user (This does have some autocomplete.)
   3. Enter subject
   4. Enter text

All other fields are not required

1. Click “send mail” button at bottom of page

### To Lock A Ticket To Your Account

Either check a ticket in one of the list views or select it, then click the “Lock” button, as shown below.

.

### https://lh3.googleusercontent.com/9c5pzXD9zJQxZ9ABeWKJhCedzmmvWKWGgSnO0E2zHO4-mdUyxOxeU60kIjm3BinTnCnyaFNWghJNT5nSW3M26QLZIN-PvxUA3ApIAfr-l0dF6JfLeEp3y5dDvt3pthnJFqg=s1600 TO CLOSE A TICKET

Same as locking a ticket except you click the “Close” button instead of “Lock.” This button in the same row, farther to the right, as shown in the above screenshot.

# View/Search Text Files

## head

The head command displays the beginning of a file. The format of the head command is:

$ head *<filename>*

By default, you can only read the first ten lines of a file. You can change the number of lines displayed by specifying a number option.

$ head -20 <*filename*>

The above command would display the first 20 lines of a file named <*filename>*.

## tail

The reverse of head is tail. Using tail, you can view the last ten lines of a file. You can also change the number of lines displayed by specifying a number option. This can be particularly useful for viewing the most recent lines of a log file for important system messages.

$ tail -15 /var/log/auth.log

### Example:

You can use tail to watch log files as they are being updated. Using the -f option, tail automatically prints new messages from an open file to the screen in real-time. For example, to actively watch /var/log/syslog, enter the following at a shell prompt:

$ tail -f /var/log/syslog

Press Ctrl-c when you are finished with automated updating.

## cat

The cat command is a versatile utility. It can be used to view text, to create text files, and to join files. Its name is short for *concatenate*, which means to combine files.

Entering the cat command followed by a file name displays the entire contents of the file on the screen. If the file is long, the contents scroll off the screen. In this case, you may want to use the less command instead, described below.

## less

less is a program that lets you view text files. This is very handy since many of the files used to control and configure Linux are human readable.

less is invoked by simply typing:

$ less *<filename>*

### Controlling less

Once started, less will display the text file one page at a time. You may use the Page Up and Page Down keys to move through the text file. To exit less, type "q". Here are some commands that less will accept.

|  |  |
| --- | --- |
| Page Up or b | Scroll back one page |
| Page Down or space | Scroll forward one page |
| G | Go to the end of the text file |
| 1G | Go to the beginning of the text file |
| /*characters* | Search forward in the text file for an occurrence of *characters* |
| n | Repeat the previous search |
| h | Display a complete list less commands and options |
| q | Quit |

## grep

The grep command is useful for finding specific character strings in a file. For example, to find every reference made to the text "pattern" in the file <filename>, enter:

$ grep *<pattern> <filename>*

Each line in *<filename>* that includes the text *<pattern>* is located and displayed on the screen.

The grep command accepts a number of options that will change its behavior. A small selection of these is shown below. For more information, see the man page for grep.

|  |  |
| --- | --- |
| -v | Invert the sense of matching, to select *non-matching* lines for the given search criteria. |
| -A NUM | Print NUM  lines  of  trailing  context  after  matching  lines. |
| -B NUM | Print NUM  lines  of  leading  context  before  matching  lines. |

### Example:

To search an Apache Web server’s access log for entries pertaining to a particular client IP address or address range, you can use all or part of the address as a keyword for the search:

$ grep 10.0.1.212 /var/log/apache2/access.log

## sort

The sort command is used to sort/order lines in text files. You can sort the data in a text file and display the output on the screen, or redirect it to a file.

Based on your requirements, sort provides several command line options for sorting data in a text file. For example, to sort the lines of a text file *<filename>* in reverse order, use the -r option:

$ sort -r *<filename>*

In addition, files may sometimes contain duplicate entries that are not important to the goals of the user. To sort the lines of a text file *<filename>* while removing duplicate lines, use the -u (for unique) option:

$ sort -u *<filename>*

Finally, when sorting on numeric values, best results are obtained by using the -n option, which sorts by numeric value, rather than the alphabetical order of digits.

$ sort -n *<filename>*

### Examples:

It is sometimes desirable to sort the lines of a file based on some characters that are not at the beginning of a file. Entries in many log files are highly structured so that it is possible to specify sorting on a specific field. As an example, /var/log/syslog records log messages from a wide variety of services, the names of which are given in the fifth field of each log entry. To sort log entries based on the name of the service, located in field #5, you would use the -k option to specify sorting on this field:

$ sort -k 5 /var/log/syslog

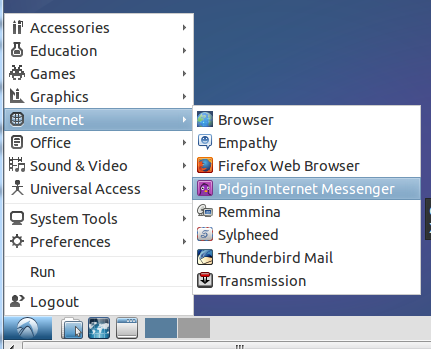
The sort command is often used in combination with other Linux commands, which are joined using a pipe. The pipe takes the output of a command on its left and redirects that output as input to the command on its right.

For example, say you would like to get a sorted list of IP addresses that have connected to your email server’s SMTP service for which reverse DNS lookups have failed. (This may indicate that the IP does not belong to a legitimate mail server.) These entries will be found in a file named /var/log/mail.log and will contain the text “connect from unknown” with the IP address appearing in field #8. To obtain a list of these specific entries, sorted by IP address, use the following command.

$ grep 'connect from unknown' /var/log/mail.log | sort -k 8

# Pidgin instructions  (v0.1\_04/14/2015)

Pidgin is the IM client provided for use on all computers within the range. Pidgin is located in the “Start” menu in the Internet folder.



Pidgin may also be started by entering “pidgin” into terminal.

## Adding contacts

In the Buddy list window click the “Buddies” menu and select:

    New Instant Message  -  to open a chat with that person

    Add Buddy  -   to add someone to your buddy list

*Note:* Both these options will open a window and ask for the name of the contact you wish to add. Add Buddy will also have some additional fields that may be left blank.

## Joining Group/Chatrooms

*Note*: at present you may only join chatrooms on the server your account is registered to.

In the Buddy list window click the “Buddies” menu and select:

    Join A Chat  -  to open a room

    Add Chat  -  to add a room to your buddy list

*Note*: Both these options will open a window and ask for the name of the room you wish to add. At the bottom of the window will be a “Room List” button that can be used to get a list of the chatrooms on a server. Add Chat will also have some additional fields that may be left blank. If you enter the name of a group that does not exist, weather by typo or deliberately, a room with the given name will be created.

# User Management Commands

## w

w displays information about the users currently on the machine, as well as their processes. The header shows, in this order, the current time, how long the system has been running, how many users are currently logged on, and the system load averages for the past 1, 5, and 15 minutes. Note that sudo privileges are not required to run this command.

### Example:

To show information about a particular *username* only, add the username as an argument.

$ w *username*

## who

The who command can be used to list the user’s name, terminal line, login time, elapsed time since activity occurred on the line, and the process ID of the command interpreter for each current system user.

### Examples:

Use who to display information about users currently logged in:

$ who

Show the same information as above, with column headings:

$ who -H

Display all login names and the number of users currently logged on:

$ who –q

## adduser

The addusercommand is used to add users to the system according to command-line options and configuration information contained in the file /etc/adduser.conf.   adduser offers a friendlier front-end than the lower level useradd command.  The command will, by default, choose policy-conformant userid (UID) and groupid (GID) values, create a home directory with skeletal configuration, and run a custom script, among other features.

When running adduser, you will be asked a series of questions. The procedure will be:

* Assign and confirm a password for the new user.
* Enter any additional information about the new user, known as GECOS data, including name, building and office number, telephone, etc. These fields are entirely optional and can each be skipped by pressing *<Enter>* if you do not wish to utilize these fields.
* Finally, you will be asked to confirm that the information you provided was correct. Press *<y>* to continue.

### Examples:

Adding or modifying user accounts requires sudo privileges. You can add a new user by typing:

$ sudo adduser *username*

If the new user should have the ability to execute commands with *root* (administrative) privileges, you will need to give the new user access to sudo. You can do this by adding the new user’s username to a special group called sudo:

$ sudo usermod -a -G sudo username

Users who belong to the group sudo are able to execute any command with root privileges.

## deluser

The deluser command is used to remove a user from the system:

$ sudo deluser *username*

### Examples:

It is often desirable to remove the user’s home directory at the same time that the account is deleted:

$ sudo deluser --remove-home *username*

If, additionally, all files owned by the user should be deleted, including files outside the user’s home directory, use this option instead:

$ sudo deluser --remove-all-files *username*

## addgroup

The addgroupcommand is used to add groups to the system. You can add a new user by typing:

$ sudo addgroup *newgroup*

### Example:

Use addgroupto create the group programmers:

$ sudo addgroup programmers

## delgroup

The delgroup command is used to remove a group from the system:

$ sudo delgroup *groupname*

It should be noted that some limitations exist on which groups can be removed. For example, the primary group for any existing user cannot be removed.

### Example:

It may not be desirable to remove groups that still have members. To prevent this from occurring accidently, add the following option:

$ sudo delgroup --only-if-empty *groupname*

## passwd

The passwdcommand changes passwords for user accounts. A normal user may only change the password for his/her own account, while a user with root privileges may change the password for any account.  passwd also changes account information, such as the full name of the user, the user´s login shell, or his/her password expiry date and interval.

For password changes, the user is first prompted for his/her old password, if one is present.  This password is then encrypted and compared against the stored password. The user has only one chance to enter the correct password.  Users with root privileges are permitted to bypass this step so that forgotten passwords may be changed.

After the password has been entered, password aging information is checked to see if the user is permitted to change the password at this time. If not, passwd refuses to change the password and exits.

The user is then prompted twice for a replacement password. The second entry is compared against the first and both are required to match in order for the password to be changed.

The password is then tested for complexity. As a general guideline, passwords should consist of 6 to 8 characters including one or more characters from each of the following sets:

* lower case alphabetics
* digits 0 thru 9
* punctuation marks

### Examples:

Use passwd to display account status information for all users. The status information consists of seven fields. The first field is the user´s login name. The second field indicates whether the user account is locked (L), has no password(NP), or has a usable password (P). The third field gives the date of the last password change. The next four fields are the minimum age, maximum age, warning period, and inactivity period for the password. These ages are expressed in days.

$ sudo passwd –a –S

Use passwd to lockthe account named *user1*. This option disables an account by changing the password to a value that matches no possible encrypted value, and by setting the account expiry field to 1.

$ sudo passwd –l *user1*

# System Monitoring Commands

## ps

The ps command reports on active processes. (A process is defined as a program in execution.) The syntax of this command is quite simple.

$ ps [options]

By default (when run without options), ps displays information on all processes belonging to the logged-in user. It displays the process ID (pid=PID), the terminal associated with the process (tname=TTY), the cumulated CPU time in [DD-]hh:mm:ss format (time=TIME), and the executable name (ucmd=CMD). Output is unsorted by default.

A more common invocation of the ps command, especially for system administrators, displays information on all processes that are currently running on the system. The -e option specifies that all running processes be displayed, while the -f option specifies a full format listing. As shown in this example, options can be grouped behind a single - character.

$ ps -ef

The ps command provides a snapshot of the system state at a single moment in time. If you want to have repetitive updates in real time, use the top or htop commands, described below.

## top

The top program provides a dynamic real-time view of a running system.  It can display system summary information as well as a list of processes or threads currently being managed by the Linux kernel.  The types of system summary information shown and the types, order and size of information displayed for processes are user configurable.

top is useful for system administrators, as it shows which users and processes are consuming the most system resources at any given time.

Although top can take a number of options to tailor the information displayed, running top without any options provides a wealth of information about a running system.

$ top

### Example:

Pressing n while top is running allows the user to change the number of processes displayed. The default is an unlimited number of processes.

## htop

htop is designed as an alternative to the top program. It shows a frequently updated list of the processes running on a computer, normally ordered by the amount of CPU usage. Unlike top, htop provides a full list of processes running, instead of the top resource-consuming processes. htop uses color and provides visual information about processor, swap and memory status. Unlike top, htop provides more flexible scrolling capabilities.

It is important to use commands such as htop to monitor processes and resource utilization to be able to quickly detect unusual events or system behavior.

### Example:

Running htop with the -u option, followed by a username, shows only the processes of a given user.

$ htop -u bondj

It is best to take advantage of the interactive display control features of htop by using the special keys while the command is active.  Start htop by simply entering the following command.

$ htop

The following keys, among others, are active while using htop:

#### Pressing <F1> displays a help screen.  Press any key to return to htop main screen.

#### Pressing <F2> or <s> displays the setup menu for controlling what is displayed and how.

#### Pressing <Arrows>, <PgUP>, <PgDn>, <Home>, <End> keys provide scrolling capability.

#### Pressing <F5> or <t> displays a tree view in which processes are organized by parenthood

#### Pressing <space> tags/untags a process.

#### Pressing <F9> or <k> sends a kill signal to the currently tagged process(es). Use cautiously!

#### Pressing <F10> or <q> to quit htop.

## netstat

netstat is a useful tool for checking your network configuration and activity. It is very often used to check active connections, as well as network sockets that are that are waiting for a connection (i.e., listening).

### Examples:

To display all active connections and services, including TCP and UDP, use the netstat command with options -atun, for **a**ll, **t**cp, **u**dp, with **n**umeric address.

$ netstat –atun

Active Internet connections (servers and established)

Proto Recv-Q Send-Q    Local Address           Foreign Address     State

tcp        0      0     0.0.0.0:5269          0.0.0.0:\*           LISTEN

tcp        0      0     0.0.0.0:22            0.0.0.0:\*           LISTEN

tcp        0      0     0.0.0.0:25            0.0.0.0:\*           LISTEN

tcp        0      0     0.0.0.0:993           0.0.0.0:\*           LISTEN

tcp        0      0     0.0.0.0:995           0.0.0.0:\*           LISTEN

tcp        0      0     0.0.0.0:5222          0.0.0.0:\*           LISTEN

tcp        0      0     0.0.0.0:110           0.0.0.0:\*           LISTEN

tcp        0      0     0.0.0.0:143           0.0.0.0:\*           LISTEN

tcp        0      0     10.0.1.195:22         10.0.1.54:51643     ESTABLISHED

tcp        0    368     10.0.1.195:22         10.0.1.54:51646     ESTABLISHED

tcp6       0      0     :::5269               :::\*                LISTEN

tcp6       0      0     :::22               :::\*                LISTEN

tcp6       0      0     :::25                 :::\*                LISTEN

tcp6       0      0     :::993                :::\*                LISTEN

tcp6       0      0     :::995                :::\*                LISTEN

tcp6       0      0     :::5222               :::\*                LISTEN

tcp6       0      0     :::110                :::\*                LISTEN

tcp6       0      0     :::143                :::\*                LISTEN

tcp6       0      0     :::80                 :::\*                LISTEN

udp        0      0     10.0.1.195:123        0.0.0.0:\*

udp        0      0     127.0.0.1:123         0.0.0.0:\*

udp        0      0     0.0.0.0:123           0.0.0.0:\*

udp6       0      0     ::1:123              :::\*

udp6       0      0     :::123                :::\*

To display all active TCP connections and services, including the associated processes, use the netstat command with options -atnp, for **a**ll, **t**cp, with **n**umeric address and process information. Note that sudo is required to display the process names.

$ sudo netstat –atnp

Active Internet connections (servers and established)

Proto Recv-Q Send-Q Local Address  Foreign Address    State   PID/Program name

tcp        0      0 0.0.0.0:5269      0.0.0.0:\*       LISTEN  1046/lua5.1

tcp        0      0 0.0.0.0:22        0.0.0.0:\*       LISTEN  878/sshd

tcp        0      0 0.0.0.0:25        0.0.0.0:\*       LISTEN  2051/master

tcp        0      0 0.0.0.0:993       0.0.0.0:\*       LISTEN  2329/dovecot

tcp        0      0 0.0.0.0:995       0.0.0.0:\*       LISTEN  2329/dovecot

tcp        0      0 0.0.0.0:5222      0.0.0.0:\*       LISTEN  1046/lua5.1

tcp        0      0 0.0.0.0:110       0.0.0.0:\*       LISTEN  2329/dovecot

tcp        0      0 0.0.0.0:143       0.0.0.0:\*       LISTEN  2329/dovecot

tcp6       0      0 :::5269           :::\*            LISTEN  1046/lua5.1

tcp6       0      0 :::22             :::\*            LISTEN  878/sshd

tcp6       0      0 :::25             :::\*            LISTEN  2051/master

tcp6       0      0 :::993            :::\*            LISTEN  2329/dovecot

tcp6       0      0 :::995            :::\*            LISTEN  2329/dovecot

tcp6       0      0 :::5222           :::\*            LISTEN  1046/lua5.1

tcp6       0      0 :::110            :::\*            LISTEN  2329/dovecot

tcp6       0      0 :::143            :::\*            LISTEN  2329/dovecot

tcp6       0      0 :::80             :::\*            LISTEN  1710/apache2

## ssh

ssh is a program that enables secure communications from a client to a host computer.  You should get the usual password prompt (or be told you can't log in, if passwords are disabled) when the connection is made.

### Examples:

The general form for logging into a host is shown below.

$ ssh *<username>*@*<hostname* or *IP address>*

Use ssh to connect to the computer named laptop as user joe:

$ ssh joe@laptop

Use ssh to connect to the computer with IP address 192.168.1.1 as user mike:

$ ssh mike@192.168.1.1

# File Permission Management Commands

## chmod

The chmodcommand is used to change the access rights for a file/directory.  More specifically, chmod changes the file mode bits of each given file, which can be either a symbolic representation of changes to make, or an octal number representing the bit pattern for the new mode bits.  Note that everything in Linux is a file. Directories are files, files are files and devices are files. Users with sudo privileges are able to access any file on the system.

The format of a *symbolic mode* is [**ugoa**...][[**+-=**][**perms**...]...], where perms is either zero or more letters from the set **rwxXst**, or a single letter from the set **ugo**.  Multiple symbolic modes can be given, separated by commas.

A combination of the letters **ugoa** controls which users’ access to the file will be changed: the user who owns it (**u**), other users in the file’s group (**g**), other users not in the file’s group (**o**), or all users (**a**). If none of these are given, the effect is as if (**a)** were given, but bits that are set in the umask are not affected.

The operator **+** causes the selected file mode bits to be added to the existing file mode bits of each file; **-** causes them to be removed; and **=** causes them to be added and causes unmentioned bits to be removed except that a directory’s unmentioned set user and group ID bits are not affected.

The letters **rwxXst** select file mode bits for the affected users: read (**r**), write (**w**), execute (or search for directories) (**x**), execute/search only if the file is a directory or already has execute permission for some user (**X**), set user or group ID on execution (**s**), restricted deletion flag or sticky bit (**t**).  Instead of one or more of these letters, you can specify exactly one of the letters ugo:  the permissions granted to the user who owns the file (**u**), the permissions granted to other users who are members of the file’s group (**g**), and the permissions granted to users that are in neither of the two preceding categories (**o**).

A *numeric mode* is from one to four octal digits (0-7), derived by adding up the bits with values 4, 2, and 1. Omitted digits are assumed to be leading zeros. The *first* digit selects the set user ID (**4**) and set group ID (**2**) and restricted deletion or sticky (**1**) attributes. The *second* digit selects permissions for the user who owns the file: read (**4**), write (**2**), and execute (**1**); the *third* selects permissions for other users in the file’s group, with the same values; and the fourth for other users not in the file’s group, with the same values.

### Examples:

Given the following files and permissions:

-rw-r--r--  1 user user 0 Nov 19 20:13 file1

-rw-r--r--  1 user user 0 Nov 19 20:13 file2

-rw-r--r--  1 user user 0 Nov 19 20:13 file3

-rw-r--r--  1 user user 0 Nov 19 20:13 file4

Use chmod to add *owner* execute(x) bit to file1:

$ chmod u+x file1

This is what the file listing for file1 now looks like:

-rwxr--r--  1 user user 0 Nov 19 20:13 file1

Use chmod to add *other* write(w) and execute(x) bits to file2:

$ chmod o+wx file2

This is what the file listing for file2 now looks like:

-rw-r--rwx  1 user user 0 Nov 19 20:13 file2

Use chmod to remove *group* read(r) bit from file3:

$ chmod g-r file3

This is what the file listing for file3 now looks like:

-rw----r--  1 user user 0 Nov 19 20:13 file3

Use chmod, in numeric mode, to add read(4), write(2) and execute(1) to *everyone* for file4:

$ chmod 777 file4

This is what the file listing for file4 now looks like:

-rwxrwxrwx  1 user user 0 Nov 19 20:13 file4

## chown

The chowncommand is used to change the file *owner* and/or *group*.

### Examples:

Usechown **to c**hange the owner of mydirectory to root:

$ chown root mydirectory

You can change the owner of mydirectoryto rootand the group to students:

$ chown root:students mydirectory

You can change the owner of mydirectory, along with its included files, to root(Note: *Great care should be exercised when applying chown recursively*):

$ chown -hR root mydirectory

## sudo

The sudocommand is used to execute a privileged command (one that typically requires being logged on as root).  You will be prompted for your password the first time you use **sudo**.

### Example:

Use sudo to execute the mountcommand, which can only be executed by root:

$ sudo mount /dev/sda3 /media/flashdrive

# Help and Editor Commands

## man

man is used for displaying manual pages (usually called *manpages*). Manpages are self-contained reference documents stored on the host’s hard drive. They are usually short, but some manpages can be quite long. Ubuntu's manpage system has a reference for every command-line program, and in many cases it is the only source of information. It offers a quick reference to the switches and options available in terminal-based commands and programs.  man displays manpages using a paging system, which displays the information one page at a time.

### Example:

Use man to display the manpages for the ps command:

$ man ps

## apropos

The apropos command provides a means of searching manpage names and descriptions.  Each manual page has a short description available within it.   apropos searches the descriptions for instances of *keyword*.  K*eyword* is usually a regular expression, as if (-r) was used, or may contain wildcards (-w), or match the exact keyword (-e).   Using these options, it may be necessary to quote the keyword or escape (\) the special characters to stop the shell from interpreting them.

### Options:

**apropos** [**-e**|**-w**|**-r**] keyword ...

**-r**

Interpret each keyword as a regular expression.   This is the default behavior.  Each keyword will be matched against the manpage names and the descriptions independently.  It can match any part of either.  The match is not limited to word boundaries.

**-w**

Interpret each keyword as a pattern containing shell style wildcards.  Each keyword will be matched against the manpage names and the descriptions independently.  If (-e, --exact) is also used, a match will only be found if an expanded keyword matches an entire description or page name.  Otherwise, the keyword is also allowed to match on word boundaries in the description.

**-e**

Each keyword will be exactly matched against the page names and the descriptions.

### Example:

To find the manpages containing the keyword ps you could enter the following command:

$ apropos –e ps

## emacs

emacs is one of the oldest and most versatile text editors available for Linux and UNIX-based systems. It is well known for its powerful and rich editing features. emacs is also more than just a text editor; it can be customized and extended with different "modes" that support specialized editing features particular to tasks (e.g., writing Java, C or Python programs).  It takes time to learn how best to interact with emacs--be patient.

Commands in emacs are control characters (e.g., hold down the <CTRL> key while typing another character).  Here are a few of the most commonly used commands.

### Help Commands

|  |  |
| --- | --- |
| <CTRL>-h | **help-command**: first character in lots of useful help commands |
| <CTRL>-h t | **help-with-tutorial**: command to run the tutorial |
| <CTRL>-h a | **command-apropos**: prompts for a string and then searches for all emacs commands that contains that string |
| <CTRL>-h ? | **help-for-help**: describes how to use the help facilities |

### File Reading/Writing Commands

|  |  |
| --- | --- |
| <CTRL>-x <CTRL>-f | **find-file**: first prompts for a filename and then loads that file into an editor buffer of the same name |
| <CTRL>-x <CTRL>-s | **save-buffer**: saves the buffer into the associated filename |

### Other Commands

|  |  |
| --- | --- |
| <CTRL>-x <CTRL>-c | **save-buffers-kill-emacs**: when you are finished editing, to save the edited but unsaved buffers and to return you to the UNIX prompt |
| <CTRL>-g | **keyboard-quit**: if while typing a command you make a mistake and want to stop, this aborts a command in progress |
| <CTRL>-x u | **undo**: undoes the last command typed, in case you made a mistake |

### Example:

To start emacs and open a file named myfile:

$ emacs myfile

# File System Commands

## df

The df (disk filesystem) command displays the amount of disk space available on one or more file systems. By default, df reports the space available on all currently mounted file systems. Disk space is shown in 1-kilobyte blocks by default, which can be difficult to read. Using the -h (human-readable) option causes df to print sizes in human readable format (e.g., 1K, 234M, 2G).

### Example:

To display all mounted file systems and their disk usage in human-readable form:

$ df -h

## du

The du (disk usage) command reports the sizes of directory trees, including of all of their contents and the sizes of individual files. This makes it useful for tracking down space hogs, i.e., directories and files that consume large or excessive amounts of space on a hard disk.

du is commonly employed by system administrators as a supplement to automated monitoring and notification programs that help prevent key directories and partitions (logically independent sections of a hard disk) from filling up. Full, or even nearly full, directories and partitions can cause a system to slow down, prevent users from logging in and even result in a system crash. Although visually identifying heavy consumers of disk space can be practical if there are relatively few users on a system, it is clearly not efficient for large systems with hundreds or thousands of users.

The basic syntax is as follows. Of course, like most Linux commands, du accepts a number of options that can change its behavior.

### General format:

$ du <directories and/or files>

### Example:

It’s not a bad idea for a system administrator to keep an eye on the sizes of users’ home directories, looking out for space hogs. The following command line displays the overall size of the /home directory, along with all its first-level subdirectories. That’s the job of the given option. In this example the results are piped through the sort command, which displays the directories in reverse order by size, from largest to the smallest. For more information on the sort command, see View/Search Text Files in the Supporting Documents section.

$ sudo du --max-depth=1 /home/ | sort -rn

# Linux Cheat Sheet



Index

addgroup, 45

adduser, 44

Apache Web server, 19

apropos, 52

cat, 40

chmod, 50

chown, 51

delgroup, 45

deluser, 27, 45

df, 54

Dovecot, 20

du, 54

emacs, 53

firewall rules, 14

grep, 41

head, 40

htop, 47

less, 40

Linux cheat sheet, 55

man, 52

manpages, 52

netstat, 48

network diagram, 7

network traffic monitoring, 9

**OTRS**, 38

packet capture, 15

passwd, 46

pfSense, 12

pfSense firewall, 13

pftop, 15

Pidgin, 43

ping, 7, 35

Postfix, 19

process ID (PID), 23

Prosody, 26

prosodyctl, 27

ps, 47

show interfaces, 11

show ip route, 12

sort, 42

SquirrelMail, 18

ssh, 49

sudo, 51

tail, 40

tcpdump, 10, 15

telnet, 35

top, 47

traceroute, 8, 35

user accounts, 25

VyOS, 11

w, 44

who, 44

XMPP, 26