Linux: Configuring and securing your server howto guide

A technical howto document presented to H3ABioNet

H3ABioNet

Pan African Bioinformatics Network for H3Africa

Created by

The System Administrator Task-force

Prepared for

The greater H3ABioNet and H3Africa Consortium community

Document Control

Date	Author	Authorization By	Version	Description
27 June 2014	Suresh Maslamoney	System Administrator Task-force	1.0	First draft

Contributors

Last Name	First Name	Institution	Country
Alibi	Mohamed	Pasteur Institute of Tunis (IPT)	Tunisia
Brown	David	Rhodes University (RU)	South Africa
Indome	David	Noguchi Memorial Institute for Medical Research (NMIMR)	Ghana
Scheepers	Inus	Centre for High Performance Computing (CHPC)	South Africa
Maslamoney	Suresh	Computational Biology Group – UCT (CBIO)	South Africa
Panji	Sumir	Computational Biology Group – UCT (CBIO)	South Africa
Van Heusden	Peter	South African National Bioinformatics (SANBI)	South Africa
Marcello	Lucio	(CIDRES)	Burkina Faso

Reviewers

Last Name	First Name	Institution	Country

Acronyms and Abbreviations

Acronym and Abbreviations	Description
CLI	The Command Line Interface refers to the actual local terminal on the Linux server used to navigate, configure and manage the system
NIC	A Network Interface Card is a physical network card installed the physical server
OS	A Operating System is a piece of software which is installed on a computer system and manages communication between the physical hardware and user based applications
SL	Scientific Linux Operating System

Table of Contents

Introduction

This document was developed by the H3ABioNet system administrator task force and is focused specifically on H3ABioNet system administrators who do not have Linux specific skills. In the level one documentation we looked at how to configure RAID on your hard drives and how to install the three officially supported Linux distributions. In this document we concentrate on configuring and securing your local server

Support Contact Information

Table 1 below lists all the support contact details for the C6145 server. Both groups of support personnel will provide both hardware and software support to H3ABioNet consortium members. The H3ABioNet helpdesk will however provide additional bioinformatics support.

Table 1

Vendor	Contact Number	Contact Person	Description
H3ABioNet Helpdesk	helpdesk@h3abionet.org	Helpdesk	Log all calls via the H3AbioNet helpdesk and a support specialist will be assigned to your call

Overview

This howto document sets out to provide tips, best practices and step by step instructions for configuring and securing your server. The level 2 series of howto guides starts with this document and branches out to additional howto guides such as installing and using a HPC cluster and Globus Online application for transferring data between nodes.

This document begins with the configuration of your network cards. Once your operating system is installed, your next step is to configure the network cards. This step is important as it makes your server accessible via the network.

1. Network Interface Card (NIC) Configuration

For a computer system to work on a network it requires an IP address. The server can be configured to us a static IP address or one assigned by a DHCP server. The below instruction will provide step by step instruction to configure your network interface cards to use either a statically assigned IP address or one automatically assigned by a DHCP server. Setting up of a DHCP server is beyond the scope of this howto and as such will not be discussed.

NOTE:

1. Network Interface Cards in Linux are labelled as "eth" and assigned a number starting at "0" the number assigned to the eth label would increment by 1 for any additional NIC.

2. When adding a comment line in a configuration file, the first character in the line should be a number sign which is represented by the "#" symbol.

Ubuntu / Debian

In the Ubuntu and Debian distributions, the network interface cards are defined and configured in the /etc/network/interfaces configuration file.

```
:: sudo vim /etc/network/interfaces
```

The format of the configuration file is as follows:

The first entry is the internal loopback network interface card (NIC). This configuration is standard and should not be edited.

```
# The loopback network interface
auto lo
iface lo inet loopback
```

The second entry is usually the first physical NIC and labelled as "eth0". To configure the NIC to get an automatically assigned IP address, do the following:

```
auto eth0 (this example assumes you are configuring the primary NIC to use a dynamically
assigned IP address)
iface eth0 inet dhcp
```

To define a static IP address,

```
auto eth2 (this example assumes you are configuring the second NIC to us a statically
assigned IP address)
Address              <IP address>         # define IP address
network              <IP address>         # define network address
netmask              <Subnet address>     # define subnet mask
broadcast            <IP address>         # define broadcast address
gateway              <IP address>         # define default gateway
dns-nameservers      <IP address>         # define name server
```

Once you have defined the configuration for the network cards, you need to bring up the interface. Use the following command;

```
sudo ifup eth0 (to bring up the NIC) and,
sudo ifdown eth0 (to disable the NIC)
```

Tip:
In the above command, change eth0 to reflect the NIC you want to enable or disable

SL
To view the list of network interfaces available on your SL system, from the cli type;

```
ls /etc/sysconfig/network-scripts/
```

NOTE: In SL, the eth label is prefixed with a "ifcfg" label

To configure the NIC, edit the command as follows to access the configuration file

```
sudo vim /etc/sysconfig/network-scripts/ifcfg-eth0
```

The format for a statically assigned IP is as follows:

```
DEVICE=eth0
HWADDR=00:50:43:00:3B:AE            # This is the MAC address of the NIC
ONBOOT=yes                          # Instructs the NIC to enable on bootup
BOOTPROTO=none                      # Do not use DHCP
IPADDR=10.1.1.100         # this server's IP address
NETMASK=255.255.255.0     # subnet mask
TYPE=Ethernet
GATEWAY=10.1.1.1          # default gateway
DNS1=10.1.1.10            # DNS server's IP address
IPV6INIT=no               # Disable IPv6 on this NIC
USERCTL=no                # Users are not allowed to control this NIC
```

To configure the NIC to use a DHCP assigned IP

```
DEVICE=eth0
HWADDR=00:50:43:00:3B:AE  # This is the MAC address of the NIC
ONBOOT=yes                # Instructs the NIC to enable on bootup
BOOTPROTO=DHCP            # Use DHCP
TYPE=Ethernet
IPV6INIT=no               # Disable IPv6 on this NIC
USERCTL=no                # Users are not allowed to control this NIC
```

As with Ubuntu and Debian, once you have made any changes to the network configuration file. You need to restart the network deamon. The command for this is as follows:

```
sudo /etc/rc.d/init.d/network restart
```

The above command will shut down all the NI's and then bring them up again.

If this is a new installation, it's recommended running the following commands:

```
sudo chkconfig network on     # this command instructs the NIC to start automatically
```

With the network cards configured and the IP addresses assigned, we look at setting up the software repositories for installing applications.

2. Software repositories

Repositories can be best described as a central location which houses applications for a particular Linux distribution and version. In Microsoft Windows, if you need a piece of software or driver, you would navigate to the relevant website or DVD supplied with your system and download the required piece of software. With Linux, you would initially configure your system to point to a repository/s (you can have more than one) and when you wish to install any software, you simple run one of the commands listed in the package manager paragraph above. If the repository has your requested software, the software and its dependencies will be downloaded and installed on your server.

NOTE:
if you are logged into the server as a normal user, the application will be installed in your home directory and accessible only to you. If the application needs to be accessible to the entire system, install the application as root

If you do not have access to the internet in which to access the repositories, you could download the source code, build and install the application separately. However, installing applications from a recognized repository and via the package manager is the preferred method.

The two main methods of installing applications on your Linux server is: (a) downloading and compiling the application from source code and (b) automatically installing the software package via a software repository.

Unlike Windows based systems where you would download an .exe file and install it on your system. On Linux servers, the administrator is able to simply issue an install command via the CLI using the Linux distribution specific package manager to install an application. If the application is contained in the repository, it would run a few checks such as disk space required, software dependencies, etc; once all the conditions are met, the software package with any dependencies is usually then automatically downloaded and installed on the intended server.

In Scientific Linux (SL), the repository configuration file can be found in /etc/yum.repos.d/. Following a bare OS installation, the /etc/yum.repos.d/folder should have the following configuration files:

- Sl6x.repo
- Sl-other.repo, and
- Sl.repo

In Ubuntu and Debian distributions, the configuration file is located in /etc/apt/, the file to edit in these distributions is the “sources.list” file.

To configure your OS to point to your country specific data repository - using your text editor of choice, open the applicable file and add in the repositories specific to your country or institution and OS distribution.

SL
Sudo vim /etc/yum.repos.d/sl.repo
When configuring this file to point to your country specific repository, add the epel repository (see tip 2 below)

Ubuntu / Debian
Sudo vim /etc/apt/soruces.list

Tip: Configure server to point to Epel repository – for SL only

- Download the epel repository- from the cli > wget http://ftp.leg.uct.ac.za/pub/linux/fedora-epel/6/i386/repoview/epel-release.html
- cd into the download folder and install the repository with: rpm -ivh epel-release.html

Tip:
Make a copy of the original repo or source file before editing it.

After editing the repository configurations file, it is recommended to refresh the repository cache to take advantage of the new additions by typing the following command

SL
sudo yum update

Ubuntu / Debian
sudo apt-get update

Once your repositories are configured and updated, you can proceed with installing any applications.

2.1. Package manager

A package manager is an application installed on the server as part of the default OS installation which allows you to interact with the software repositories. This would include, searching, updating an installing applications via the software repository.

The Ubuntu and Debian based Linux distributions use the "APT" package manager while the Red Hat based distributions such as Scientific Linux uses the "YUM" package manager.

The package manager command syntax is as follows:

SL
sudo yum install <application name>

Ubuntu and Debian
sudo apt-get install <application name>

Example of command syntax to install applications using a package manager via the command line:

Ubuntu / Debian
:: sudo apt-get install <application name>

SL
:: sudo YUM –y install <application name>

Command explained

- ***Sudo*** *- is used when the logged in account does not have the necessary rights to install the application*
- ***Apt-get or YUM*** *- is the package manager*
- ***Install*** *- is the instruction*
- ***-y*** *- in the YUM command instructs the YUM package manager to assume the "Yes" answer during the application installation*
- ***<Application name>*** *- is the name of the application you intend installing*

Useful Tools to install

- Lsscsi lists all the scsci devices on your server
  sudo yum -y install lsscsi or sudo apt-get install lsscsi
- Lshw lists all hardware installed on the server
  sudo yum -y install lshw or sudo apt-get install lsscsi

Once your server is able to install software, it is time to create some user accounts for the individuals who will be using the server.

Note: The following section assumes that users will be authenticated by the local server and not via a network authentication tool such as ldap or active directory. It therefore provides step by step instructions for creating user accounts on the local server.

3. User Accounts

Each person who accesses a computer system should have a unique username and a strong password. When a user account is created, all files and applications installed and created buy this user is owned by the specific user. User accounts are also used to differentiate between levels of access and roles. It is also used to manage and monitor access to the server.

It is highly recommended to assign strong passwords to user accounts on your local, more so when these accounts have sudo rights. If you cannot think of a strong password –you could use the Ubuntu password tool to generate a secure password. Pwgen is not installed by default; this tool will need to be installed.

Once installed, at the command prompt, type "pwgen" to generate a list of random passwords. At the CLI, type "man pwgen" to get a list of options to use in this command to get stronger passwords.

A strong password is considered to be at least 7 characters long, is made up of upper and lowercase alphabet, numbers and special characters.

3.1. Creating general user accounts

The below commands will create a user account with limited access. This user will have the necessary rights to affect anything within their own profile or home directory. To do anything that affects something outside of their profile or home directory, they would need to grant the user account "sudo" rights, see 3.2 below for instruction on granting a general user account sudo rights.

Ubuntu / Debian

```
sudo adduser <username>
```

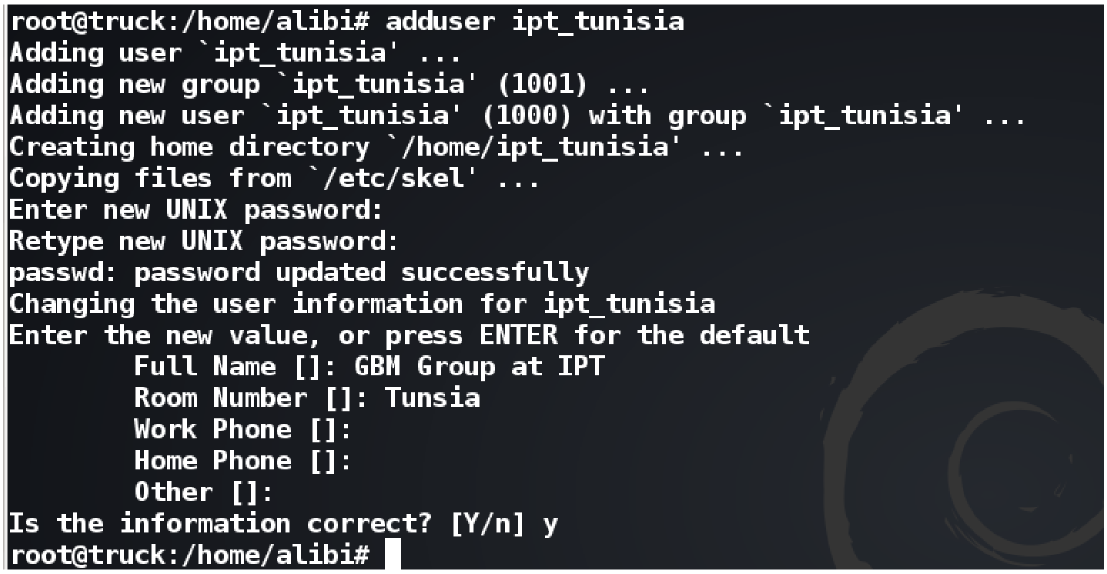
When invoking this command, it would automatically prompt you to give the account a password. Once you enter the password, you are given the opportunity of providing additional user account information.

SL

```
sudo useradd <username>
Passwd <password for account>
```

In SL you would specify the actual username and the password

Example of adding a general user in Ubuntu / Debian



```
root@truck:/home/alibi# adduser ipt_tunisia
Adding user `ipt_tunisia' ...
Adding new group `ipt_tunisia' (1001) ...
Adding new user `ipt_tunisia' (1000) with group `ipt_tunisia' ...
Creating home directory `/home/ipt_tunisia' ...
Copying files from `/etc/skel' ...
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
Changing the user information for ipt_tunisia
Enter the new value, or press ENTER for the default
        Full Name []: GBM Group at IPT
        Room Number []: Tunsia
        Work Phone []:
        Home Phone []:
        Other []:
Is the information correct? [Y/n] y
root@truck:/home/alibi#
```

3.2. Granting user accounts sudo rights

The sudo account allows a normal user to run a command with elevated privileges. The sudo command is equivalent to the "run as" command in windows. It is best practice to log into your server as a normal user and make use of the sudo command to be temporarily grant the logged in user sudo or administrator rights. To use the sudo command, simply type the word "sudo" before your command, after depressing the enter key, you will be prompted to type in your password. If your account has been granted sudo rights, the command will execute with evaluated privileges.

Ubuntu / Debian, to add a user to a group, type the below command. In this example, we will add a user to the sudoers group to grant them administrator rights.

Ubuntu / Debian

Sudo adduser <username> <groupname>

Example:

sudo adduser <username> sudo

SL

In SL we need to edit the "groups" configuration file and add the user into the group

sudo vim /etc/group > scroll down and look for the line starting with "wheel" > add the new user to this line separated by a comma > save and exit the file, the new user should now have sudo rights.

4. Services

Basic service syntax

To view the status or to start or stop a particular daemon, the syntax is:

sudo service <daemon> <option or command>

- Sudo allows the command to execute with administrative rights
- Service is the command
- <daemon> is the actual service or daemon you want the command to act on
- <option> will determine what action is performed on the daemon in question
    - Status = gives the current status of the specified daemon
    - Stop = stops the specified daemon for the current session
    - Start = starts the specified daemon for the current session

To list all the service available on a Linux computer system

```
sudo service --status-all
```

The output however is displayed differently. The output on Ubuntu or Debian will list all the services running preceded with a plus or minus sign. The plus sign indicates that the service is running and the minus sign indicates that the service is in a stopped state.

In SL, the output will print all the services on the screen with the process ID and their status

To see if a particular service is running, on either distribution, run the following command:

```
sudo service <service name> status
```

5. Securing your local server: The basics

There are various methods and levels of local server security, some of which are beyond the scope of this howto document. For the purposes of this document we will focus on the basics of securing your local server.

Local server security best practices

- The first level of security is physical access to your server. It is recommended to keep the server in a lockable room or cabinet and closely monitor access.
- Limit the amount of local accounts created on the server and pay special attention to which accounts are granted sudo rights
- Insist that users use strong passwords. Strong passwords are considered to be a minimum of seven characters, include a mixture of upper and lower case letters and numbers and special characters. Linux has a password generation tool which could help you create strong passwords – pwgen
- Don’t let users share accounts
- School users to only install software from recognized sources
- From a system administrator point of view, it is advised to:
    - Configure a firewall
    - Secure SSH access
    - Implement a level of server hardening if needed
    - Monitor and regularly scan logs
    - Scan open ports on your system
    - Audit your system security

- Implement regular backups
- Depending on how sensitive the data stored on your server is, consider encrypting the data
- Secure any additional applications in particular PHP, MySQL, etc
- Implement a software and OS update maintenance schedule from recognized sources

5.1. Install and configure a Linux firewall

A firewall is a system designed to prevent unauthorized access to or from your server or network. Installing and configuring a firewall would be a good place to start when looking at securing your server from the internet, your network or your clients. Linux comes bundled with a firewall called "iptables". Iptables is the command line interface. There are some front end applications which administrators prefer to use when configuring the firewall. The more common frontend used is the Ufw (Uncomplicated firewall).

5.1.1. To install ufw from the CLI

Ubuntu / Debian

```
Sudo apt-get install ufw
```

SL 6.4

```
Yum –y install ufw
```

To allow SSH and Http services.

```
sudo ufw allow ssh
sudo ufw allow http
```

Enable the firewall.

```
sudo ufw enable
```

Check the status of the firewall.

```
sudo ufw status verbose
```

5.1.2. iptables

iptables are installed as part of the basic server installation across all three operating systems

To see the status of the iptable on your system, from the CLI type

```
Service iptables status
```

To stop temporarily stop the iptable for the current session:

Service iptables stop (disables the firewall until the server is reboot or service started)

To save the changes made to iptables

Service iptables save (saves the iptables)

To permanently switch off iptables for all current and future sessions:

Chkconfig iptables off (permanently disables the firewall even after a reboot)

To list all firewall rules, type *##* at the CLI

/sbin/iptables –L –v –n | less (shows both input and forward chains)

/sbin/iptables –L INPUT –v –n | less (will show only the INPUT chains)

5.2. Secure shell (SSH)

The ability to manage a server remotely is paramount to a system administrator's day to day activities. Many of the older tools used for this purpose is not as secure as it should be thereby potentially leaving your server vulnerable to attack especially when working remotely via the internet.

In Linux, the defacto tool used to securely manage your server remotely is the Secure Shell protocol or SSH for short. SSH encrypts all communication from the source to the remote computer system.

Secure Shell is a cryptographic network protocol for secure data communication, remote command-line login, remote command execution, and other secure network services between two networked computers. (Source: http://en.wikipedia.org/wiki/Secure_Shell)

Tip:
Linux servers are accessible from Microsoft Windows based environments using the PuTTy application. PuTTY application however is beyond the scope of this document and as such, the document will concentrate solely on SSH

SSH basic syntax

The most basic form of the command is:

ssh remote_host

The remote_host in this example is the IP address or domain name that you are trying to connect to.

This command assumes that your username on the remote system is the same as your username on your local system. If your username is different on the remote system, you can specify it by using this syntax:

ssh remote_username@remote_host

Once you have connected to the server, you will probably be asked to verify your identity by providing a password. For added security, you could replace password based login's with ssh key based login. We cover ssh key based logins in 5.2.3 below

To end your remote session, simply type the work "exit" to logout of the remote server and be returned to your local session,

5.2.1. How Does SSH Work?

SSH works by connecting a client program to an ssh server.

In the above command, ssh is the client program. The ssh server is already running on the remote_host that we specified.

5.2.2. How to configure SSH

When you change the configuration of SSH, you are changing the settings of the sshd server. Be careful editing the file */etc/ssh/sshd_config*. If you make a mistake you may disable the SSH service altogether rending the server inaccessible remotely. You would have to then login to the server at the physical console to repair the problem.

In Ubuntu, the main sshd configuration file is located at /etc/ssh/sshd_config.

Back up the current version of this file before editing:

```
sudo cp /etc/ssh/sshd_config{,.bak}
```

Open it with a text editor:

```
sudo nano /etc/ssh/sshd_config
```

You will want to leave most of the options in this file alone. However, there are a few you may want to take a look at:

```
Port 22
```

The port declaration specifies which port the sshd server will listen on for connections. By default, this is 22.

It may be a good idea to change this to a non-standard port to help obscure your server from random port scans. If you do change your port, we will show you how to connect to the new port later on.

```
HostKey /etc/ssh/ssh_host_rsa_key

HostKey /etc/ssh/ssh_host_dsa_key

HostKey /etc/ssh/ssh_host_ecdsa_key
```

The host keys declarations specify where to look for global host keys. We will discuss what a host key is later. The below two items indicate the level of logging that should occur.

```
SyslogFacility AUTH

LogLevel INFO
```

If you are having difficulties with SSH, increasing the amount of logging may be a good way to discover what the issue is. The below parameters specify some of the login information you need to edit to increase the ssh logging.

```
LoginGraceTime 120

PermitRootLogin yes

StrictModes yes
```

LoginGraceTime specifies how many seconds to keep the connection alive without successfully logging in. It may be a good idea to set this time just a little bit higher than the amount of time it takes you to log in normally.

PermitRootLogin selects whether root is allowed to log in. In most cases, this should be changed to "**no**" when you have created user account that has access to elevated privileges (through su or sudo) and can log in through ssh. To deny ssh access to a specific user, add the below line to the config file. This line should include all the users who are DTO

```
DenyUsers <<first user name>>, <<second user name>>…
```

strictModes is a safety guard that will refuse a login attempt if the authentication files are readable by everyone. This prevents login attempts when the configuration files are not secure.

The below parameters configure an ability called X11 Forwarding. This allows you to view a remote system's graphical user interface (GUI) on the local system. This option must be enabled on the server and given with the client during connection with the "-X" option.

```
X11Forwarding yes

X11DisplayOffset 10
```

If you changed any settings in this file, make sure you restart your sshd server to implement your modifications:

```
sudo service sshd restart
```

You should thoroughly test your changes to ensure that they operate in the way you expect. It may be a good idea to have a few sessions active when you are making changes. This will allow you to revert the configuration if necessary. If you run into problems, or lose remote access, you will have to login at the physical server.

5.2.3. How to log into SSH with keys

While it is helpful to be able to log in to a remote system using passwords, it's often a better idea to set up key-based authentication. Key-based authentication works by creating a pair of keys: a private key and a public key.

The private key is located on the client machine and is secured and kept secret. The public key can be given to anyone or placed on any server you wish to access. When you attempt to connect using a key-pair, the server will use the public key to create a message for the client computer that can only be read with the private key. The client computer then sends the appropriate response back to the server and the server will know that the client is legitimate. This entire process is done in the background automatically after you set up keys.

5.2.4. How to create SSH keys

SSH keys should be generated on the computer you wish to log in from. This is usually your local computer.

Note:
This will only give you passwordless login from the machine you are currently on. If you ssh into the remote server from a different machine, you will have to provide your password

The keys are generated on the computer system you want to login from. On the source computer system, type the following command to generate the key pairs:

```
ssh-keygen -t rsa
```

Press enter to accept the defaults. Your keys will be created at ~/.ssh/id_rsa.pub and ~/.ssh/id_rsa.

Change into the .ssh directory by typing:

```
cd ~/.ssh
```

Look at the permissions of the files:

```
ls -l

-rw-r--r-- 1 demo demo  807 Sep  9 22:15 authorized_keys

-rw------- 1 demo demo 1679 Sep  9 23:13 id_rsa

-rw-r--r-- 1 demo demo  396 Sep  9 23:13 id_rsa.pub
```

As you can see, the id_rsa file is readable and writable only to the owner. This is how it should be to keep it secret. The id_rsa.pub file, however, can be shared and has permissions appropriate for this activity.

5.2.5. How To Transfer Your Public Key to the Server

You can copy the public key to the remote server by issuing this command:

```
ssh-copy-id remote_host
or
ssh-copy-id username@remote_host
```

This will start an SSH session, which you will need to authenticate with your password.

After you enter your password, it will copy your public key to the server's authorized keys file, which will allow you to log in without the password next time.

Client-Side Options

There are a number of optional flags that you can select when connecting through SSH. Some of these may be necessary to match the settings in the remote host's sshd file. For instance, if you changed the port number in your sshd configuration, you will need to match that port on the client-side by typing:

```
ssh -p port_number remote_host
```

If you only wish to execute a single command on a remote system, you can specify it after the host like so:

```
ssh remote_host command_to_run
```

You will connect to the remote machine, authenticate, and the command will be executed.

As we said before, if X11 forwarding is enabled on both computers, you can access that functionality by typing:

```
ssh -X remote_host
```

Providing you have the appropriate tools on your computer, GUI programs that you use on the remote system will now open their window on your local system.   (Source:

https://www.digitalocean.com/community/tutorials/how-to-use-ssh-to-connect-to-a-remote-server-in-ubuntu)

5.3. Server hardening (SELinux and AppArmor)

SELinux (Security-enhanced Linux) was developed by the National Security Agency (NSA) and is a security application which secures / hardens the Linux OS. SELinux is part of the basic server installation on SL 6.4. Ubuntu and Debian however come bundled with SELinux's competitor named "AppArmor". SELinux is a command line tool while AppArmor is a GUI and is often considered to be easier to implement and manage compared to SELinux.

Tip:
This hardening of the server could cause all sorts of knock on effects. If your server is not exposed to the internet or a volatile network, it would be easier to simply switch off SELinux as a default.

AppArmor

AppArmor is part of the basic Ubuntu and Debian server installation from version 7. Should you want to install this application on earlier versions:

sudo apt-get install apparmor apparmor-profiles

Check to see if the application is running:

sudo apparmor_status

Read more: http://en.wikipedia.org/wiki/AppArmor

SELinux

To see the status of SELinux, run the following command:

Getenforce or sestatus

To temporarily disable or reduce the SELinux restrictions. From the command line, type the following (this will disable or enable SELinux for the duration of the session. The status, state or mode goes back to the state SELinux was in before running this command.

sudo setenforce 0 or *sudo setenforce disabled* "to temporarily disable SELinux"

sudo setenforce 1 or *sudo setenforce enable* "to temporarily enable SELinux"

setenforce Permissive (will reduce security and prompt first before blocking. This will remain intact until the server is rebooted or the service restarted)

setenforce Disabled (will disable SELinux until the server is rebooted or the service is restarted)

To permanently (even after a reboot) change the behaviour of SELinux, edit the /etc/sysconfig/selinux file. The line "SELinux=" defines the behaviour of SELinux

Enforce -enforces SELinux to run permanently

Permissive –this mode will prompt first, and

Disabled - disables SELinux permanently

Read more: http://en.wikipedia.org/wiki/Security-Enhanced_Linux

5.4. Nmap - Scanning the local machine or network

Nmap (Network Mapper) is a free and open source utility for network discovery and security auditing. It was originally written by Gordon Lyon and is primarily used to discover hosts and services on a computer network, thus creating a "map" of the network. To accomplish its goal, Nmap sends specially crafted packets to the target host and then analyses the responses.

The software provides a number of features for probing computer networks, including host discovery and service and operating system detection. These features are extensible by scripts that provide more advanced service detection,[2] vulnerability detection,[2] and other features. Nmap is also capable of adapting to network conditions including latency and congestion during a scan. Nmap is under development and refinement by its user community. Nmap was originally a Linux-only utility, but it was ported to Microsoft Windows, Solaris, HP-UX, BSD variants (including Mac OS X), AmigaOS, and SGI IRIX. (Source: http://en.wikipedia.org/wiki/Nmap and http://nmap.org/book/man.html)

Some of the key features of nmap are:

- Host discovery – Identifying hosts on a network. For example, listing the hosts that respond to TCP and/or ICMP requests or have a particular port open.
- Identifying open ports on a target host in preparation for auditing.
- Version detection – Interrogating network services on remote devices to determine application name and version number.
- OS detection – Determining the operating system and hardware characteristics of network devices.
- Scriptable interaction with the target – using Nmap Scripting Engine (NSE) and Lua programming language.
- Nmap can provide further information on targets, including reverse DNS names, device types, and MAC addresses.
- Auditing the security of a device or firewall by identifying the network connections which can be made to, or through it.
- Network inventory, network mapping, maintenance and asset management.
- Auditing the security of a network by identifying new servers. (Source: http://en.wikipedia.org/wiki/Nmap)

5.4.1. Installing nmap

Ubuntu / Debian

sudo apt-get install nmap

SL 6.4

yum –y install nmap

5.4.2. Basic nmap usage commands

Scan your local system for open ports with

```
nmap -v -sT localhost
```

SYN scanning with the following

```
sudo nmap -v -sS localhost
```

For target specifications

```
nmap <targets' URL's or IP's with spaces between them (can also use CIDR notation)
```

Example

```
Nmap scanme.nmap.org, gnu.org/24, 192.168.0.1; 10.0.0-255.1-254
```

For OS detection

```
nmap -O <target domain or IP address>
```

For version detection

```
nmap -sV <target domain or IP address>
```

For configuring response timings (-T0 to -T5 :increasing in aggressiveness)

```
nmap -T0 -sV -O <target domain or IP address>
```

For SYN-stealth scanning by sending TCP packets with the SYN flag set

```
nmap -sS -p <port of target> <IP address of target>
```

(Source: http://en.wikipedia.org/wiki/Nmap)

5.5. Log Files

System log file analysis is one of the most important tasks when analysing your system. In fact, looking at the system log files should be the first thing to do when maintaining or troubleshooting a system. Linux based systems automatically logs almost everything that happens on the system in detail. Normally, system log files are written in plain text and therefore, can be easily read using an editor or pager. They are also parsable by scripts, allowing you to easily filter their content.

For viewing log files in a text console, use the commands less or more. Use head and tail to view the beginning or end of a log file. To view entries appended to a log file in real-time use tail -f. For information about how to use these tools, see their man pages

- Dmesg
- Boot.log

- Syslog
- Auth.log
- Dpkg.log

Linux does have some additional applications which simplify and automate log file management. These applications include logwatch, fail2ban and denyhosts to name but just a few options

5.5.1. LogWatch

Logwatch is a customizable, pluggable log-monitoring script. It parses system logs, extracts the important information and presents them in a human readable manner. To use logwatch, install the logwatch package. logwatch can either be used at the command-line to generate on-the-fly reports, or via cron to regularly create custom reports. Reports can either be printed on the screen, saved to a file, or be mailed to a specified address. The latter is especially useful when automatically generating reports via cron.

logwatch can be customized to great detail. The default configuration should be sufficient in most cases. The default configuration files are located under /usr/share/logwatch/default.conf/. Never change them because they would get overwritten again with the next update. Rather place custom configuration in /etc/logwatch/conf/ (you may use the default configuration file as a template, though).

A detailed HOWTO on customizing logwatch is available at /usr/share/doc/packages/logwatch/HOWTO-Customize-LogWatch. The following config files exist:

- logwatch.conf
- ignore.conf

Filter for all lines that should globally be ignored by logwatch.

- services/*.conf

The service directory holds configuration files for each service you can generate a report for.

- logfiles/*.conf

To install LogWatch, from the CLI

Ubuntu / Debian

```
sudo apt-get install logwatch libdate-manip-perl
```

SL 6.4

```
yudo yum –y install
```

The command-line syntax is easy. You basically tell logwatch for which service, time span and to which detail level to generate a report:

To view logwatch's output use the "less" and "more" options. Using "tail" and "head" will give you the top few lines of the log or the last couple of lines of the log respectively

```
sudo logwatch | less
```

To email a logwatch report for the past 7 days to an email address, enter the following and replace mail@domain.com with the required email. :

```
sudo logwatch --mailto mail@domain.com --output mail --format html --range 'between -7 days and today'
```

Detailed report on all kernel messages from yesterday

```
logwatch --service kernel --detail High --range Yesterday --print
```

Low detail report on all sshd events recorded (incl. archived logs)

```
logwatch --service sshd --detail Low --range All --archives --print
```

Mail a report on all smartd messages from May 5th to May 7th to root@localhost

```
logwatch --service smartd --range 'between 5/5/2005 and 5/7/2005' \

--mailto root@localhost --print
```

5.5.2. Deny Hosts

Deny Hosts is another useful application which is a log-based intrusion prevention security tool specifically for SSH servers written in Python. It is intended to prevent brute-force attacks on SSH servers by monitoring invalid login attempts in the authentication log and blocking the originating IP addresses.

Deny Hosts checks the end of the authentication log for recent failed login attempts. It records information about their originating IP addresses and compares the number of invalid attempts to a user-specified threshold. If there have been too many invalid attempts it assumes a dictionary attack is occurring and prevents the IP address from making any further attempts by adding it to /etc/hosts.deny on the server. Deny Hosts 2.0 and above support centralized synchronization, so that repeat offenders are blocked from many computers. The site denyhosts.net gathers statistics from computers running the software. DenyHosts is restricted to connections using IPv4. It does not work with IPv6.

More reading: http://denyhosts.sourceforge.net/

Installing Deny Hosts

Ubuntu / Debian

```
sudo apt-get install denyhosts
```

SL 6.4

```
sudo yum –y install denyhosts
```

Once the program has finished downloading, denyhosts is installed and configured on your virtual private server. The next step is to whitelist some IP addresses

Note:
After you install DenyHosts, make sure to whitelist your own IP address. Skipping this step will put you at risk of locking yourself out of your own machine.

Open up the list of allowed hosts allowed on your local system

```
sudo nano /etc/hosts.allow
```

Under the description, add in any IP addresses that cannot afford to be banned from the server; you can write each one on a separate line, using this format:

```
sshd: 12.34.45.678
```

After making any changes, be sure to restart DenyHosts so that the new settings take effect on your server:

```
sudo /etc/init.d/denyhosts restart
```

Usually this is all that is needed, however, if you would like to add in some IP's to deny or to if you would like to implement some customized ban time and alerts, configure the Deny Hosts configuration file.

```
sudo nano /etc/denyhosts.conf
```

(**Source:** https://www.digitalocean.com/community/tutorials/how-to-install-denyhosts-on-ubuntu-12-04 and http://en.wikipedia.org/wiki/DenyHosts)

5.5.3. Fail2ban

Fail2ban is more advanced than Deny Hosts as it extends the log monitoring to other services including SSH, Apache, Courier, FTP, and more. Fail2ban scans log files and bans IPs that show the malicious signs -- too many password failures, seeking for exploits, etc.

Out of the box Fail2Ban comes with filters for various services (apache, courier, ftp, ssh, etc).

To install fail2ban

Ubuntu / Debian

```
sudo apt-get install fail2ban
```

SL 6.4

```
sudo yum –y install fail2ban
```

After installation edit the configuration file /etc/fail2ban/jail.local and create the filter rules as required.

```
sudo vi /etc/fail2ban/jail.conf
```

Activate all the services you would like fail2ban to monitor by changing "**enabled = false**" to "**enabled = true**"

For example if you would like to enable the SSH monitoring and banning jail, find the line below and change enabled from false to true.

[ssh]

```
enabled  = true
port     = ssh
filter   = sshd
logpath  = /var/log/auth.log
maxretry = 3
```

If you have selected a non-standard SSH port in our ssh configration then you need to change the port setting in fail2ban for ssh which by default is port 22, to your new port number, for example if you have chosen 1234 then port = 1234

[ssh]

```
enabled  = true
port     = <ENTER YOUR SSH PORT NUMBER HERE>
filter   = sshd
logpath  = /var/log/auth.log
maxretry = 3
```

If you would like to receive emails from Fail2Ban if hosts are banned change the following line to your email address.

```
destemail = root@localhost
```

and change the following line from

```
action = %(action_)s
```

to

```
action = %(action_mwl)s
```

You can also create rule filters for the various services that you would like fail2ban to monitor that is not supplied by default.

```
sudo vi /etc/fail2ban/jail.local
```

When done with the configuration of Fail2Ban restart the service with

```
sudo /etc/init.d/fail2ban restart
```

You can also check the status with.

```
sudo fail2ban-client status
```

5.6. Audit your system security

Tiger is a security tool that can be used both as a security auditing and intrusion detection tool.

To install Tiger

Ubuntu / Debian

```
sudo apt-get install tiger
```

SL 6.4

The latest copy of tiger can be downloaded from http://www-arc.com/tara/ using the “wget” command or from http://www.nongnu.org/tiger/index.html#download

To install, use the following command

```
rpm –ivh “download_file_name.rpm”
```

Once the application has been installed, run tiger by typing the below command

```
sudo tiger -H
```

All Tiger output can be found in the /var/log/tiger folder

It would take some time to complete. Once Tiger has analysed your system, you will be able to view the full report at the following location

```
sudo less /var/log/tiger/security.report.*
```

6. Data Encryption

When working with genomic data, it is at times a requirement to apply additional security measures to secure your research data, particularly when transporting the data. An example of this would be when you transport the data to the H3ABioNet archive. The data would need to be in an encrypted format.

Definition:
Encryption is the coding or scrambling of information so that it can only be decoded and read by someone who has the correct decoding key. Encryption is used in secure Web sites as well as other mediums of data transfer. If a third party were to intercept the information you sent via an encrypted connection, they would not be able to read it.

Note:
Disk encryption does not protect your data from all threats. You will still be vulnerable to attackers who can break into your system (e.g. over the Internet) while it is running and after you've already unlocked and mounted the encrypted parts of the disk.

Disk encryption also won't protect you against someone simply wiping your disk. Regular backups are recommended to keep your data safe. (**Source:** https://wiki.archlinux.org/index.php/disk_encryption)

There are many different types of data encryption, but not all are reliable. In the beginning, 64-bit encryption was thought to be strong, but was proven wrong with the introduction of 128-bit solutions. AES (Advanced Encryption Standard) is the new standard and permits a maximum of 256-bits. Data encryption schemes generally fall into two categories: symmetric and asymmetric. AES, DES and Blowfish use symmetric key algorithms. Each system uses a key which is shared among the sender and the recipient. This key has the ability to encrypt and decrypt the data. With asymmetric encryption such as Diffie-Hellman and RSA, a pair of keys is created and assigned: a private key and a public key. The public key can be known by anyone and used to encrypt data that will be sent to the owner. Once the message is encrypted, it can only be decrypted by the owner of the private key. Asymmetric encryption is said to be somewhat more secure than symmetric encryption as the private key is not to be shared. (**Source:** http://www.spamlaws.com/data-encryption.html)

Linux based systems also have access to several different command line tools that can encrypt and decrypt files using a password supplied by the user. These CLI based tools allow the user the ability to easily encrypt files on the fly to send over the Internet or unsecured channels without the worry of third parties accessing the files if somehow the transmission is intercepted.

There are a myriad of encryption tools both free and commercial. In this document we will cover the following applications:

- GnuPG
- crypt
- 7-Zip

Note:
Some counties have laws around data encryption. When encrypting your data, please ensure it is in accordance with the local law.

6.1. GnuPG

GPG or GnuPG stands for GNU Privacy Guard and is GNU's tool for secure communication and data storage on Linux type systems. It can be used to encrypt data and to create digital signatures. It also includes an advanced key management facility.

Some Linux distributions have GPG installed by default. To confirm if GnuPG is already installed on your server, run the following command:

```
which gpg
```

If the application is not installed, follow the below instruction to install the application on your preferred OS:

Ubuntu / Debian

```
sudo apt-get install gnupg pbuilder ubuntu-dev-tools bzr-builddeb apt-file
```

Description of applications installed:

gnupg – GNU Privacy Guard contains tools you will need to create a cryptographic key with which you will sign files you want to upload to Launchpad.
pbuilder – a tool to do reproducible builds of a package in a clean and isolated environment.
Ubuntu-dev-tools (and devscripts, a direct dependency) – a collection of tools that make many packaging tasks easier.
bzr-builddeb (and bzr, a dependency) – distributed version control with Bazaar, a new way of

working with packages for Ubuntu that will make it easy for many developers to collaborate and work on the same code while keeping it trivial to merge each other's work.
apt-file - provides an easy way to find the binary package that contains a given file.

SL

```
sudo yum –y install gnupg2
```

If the gpg application is not part of your repository, the source code can be downloaded and install following the below instruction:

Download from >>
https://www.gnupg.org/download/
Once downloaded >>

```
tar xfvz ./gnupg-1.4.11.tar.gz
cd ./gnupg-1.4.11
./configure
make
sudo make install
Add /GnuPG to PATH if it does not automatically appear
```

If you wish to generate a gpg key. Run the below command at the CLI and follow the prompts choosing from the various options depending on how you want the pgp application to encrypt the file/s.

```
sudo gpg --gen-key
```

To encrypt a single file using GPG, use the following command:

```
sudo gpg -c filename
```

Example, to encrypt myfinancial.info.txt file, type the command:

```
sudo gpg -c myfinancial.info.txt
```

You will be prompted to enter a password (twice). A new file is created during the encryption process. The original file will also remain, so you will need to delete it if you only intend to keep an encrypted copy. If you compare the file sizes of the original file and the encrypted file, you will see that the encrypted file is smaller. This is because gpg compresses the file during encryption. If the file is already compressed (e.g. a .zip file or a .tgz file) then the encrypted file might actually end up being slightly larger.

Above command explained:
The -c instructs gpg to encrypt with symmetric cipher using a passphrase. The default symmetric cipher used is CAST5, but may be chosen with the --cipher-algo option. This option may be combined with --sign (for a signed and symmetrically encrypted message), --encrypt (for a message that may be decrypted via a secret key or a passphrase), or --sign and --encrypt together (for a signed message that may be decrypted via a secret key or a passphrase).

Warning:
Please note that if you ever forgot your password (passphrase), you cannot recover the data as it uses very strong encryption.

To decrypt file use the gpg command as follow:

```
sudo gpg myfinancial.info.txt.gpg
```

Sample outputs:

```
gpg myfinancial.info.txt.gpg
gpg: CAST5 encrypted data
Enter passphrase:<YOUR-PASSWORD>
```

To decrypt file and write output to file of your choosing, you can run the below command:

```
sudo gpg myfinancial.info.gpg –o “foldername”
```

Also note that if the file extension is .asc, it is a ASCII encrypted file and if file extension is .gpg, it is a binary encrypted file. (**Source:** http://www.cyberciti.biz/tips/linux-how-to-encrypt-and-decrypt-files-with-a-password.html)

To see a list of the algorithms available type:

```
gpg --version
```

The list of available algorithms is shown in the “Supported algorithms” section of the output under the “Cipher” tag. To use a different algorithm add the “-crypto-algo” parameter followed by the algorithm you want to use, e.g. “-crypto-algo=3DES”

The full command then becomes:

```
gpg -c -crypto-algo=3DES big.txt
```

6.2. Crypt

The original Unix systems included a command called “crypt“, however the level of security it provided was very low. In its honour, there are some other commands which can replace it including “bcrypt” and “ccryrpt“.

bcrypt uses the blowfish algorithm while ccrypt is based on the Rijndael cipher, which is the algorithm used for AES. Many cryptoanalysts no longer recommend the use of the blowfish algorithm as there are some theoretical attacks published which weaken it, however for casual encryption, which won’t be subject to state-level (NSA, MI5, FSA) snooping, it is still useful.

Ubuntu / Debian

```
sudo apt-get install bcrypt ccrypt
```

SL

```
Sudo yum –y install bcrypt ccrypt
```

To encrypt a file with bcrypt use:

```
bcrypt filename.txt
```

Similarly, to encrypt a file with ccrypt use the below command. There are three possible ways to call the ccrypt command. By using ccrypt directly with either the -e or -d options to encrypt or decrypt respectively or by using the ccencrypt ccdecrypt commands.

```
ccrypt filename.txt
```

Unlike gpg, the bcrypt command will replace the original file with the encrypted file and add .bfe to the end of the file name. Like gpg, the resulting file is also compressed and so the file size should be significantly smaller for uncompressed files. Compression can be disabled by using the "-c" parameter. When encrypting with ccrypt, the original file will be replaced by big.txt.cpt. Unlike gpg and bcrypt, the output isn't compressed. If compression is needed then tools like gzip can be used. Suggested file extensions for compressed and encrypted files are .gz.cpt or .gzc.

To decrypt a file using bcrypt:

```
bcrypt filename.txt.bfe
```

The .bfe file will be replaced by the original unencrypted file.

Similarly, to decrypt a file using using ccrypt:

```
ccdecrypt filename.txt.cpt
```

6.3. 7-Zip

The 7-Zip compression tool also incorporates AES encryption. To create an encrypted archive use the "-p" parameter with the 7z command:

```
7z a -p filename.txt.7z filename.txt
```

You will be prompted to enter a password (twice). The file will then be compressed and encrypted. The original file will remain so as with the gpg command, you will need to delete it if you only want to keep an encrypted copy. The advantage of using 7-Zip is that multiple files and folders can be archived and encrypted at once. (**Source:** http://www.maketecheasier.com/encrypt-files-on-linux/)

Note:
Using longer more complex phrases provides better security than shorter ones. Linux based systems offer the "pwgen" application which allows you to generate strong passwords.

7. Software application and OS updates

Most security incidents on servers and in networks are cantered around flaws in the OS. As these flaws are discovered, vendors release patches to cover these security holes – by updating your operating system you ensure it has all the latest patches. Linux systems have a command-line software update tool (for example, up2date for Scientific Linux and apt-get for Ubuntu and Debian distributions. If nothing else, the makers of your distribution will keep a mailing list for notifying users of updates to the distribution. Check at your distribution's website for more specifics.

If possible, configure your software to automatically install security patches. Security patches are often released in response to attacks that are actively being used throughout the Internet. It is debatable, but some expert's advice on installing security patches the day that they are announced.

Attackers know that they have a small window of time before machines are patched, so they try to attack as many machines as possible as quickly as possible. Often these attacks are made on a holiday when the attackers hope that most sysadmins will not be able to patch their machines. These attacks are automated, with each newly compromised machine joining in the attacking force.

Any application can have a security hole. For example, Oracle, Adobe Reader, Adobe Flash, and Java have had serious security holes that needed patching. It is highly recommended that you implement a software application and OS update schedule to keep your server patched. (**Source:** https://itservices.uchicago.edu/page/updating-and-patching-unixlinux-oses)

:: Instruction Complete ::

8. References

8.1. **SSH**

- 8.1.1. https://www.digitalocean.com/community/tutorials/how-to-use-ssh-to-connect-to-a-remote-server-in-ubuntu
- 8.1.2. http://en.wikipedia.org/wiki/Secure_Shell

8.2. Data Encryption

8.2.1. GnuPG

https://help.ubuntu.com/community/GnuPrivacyGuardHowto
http://packaging.ubuntu.com/html/getting-set-up.html
http://www.laurencegellert.com/2011/04/gnupg-howto-tutorial-notes/

Please forward any queries, comments or complaints you may have about howto to the H3ABioNet System Administrator Task-force: sys_admin_tf@lists.h3abionet.org