5.2 INTRODUCTION

Most of the computers today are networked in some way to each other either on the Internet or privately in universities, offices, campus wide, at home, etc. Therefore, though any standalone Linux System can act as a fully functional server. Its true power and usefulness can only be realized if the Linux System is networked and interconnected with other systems.

In this unit, we will be specifically looking at networking using the Linux Operating System. This unit assumes that you have already installed a Linux distribution on a computer or computers and ready to try out all the examples stated in the unit as we progress. This unit also assumes that you are aware of the basics of TCP/IP networking. If you need to brush up on that topic, you are highly encouraged to refer to the numerous ebooks and tutorials available online on the Internet.

5.3 INSTALLATION AND CONFIGURATION OF A LAN

A Local Area Network or LAN is generally a small setup of interconnected computers connected via Ethernet Switches. A LAN would consist of both passive and active components. The passive components would include CAT6 cables, Patch Cords, RJ45 ports, Patch Panels, etc. The active components would include the Ethernet Switches, etc.

5.3.1 INSTALLATION

In order to setup a simple LAN of two computers and walk through our examples in this unit, we require the following items.

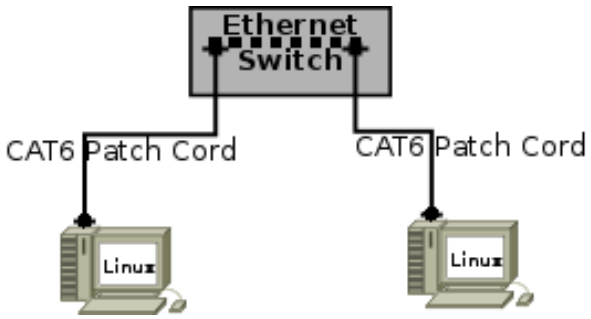
Passive Components : 2 nos. of CAT 6 patch cords.

Active Components : 1 nos. Ethernet Switch.

Computers : 2 nos. of computers running Linux. Both the computers should have at least one Network Interface each on them.

Operating System : Fedora 18 (64 bit) and Ubuntu 12.04.2 LTS (64 bit).

Once we have all the components we connect them as shown in the figure below.



We will need at least two IP Addresses to configure the network interfaces on these two computers. Let us suppose that we will use the following configurations.

For Computer A (Fedora 18):

IP Address: 192.168.1.1 Subnet Mask: 255.255.255.0 Gateway: 192.168.1.1

For Computer B (Ubuntu 12.04.2 LTS):

IP Address: 192.168.1.2 Subnet Mask: 255.255.255.0 Gateway: 192.168.1.2

5.3.2 CONFIGURATION

Configuring the IP Address:

In the examples below the **bold-face** font indicates the commands or text that needs to be typed in.

On Computer A (Fedora 18)

```
[root@computer-a ~]# vim /etc/sysconfig/network-scripts/ifcfg-p5p1
UUID="48eed959-2611-46c0-b0b4-b90d38c05bd2"
NM_CONTROLLED="yes"
BOOTPROTO=none
```

```
DEVICE="p5p1"
ONBOOT="yes"
TYPE=Ethernet
IPV4_FAILURE_FATAL=yes
IPV6INIT=no
IPADDR0=192.168.1.1
PREFIX0=24
HWADDR=AA:BB:CC:DD:EE:FF
```

After making these changes to the file we need to type "**:wq**" and press the enter key on the keyboard to write the changes and quit. Now, in order to apply these changes we need to restart the network service.

```
[root@computer-a ~]# systemctl restart network.service
```

On Computer B (Ubuntu 12.04.2 LTS)

```
root@computer-b:~# vim /etc/network/interfaces
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).
# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
auto eth0
iface eth1 inet static
    address 192.168.1.2
    netmask 255.255.255.0
```

After making these changes to the file we need to type "**:wq**" and press the enter key on the keyboard to write the changes and quit. In order to apply these changes we need to stop and start networking.

```
root@computer-b:~# /etc/init.d/networking stop
root@computer-b:~# /etc/init.d/networking start
```

Once both the computers are setup with the IP Addresses successfully, we test the connectivity with the **ping** utility.

On Computer A (Fedora 18)

```
[root@computer-a ~]# ping 192.168.1.2
PING 192.168.1.2 (192.168.1.2) 56(84) bytes of data.
64 bytes from 192.168.1.2: icmp_seq=1 ttl=64 time=0.142 ms
64 bytes from 192.168.1.2: icmp_seq=2 ttl=64 time=0.141 ms
64 bytes from 192.168.1.2: icmp_seq=3 ttl=64 time=0.129 ms
64 bytes from 192.168.1.2: icmp_seq=4 ttl=64 time=0.139 ms
64 bytes from 192.168.1.2: icmp_seq=5 ttl=64 time=0.143 ms
^C
--- 192.168.1.2 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 3999ms
rtt min/avg/max/mdev = 0.129/0.138/0.143/0.015 ms
```

On Computer B (Ubuntu 12.04.2 LTS)

```
root@computer-b:~# ping 192.168.1.1
PING 192.168.1.1 (192.168.1.1) 56(84) bytes of data.
64 bytes from 192.168.1.1: icmp_req=1 ttl=64 time=0.447 ms
64 bytes from 192.168.1.1: icmp_req=2 ttl=64 time=0.257 ms
64 bytes from 192.168.1.1: icmp_req=3 ttl=64 time=0.182 ms
64 bytes from 192.168.1.1: icmp_req=4 ttl=64 time=0.238 ms
64 bytes from 192.168.1.1: icmp_req=5 ttl=64 time=0.416 ms
^C
--- 192.168.1.1 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4000ms
rtt min/avg/max/mdev = 0.182/0.308/0.447/0.104 ms
```

You can observe from the ping replies that we now have a basic working TCP/IP network.

5.4 INSTALLATION AND CONFIGURATION OF A PROXY SERVER – SQUID

The Squid proxy server is a cache based proxy. It will fetch all web requests to populate its cache and then allow access to its cache based on its configuration. Squid is a widely used proxy server as it permits you to save Internet Bandwidth and is feature rich. It runs on most available operating systems, including Windows and is licensed under the GNU GPL.

In this section, we will install and configure a basic squid proxy server on both the Fedora 18 and Ubuntu 12.04.2 operating systems. You need to ensure that an active Internet connection is available on the system you are performing the installation steps mentioned below, as the software repositories being used are located on the Internet.

5.4.1 INSTALLATION

On Computer A (Fedora 18)

```
[root@computer-a ~]# yum install squid
Loaded plugins: langpacks, presto, refresh-packagekit
Resolving Dependencies
--> Running transaction check
---> Package squid.x86_64 7:3.2.13-1.fc18 will be installed
--> Finished Dependency Resolution
Dependencies Resolved
================================================================================
 Package         Arch     Version                   Repository       Size
================================================================================
Installing:
 squid           x86_64 7:3.2.13-1.fc18             updates          2.5 M
Transaction Summary
================================================================================
Install  1 Package
```

```
Total download size: 2.5 M
Installed size: 8.5 M
Is this ok [y/N]: y
Downloading Packages:
squid-3.2.13-1.fc18.x86_64.rpm                    | 2.5 MB  00:00:46
Running Transaction Check
Running Transaction Test
Transaction Test Succeeded
Running Transaction
  Installing : 7:squid-3.2.13-1.fc18.x86_64                 1/1
  Verifying  : 7:squid-3.2.13-1.fc18.x86_64                 1/1

Installed:
  squid.x86_64  7:3.2.13-1.fc18
Complete!
[root@computer-a ~]#
```

After the successful installation we need to perform steps to enable the squid service and start the squid service. We will perform these steps as examples below.

Let us first check the status of the squid service on *computer-a*.

```
[root@computer-a ~]# systemctl status squid.service
squid.service - Squid caching proxy
   Loaded: loaded (/usr/lib/systemd/system/squid.service; disabled)
   Active: inactive (dead)

Sep 19 12:31:56 computer-a systemd[1]: Stopped Squid caching proxy.
[root@computer-a ~]#
```

You can see that the squid service is disabled and inactive. Therefore, we need to enable the squid service prior to starting it.

```
[root@computer-a ~]# systemctl enable squid.service
ln -s '/usr/lib/systemd/system/squid.service' '/etc/systemd/system/multi-user.target.wants/squid.service'
[root@computer-a ~]#
```

Now, when we check the status of the squid service we see that its status has changed to enabled. However, its still inactive and we need to start it.

```
[root@computer-a ~]# systemctl status squid.service
squid.service - Squid caching proxy
   Loaded: loaded (/usr/lib/systemd/system/squid.service; enabled)
   Active: inactive (dead)
Sep 19 12:31:56 computer-a systemd[1]: Stopped Squid caching proxy.
[root@computer-a ~]#
```

Let us start the squid service.

```
[root@computer-a ~]# systemctl start squid.service
[root@computer-a ~]#
```

Now, when we check the status of the squid service we can see that it is enabled and active.

```
[root@computer-a ~]# systemctl status squid.service
squid.service - Squid caching proxy
   Loaded: loaded (/usr/lib/systemd/system/squid.service; enabled)
   Active: active (running) since Thu 2013-09-19 14:47:07 IST; 3s ago
  Process: 5568 ExecStart=/usr/sbin/squid $SQUID_OPTS -f $SQUID_CONF (code=exited, status=0/SUCCESS)
  Process: 5562 ExecStartPre=/usr/libexec/squid/cache_swap.sh (code=exited, status=0/SUCCESS)
 Main PID: 5571 (squid)
   CGroup: name=systemd:/system/squid.service
           ├─5571 /usr/sbin/squid -f /etc/squid/squid.conf
           ├─5573 (squid-1) -f /etc/squid/squid.conf
           └─5574 (logfile-daemon) /var/log/squid/access.log

Sep 19 14:47:07 computer-a systemd[1]: Starting Squid caching proxy...
Sep 19 14:47:07 computer-a squid[5571]: Squid Parent: will start 1 kids
Sep 19 14:47:07 computer-a squid[5571]: Squid Parent: (squid-1) process 5573 started
Sep 19 14:47:07 computer-a systemd[1]: Started Squid caching proxy.
[root@computer-a ~]#
```

We can also check the system log for any issues if you want to be double sure :-).

```
[root@computer-a ~]# tail -f /var/log/messages
Sep 19 14:47:07 computer-a systemd[1]: Starting Squid caching proxy...
Sep 19 14:47:07 computer-a squid[5571]: Squid Parent: will start 1 kids
Sep 19 14:47:07 computer-a squid[5571]: Squid Parent: (squid-1) process 5573 started
```

```
Sep 19 14:47:07 computer-a systemd[1]: Started Squid caching proxy.
```

On Computer B (Ubuntu 12.04.2 LTS)

```
root@computer-b:~# apt-get install squid
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following NEW packages will be installed:
  squid
0 upgraded, 1 newly installed, 0 to remove and 230 not upgraded.
Need to get 6,254 B of archives.
After this operation, 128 kB of additional disk space will be used.
Get:1 http://in.archive.ubuntu.com/ubuntu/ precise-updates/universe squid amd64 3.1.19-1ubuntu3.12.04.2 [6,254 B]
Fetched 6,254 B in 1s (5,543 B/s)
Selecting previously unselected package squid.
(Reading database ... 201082 files and directories currently installed.)
Unpacking squid (from .../squid_3.1.19-1ubuntu3.12.04.2_amd64.deb) ...
Setting up squid (3.1.19-1ubuntu3.12.04.2) ...
root@computer-b:~#
```

After the successful installation of the squid proxy we need to check if its running.

```
root@computer-b:~# service squid3 status
squid3 start/running, process 1261
root@computer-b:~#
```

You can see that on Ubuntu 12.04.2 LTS the installation process took care of installing, enabling and starting to squid proxy for us.

We now have Squid Version 3.2.13 installed on Fedora 18 and Squid Version 3.1.19 installed on Ubuntu 12.04.2.

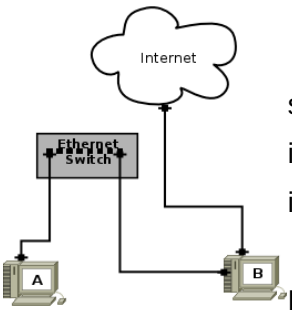
5.4.2 CONFIGURATION

For testing our Squid proxy server we do require a working Internet connection on the computer where our Squid proxy is installed. Now, there could be several

situations but for our example, since all we need is a simple proxy server, we will consider only two situations 1) Squid proxy server is directly connected to the Internet by whatever means and 2) Squid proxy server connects to the Internet via another proxy server.

The Squid configuration is located in the file "/etc/squid/squid.conf" on Fedora 18 and in the file "/etc/squid3/squid.conf" on Ubuntu 12.04.2 LTS.

Case 1: When the Squid proxy server is directly connected to the Internet.



It should pretty much work out of the box after a successful squid installation if the system running the Squid proxy is directly connected to the Internet. Nevertheless, we will look into the squid.conf file and see some of the configuration options.

Let us open the "squid.conf" file using the a text editor and look for the line with the entry "**acl localnet src 192.168.0.0/16**" and uncomment this line. The "#" character is used to comment lines within the "squid.conf" file. Therefore, by just deleting the "#" character in the beginning of a line you can uncomment a line. Fedora 18 has this line by default uncommented while on Ubuntu 12.04.2 LTS you need to uncomment this line manually. This line is an access control list or **acl** defining **localnet** as the source or **src** with IP Addresses in the range **192.168.0.0/16**. This will allow computers on the 192.168.0.0/16 network (our example LAN) to be able to use this proxy server.

Next, look for the line with the entry "**http_access allow localnet**" and uncomment it if commented. This line basically defines who should be allowed to use the proxy service. In our case this configuration would allow **http_access** to hosts on the **localnet**.

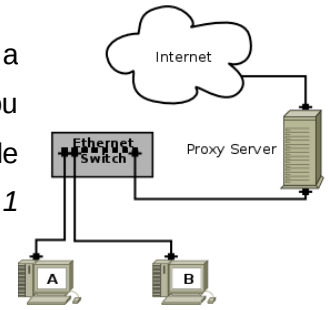
The line with the entry "**http_port 3128**" defines the port on which the http requests will be handled. You can change this port however we will go with the defaults in our example.

Restart the squid service using the command "**systemctl restart squid.service**" on Fedora 18 or "**service squid3 status**" on Ubuntu 12.04.2 LTS to apply all the changes you made to the "squid.conf" file.

Thats it, we should now be able to use our Squid proxy server with our direct Internet connection.

Case 2: When the Squid proxy server connects to the Internet via another proxy server.

In cases when your Squid proxy server does not have a direct Internet connection but uses another proxy server you really have to just add a couple of lines to the "squid.conf" file as shown below, in addition to the ones mentioned in *Case 1* above.



[If the proxy allows anonymous access]
cache_peer *parent-proxy-ip* **parent** *parent-proxy-port* **0 no-query default**
never_direct allow all

For example:
cache_peer *192.168.1.1* **parent** *3128* **0 no-query default**
never_direct allow all

[If the proxy requires a username and a password for access]
cache_peer *parent-proxy-ip* **parent** *parent-proxy-port* **0 no-query login=***username***:***password*
never_direct allow all

For example:
cache_peer *192.168.1.1* **parent** *3128* **0 no-query login=***binod.deka***:***abcd1234*
never_direct allow all

We should now be able to use our Squid proxy server via another proxy. That is via another Squid proxy server in our example. Of course we need to restart the squid service using the command "**systemctl restart squid.service**" on Fedora 18 or

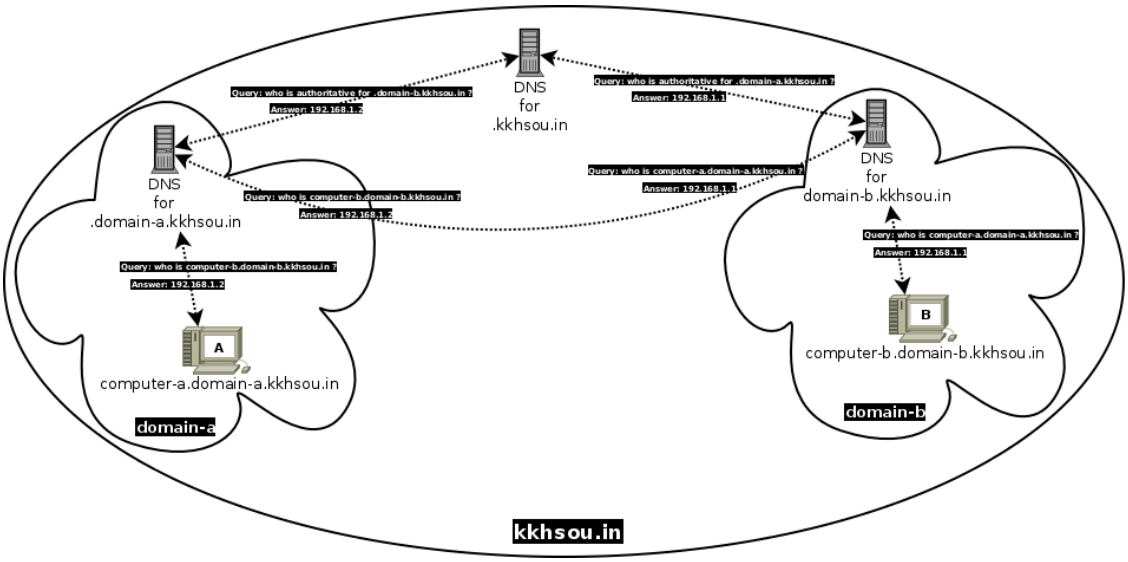
"**service squid3 status**" on Ubuntu 12.04.2 LTS to apply all the changes you made to the "squid.conf" file.

To use this proxy server we just configured, specify the proxy server as either "**192.168.1.1**" with port "**3128**" or "**192.168.1.2**" with port "**3128**" as we have installed on both *computer-a* as well as on *computer-b*. Another thing to note if you have a firewall running on your system is to allow the port "3128" or else other computers on the network would not be able to connect to the Squid Proxy on that port.

Please note that the Squid proxy server is a very powerful proxy server and there are a lot more configuration options that we have not discussed as its out of the scope of this unit. However, our goal was to configure a simple proxy server using Squid and we have done so.

5.5 INSTALLATION AND CONFIGURATION OF A DNS SERVER – BIND

Domain Name System or DNS is a service which provides resolution of fully qualified domain names (FQDN) into IP addresses and vice-versa. What this means is that domain names like say "www.kkhsou.in" is easier to remember than some IP address like "182.50.130.66". Moreover, IP addresses may change. Just think how difficult it would be to keep remembering numerical addresses. BIND (Berkeley Internet Name Domain) is the most widely used DNS software on the Internet and we will install and configure it in this section. DNS servers are also known as a nameservers as they provide a network service that associates hostnames with their respective IP addresses. DNS is usually implemented using one or more centralized servers that are authoritative for certain domains. When a client host requests information from a nameserver, it usually connects to port 53. The nameserver then attempts to resolve the name requested. If it does not have an authoritative answer, or does not already have the answer cached from an earlier query, it queries other nameservers, called root nameservers, to determine which nameservers are authoritative for the name in question, and then queries them to get the requested name.



We will setup the DNS servers more-or-less as depicted in the figure above. However, we will not be actually doing this over the Internet and neither will we be making any modifications on the DNS server for “kkhsou.in”. The figure above is only for an understanding of how DNS might be setup and work on the Internet. Since, in our current setup we only have two computers, *computer-a* and *computer-b* will both act as the DNS Server and the DNS Client for their respective domains.

5.5.1 INSTALLATION

On Computer A (Fedora 18)

```
[root@computer-a ~]# yum install bind
Loaded plugins: langpacks, presto, refresh-packagekit
Resolving Dependencies
--> Running transaction check
---> Package bind.x86_64 32:9.9.3-4.P2.fc18 will be installed
--> Finished Dependency Resolution

Dependencies Resolved

================================================================================
 Package         Arch     Version                    Repository        Size
================================================================================
```

```
Installing:
 bind               x86_64 32:9.9.3-4.P2.fc18         updates            2.1 M

Transaction Summary
================================================================================
Install  1 Package

Total download size: 2.1 M
Installed size: 6.2 M
Is this ok [y/N]: y
Downloading Packages:
bind-9.9.3-4.P2.fc18.x86_64.rpm                                | 2.1 MB  00:00:04
Running Transaction Check
Running Transaction Test
Transaction Test Succeeded
Running Transaction
  Installing : 32:bind-9.9.3-4.P2.fc18.x86_64        1/1
  Verifying  : 32:bind-9.9.3-4.P2.fc18.x86_64        1/1

Installed:
  bind.x86_64    32:9.9.3-4.P2.fc18
Complete!
[root@beefy-miracle ~]#
```

BIND is installed as the **named** service. Check the status of the named service.

```
[root@computer-a ~]# systemctl status named
named.service - Berkeley Internet Name Domain (DNS)
   Loaded: loaded (/usr/lib/systemd/system/named.service; disabled)
   Active: inactive (dead)
[root@computer-a ~]#
```

Enable the named service.

```
[root@computer-a ~]# systemctl enable named
ln -s '/usr/lib/systemd/system/named.service' '/etc/systemd/system/multi-user.target.wants/named.service'
[root@computer-a ~]#
```

Start the named service.

```
[root@computer-a ~]# systemctl start named
```

Stop the named service.

```
[root@computer-a ~]# systemctl stop named
```

On Computer B (Ubuntu 12.04.2 LTS)

```
root@computer-b:~# apt-get install bind9
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following extra packages will be installed:
  bind9utils
Suggested packages:
  bind9-doc
The following NEW packages will be installed:
  bind9 bind9utils
0 upgraded, 2 newly installed, 0 to remove and 6 not upgraded.
Need to get 455 kB of archives.
After this operation, 1,269 kB of additional disk space will be used.
Do you want to continue [Y/n]? Y
WARNING: The following packages cannot be authenticated!
  bind9utils bind9
Install these packages without verification [y/N]? y
Get:1 http://in.archive.ubuntu.com/ubuntu/ precise-updates/main bind9utils amd64
1:9.8.1.dfsg.P1-4ubuntu0.7 [108 kB]
Get:2 http://in.archive.ubuntu.com/ubuntu/ precise-updates/main bind9 amd64 1:9.8.1.dfsg.P1-
4ubuntu0.7 [347 kB]
Fetched 455 kB in 7s (57.6 kB/s)
Preconfiguring packages ...
Selecting previously unselected package bind9utils.
(Reading database ... 50965 files and directories currently installed.)
Unpacking bind9utils (from .../bind9utils_1%3a9.8.1.dfsg.P1-4ubuntu0.7_amd64.deb) ...
Selecting previously unselected package bind9.
Unpacking bind9 (from .../bind9_1%3a9.8.1.dfsg.P1-4ubuntu0.7_amd64.deb) ...
Processing triggers for man-db ...
Processing triggers for ufw ...
Processing triggers for ureadahead ...
```

```
Setting up bind9utils (1:9.8.1.dfsg.P1-4ubuntu0.7) ...
Setting up bind9 (1:9.8.1.dfsg.P1-4ubuntu0.7) ...
Adding group `bind' (GID 116) ...
Done.
Adding system user `bind' (UID 108) ...
Adding new user `bind' (UID 108) with group `bind' ...
Not creating home directory `/var/cache/bind'.
wrote key file "/etc/bind/rndc.key"
#
 * Starting domain name service... bind9                    [ OK ]
root@computer-b:~#
```

BIND is installed as **bind9**. Start the bind9 service.

```
[root@computer-b ~]# service bind9 start
```

Stop the bind9 service.

```
[root@computer-b ~]# service bind9 stop
```

Additionally, you should also install "bind-utils" on Fedora 18 and "dnsutils" on Ubuntu 12.04.2 LTS which includes the client tools like nslookup, dig and host, etc. We will use these tools to test our DNS server configurations.

5.5.2 CONFIGURATION

In a DNS server such as BIND, all information is stored in basic data elements called *resource records* (RR). The resource record is usually a fully qualified domain name (FQDN) of a host. The following are examples of resource records.

computer-a.domain-a.kkhsou.in

computer-b.domain-b.kkhsou.in

Each level of the hierarchy is divided by a period (that is, .). In the examples above, "**in**" defines the top-level domain, "**kkhsou**" its subdomain, and "**domain-a**" & "**domain-b**" the subdomain of "kkhsou". In our example, "**computer-a**" & "**computer-b**" identifies a *resource record* that is part of the "domain-a.kkhsou.in" & "domain-b.kkhsou.in" domains respectively. With the exception of the part furthest to the left (that is, computer-a and computer-b), each of these sections is called a **zone** and defines a specific *namespace*.

Zones are defined on authoritative nameservers through the use of "*zone files*", which contain definitions of the resource records in each zone. Zone files are stored on primary nameservers (also called master nameservers), where changes are made to the files, and secondary nameservers (also called slave nameservers), which receive zone definitions from the primary nameservers. Both primary and secondary nameservers are authoritative for the zone and look the same to clients. Depending on the configuration, any nameserver can also serve as a primary or secondary server for multiple zones at the same time. In our examples, we will be setting up a single primary nameserver (authoritative) for each individual domain.

The main configuration file for BIND is located in "/etc/named.conf" for Fedora 18 and “/etc/bind/named.conf” for Ubuntu 12.04.2 LTS. In this section, we will configure simple DNS servers for the example domain “domain-a.kkhsou.in” on computer-a and example domain “domain-b.kkhsou.in” on computer-b.

On computer-a (Fedora 18):

Our first step would be to add our example domain by adding a zone entry in the **named.conf** file.

```
zone "domain-a.kkhsou.in" IN {
  type master;
  file "domain-a.kkhsou.in.zone";
};
```

Specify the IPv4 network interface on which to listen for / allow queries under the options statement.

```
options {
```

```
...
listen-on port 53 { 192.168.1.1; };
allow-query { 192.168.1.0/24; };
...
};
```

The "/var/named/" by default will contain all the zone files stated in the "named.conf" file. In our example, the zone file will be "domain-a.kkhsou.in.zone" and will contain the zone data.

A zone file consists of directives and resource records. Directives tell the nameserver to perform tasks or apply special settings to the zone, resource records define the parameters of the zone and assign identities to individual hosts. While the directives are optional, the resource records are required in order to provide name service to a zone. All directives and resource records should be entered on individual lines. Directives begin with the dollar sign character (that is, $) followed by the name of the directive, and usually appear at the top of the file.

Our example zone file for the "domain-a.kkhsou.in" domain will be say "/var/named/**domain-a.kkhsou.in.zone**".

In a zone file comments are identified by ";" so anything that comes after ";" will be ignored by named.
Let us go ahead and create the zone file "/var/named/domain-a.kkhsou.in.zone" as root and make the following entries.

```
$ORIGIN domain-a.kkhsou.in.
$TTL 86400 ; how long the zone record is valid in seconds. Each resource record can contain its own TTL value, which overrides this directive.
@       IN  SOA  computer-a.domain-a.kkhsou.in.  root.domain-a.kkhsou.in. (
            20130925        ; serial. You must increment this serial number each time you
                            ; make changes to the zone file before restarting the
                            ; named service.
            21600           ; refresh after 6 hours
            3600            ; retry after 1 hour
```

```
            604800          ; expire after 1 week
            86400 )         ; minimum TTL of 1 day
;
;
          IN  NS    computer-a.domain-a.kkhsou.in.
computer-a      IN  A     192.168.1.1
;
;
```

After successfully creating the zone file for domain-a.kkhsou.in domain, we start the "named" service by typing "**systemctl start named.service**" as root. Next, we will need to configure *computer-a* to use this DNS server. To do this we add the following line to the "**/etc/resolve.conf**" file.

```
nameserver 192.168.1.1
```

To check if our DNS setup is working, we can use the **nslookup** utility to query our DNS server.

```
[root@computer-a ~]# nslookup computer-a.domain-a.kkhsou.in
Server:         192.168.1.1
Address:        192.168.1.1#53

Name:  computer-a.domain-a.kkhsou.in
Address: 192.168.1.1

[root@computer-a ~]#
```

We can also use the **dig** command to query our DNS server as well.

```
[root@computer-a ~]# dig computer-a.domain-a.kkhsou.in

; <<>> DiG 9.9.3-rl.13207.22-P2-RedHat-9.9.3-4.P2.fc18 <<>> computer-a.domain-a.kkhsou.in
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 10223
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL: 1
```

```
;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;computer-a.domain-a.kkhsou.in.          IN      A

;; ANSWER SECTION:
computer-a.domain-a.kkhsou.in. 86400 IN      A       192.168.1.1

;; AUTHORITY SECTION:
domain-a.kkhsou.in.     86400   IN      NS      computer-a.domain-a.kkhsou.in.

;; Query time: 0 msec
;; SERVER: 192.168.1.1#53(192.168.1.1)
;; WHEN: Wed Sep 25 11:31:47 IST 2013
;; MSG SIZE  rcvd: 117

[root@computer-a ~]#
```

Let us also add the following entries into our "/var/named/domain-a.kkhsou.in.zone" file. We will be using these entries in our later sections.

```
@       IN  MX    10  mail.domain-a.kkhsou.in.
mail    IN  A     192.168.1.1
;
;
ftp     IN CNAME  computer-a.domain-a.kkhsou.in.
www     IN CNAME  computer-a.domain-a.kkhsou.in.
;
;
```

Though, we find that we have a working DNS server setup, we are not done yet. We have to also setup a **Reverse Name Resolution Zone File** that will be used to resolve IP Addresses to Fully Qualified Domain Names (FQDN). It looks very similar to a standard zone file, except that the *PTR resource records* are used to link the IP addresses to a fully qualified domain name.

Firstly, we need to add the following zone statement to our “/etc/named.conf” file.

```
zone "1.168.192.in-addr.arpa" IN {
  type master;
  file "domain-a.kkhsou.in.rr.zone";
};
```

Note that a reverse name resolution zone requires the first three blocks of the IP address reversed "1.168.192" followed by ".in-addr.arpa". This allows the single block of IP numbers used in the reverse name resolution zone file to be associated with the zone.

Next, we create the reverse resolution zone file "/var/named/**domain-a.kkhsou.in.rr.zone**" and make the following entries.

```
$ORIGIN 1.168.192.in-addr.arpa.
$TTL 86400
@  IN  SOA  computer-a.domain-a.kkhsou.in.  root.domain-a.kkhsou.in. (
      2001062501          ; serial
      21600        ; refresh after 6 hours
      3600                ; retry after 1 hour
      604800       ; expire after 1 week
      86400 )      ; minimum TTL of 1 day
;
@  IN  NS   computer-a.domain-a.kkhsou.in.
;
1  IN  PTR  computer-a.domain-a.kkhsou.in.
;
```

After performing the required changed restart the named service.

```
[root@computer-a ~]# systemctl restart named.service
```

We can check to see if the reverse resolution is working using **nslookup** or **dig**.

```
[root@computer-a ~]# nslookup 192.168.1.1
Server:         192.168.1.1
Address:        192.168.1.1#53

1.1.168.192.in-addr.arpa        name = computer-a.domain-a.kkhsou.in.

[root@computer-a ~]#
```

```
[root@computer-a ~]# dig -x 192.168.1.1

; <<>> DiG 9.9.3-rl.13207.22-P2-RedHat-9.9.3-4.P2.fc18 <<>> -x 192.168.1.1
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 48303
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL: 2

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;1.1.168.192.in-addr.arpa.          IN      PTR

;; ANSWER SECTION:
1.1.168.192.in-addr.arpa. 86400 IN      PTR     computer-a.domain-a.kkhsou.in.

;; AUTHORITY SECTION:
1.168.192.in-addr.arpa. 86400   IN      NS      computer-a.domain-a.kkhsou.in.

;; ADDITIONAL SECTION:
computer-a.domain-a.kkhsou.in. 86400 IN         A       192.168.1.1

;; Query time: 0 msec
;; SERVER: 192.168.1.1#53(192.168.1.1)
;; WHEN: Wed Sep 25 12:53:50 IST 2013
;; MSG SIZE  rcvd: 126

[root@computer-a ~]#
```

On computer-b (Ubuntu 12.04.2 LTS):

Similarly, on Ubuntu 12.04.2 LTS we configure the DNS server. Once thing to keep in mind though, is the location of the files. They differ in the location from that of Fedora 18.

The DNS server options need to be put in the file "**/etc/bind/named.conf.options**" .

```
options {
```

```
...
listen-on { 192.168.1.2; };
allow-query { 192.168.1.0/24; };
...
};
```

The zone statements have to be entered in "**/etc/bind/named.conf.default-zones**" .

```
zone "domain-b.kkhsou.in" IN {
  type master;
  file "/etc/bind/db.domain-b.kkhsou.in";
};

zone "1.168.192.in-addr.arpa" {
        type master;
        file "/etc/bind/db.1.168.192";
};
```

Enter the resource records to the file "**/etc/bind/db.domain-b.kkhsou.in**" .

```
$ORIGIN domain-b.kkhsou.in.
$TTL 86400 ; how long the zone record is valid in seconds. Each resource record can contain its own TTL value, which overrides this directive.
@       IN  SOA  computer-b.domain-a.kkhsou.in.  root.domain-b.kkhsou.in. (
             20130925        ; serial. You must increment this serial number each time you
                             ; make changes to the zone file before restarting the
                             ; named service.
             21600           ; refresh after 6 hours
             3600            ; retry after 1 hour
             604800          ; expire after 1 week
             86400 )         ; minimum TTL of 1 day
;
;
          IN  NS     computer-b.domain-b.kkhsou.in.
computer-b     IN  A      192.168.1.2
;
;
@       IN  MX     10  mail.domain-b.kkhsou.in.
```

```
mail     IN  A     192.168.1.2
;
;
ftp      IN CNAME  computer-b.domain-b.kkhsou.in.
www      IN CNAME  computer-b.domain-b.kkhsou.in.
;
;
```

Add the reverse resource records to the file "**/etc/bind/db.1.168.192**" .

```
$ORIGIN 1.168.192.in-addr.arpa.
$TTL 86400
@  IN  SOA  computer-b.domain-b.kkhsou.in.  root.domain-b.kkhsou.in. (
      2001062501          ; serial
      21600               ; refresh after 6 hours
      3600                ; retry after 1 hour
      604800              ; expire after 1 week
      86400 )             ; minimum TTL of 1 day
;
@  IN  NS   computer-b.domain-b.kkhsou.in.
;
1  IN  PTR  computer-b.domain-b.kkhsou.in.
;
```

After performing the required changed restart the bind9 service.

```
[root@computer-b ~]# service bind9 restart
```

We should now have fully functional DNS servers for both the “domain-a.kkhsou.in” and “domain-b.kkhsou.in” domains. Do note here that we have configured a simple DNS server for our example purposes only. There are other advanced configurations that we did not touch upon in this section due to the scope of this unit. You are therefore encouraged to read about these advanced options before attempting to configure any DNS server either for the Internet or for your live networks.

5.6 INSTALLATION AND CONFIGURATION OF A WEB SERVER – APACHE

The Apache HTTP Server, is a robust, full-featured open source web server developed by the Apache Software Foundation and is one of the most widely used web server software that is currently used on the Internet. In this section we will install and configure a simple Web server on both the Fedora 18 and Ubuntu 12.04.2 LTS operating systems using Apache 2 in our examples.

5.6.1 INSTALLATION

On computer-a (Fedora 18):

To install the Apache Web server (httpd), as root type in the command.

```
yum install httpd
```

To enable to httpd service type the command as root.

```
systemctl enable httpd.service
```

To disable to httpd service type the command as root.

```
systemctl disable httpd.service
```

To start the httpd service type the command as root.

```
systemctl start httpd.service
```

To stop the httpd service type the command as root.

```
systemctl stop httpd.service
```

To restart the httpd service type the command as root.

```
systemctl restart httpd.service
```

On computer-b (Ubuntu 12.04.2 LTS):

To install the Apache Web server (apache2), as root type the command.

```
apt-get install apache2
```

To start the apache2 service, as root type the command.

```
service apache2 start
```

To stop the apache2 service, as root type the command.

```
service apache2 stop
```

To restart the apache2 service, as root type the command.

```
service apache2 restart
```

5.6.2 CONFIGURATION

On computer-a (Fedora 18):

The main configuration file is located at "**/etc/httpd/conf/httpd.conf**".

The line below sets the default port where httpd would listen for connections.

```
Listen 80
```

The line below sets the admin email.

```
ServerAdmin webmaster@domain-a.kkhsou.in
```

The line below sets the web server name.

```
ServerName www.domain-a.kkhsou.in:80
```

The line below sets the location of the document root of the web server.

```
DocumentRoot "/var/www/html"
```

The configuration section below sets the file that Apache will serve if a directory is requested.

```
<IfModule dir_module>
    DirectoryIndex index.php index.html
</IfModule>
```

To check the configuration for possible errors, type the following as root.

```
service httpd configtest
```

Once you are satisfied with your configuration you need to either restart the httpd service or alternatively reload the configuration for the httpd service to use.

To restart the httpd service completely, type the following as root.

```
systemctl restart httpd.service
```

To only reload the configuration, type the following as root.

```
systemctl reload httpd.service
```

If there are no errors and the httpd service started properly you should be able to open your web browser to the default webpage page as shown in the screenshot below for **localhost**.

You should now also be able to open similarly the default webpage when using www.domain-a.kkhsou.in instead of localhost from your web browser.

On computer-b (Ubuntu 12.04.2 LTS):

Ubuntu 12.04.2 LTS uses a slightly different approach for configuring the apache2 service.

The main configuration file is located at "**/etc/apache2/apache2.conf**". However, we do not require to edit this file, in our example.

The line below in the file "**/etc/apache2/ports.conf**" sets the default port where apache2 would listen for connections.

```
Listen 80
```

To setup our example site for "www.domain-b.kkhsou.in" we will need to edit the file located in "**/etc/apache2/sites-enabled/000-default** " and add the following entry to set the admin email. And we will leave rest of the configurations intact for our example purposes.

```
<VirtualHost *:80>
        ServerAdmin webmaster@domain-b.kkhsou.in
</VirtualHost>
```

The configuration section in the file "**/etc/apache2/mods-enabled/dir.conf**" shown below, sets the file that Apache will serve if a directory is requested.

```
<IfModule mod_dir.c>
        DirectoryIndex index.html index.cgi index.pl index.php index.xhtml index.htm
</IfModule>
```

Once you are satisfied with your configuration you need to either restart the apache2 service before being able to use with the new configuration.

To restart the apache2 service, type the following as root.

```
service apache2 restart
```

To stop the apache2 service, type the following as root.

```
service apache2 stop
```

To start the apache2 service, type the following as root.

```
service apache2 start
```

If there are no errors and the apache2 service started properly you should be able to open your web browser to the default webpage page as shown in the screenshot below for **localhost**. You should also be able to open similarly the default webpage when using www.domain-b.kkhsou.in instead of localhost from your web browser.

Though we have a simple working Apache Web server, which was our goal in this section. There are an exhaustive number of options available with the Apache Web server that can be used with any custom configuration requirements. You should read all about these configuration options first and ensure that you understand them, before attempting to configure a web server on the Internet or on any live network.

5.7 INSTALLATION AND CONFIGURATION OF A FILE SERVER – SAMBA

Samba is an open source implementation of the Server Message Block (SMB) protocol. It allows the networking of Microsoft Windows, Linux, UNIX, and other operating systems together, enabling access to Windows-based file and printer shares. Samba's use of SMB allows it to appear as a Windows server to Windows clients.

Samba is comprised of three daemons (smbd, nmbd, and winbindd). The **smbd** server daemon provides file sharing and printing services to Windows clients and the default ports on which the server listens for SMB traffic are **TCP ports 139 and 445**. The **nmbd**

server daemon understands and replies to NetBIOS name service requests and the default port that the server listens to for NMB traffic is **UDP port 137**. The winbind service resolves user and group information on a server running Windows NT, 2000, 2003 or Windows Server 2008 and makes Windows user / group information understandable by UNIX platforms. However, winbind is beyond the scope of this section and therefore we will not be discussing it in this section.

In this section we will install and configure a simple file server using **samba** on both the Fedora 18 and Ubuntu 12.04.2 LTS operating systems in our examples.

5.7.1 INSTALLATION

On computer-a (Fedora 18):

To install samba server, type the following as root.

```
[root@computer-a ~]# yum install samba
Loaded plugins: langpacks, presto, refresh-packagekit
Resolving Dependencies
--> Running transaction check
---> Package samba.x86_64 2:4.0.9-1.fc18 will be installed
--> Finished Dependency Resolution

Dependencies Resolved

================================================================================
 Package         Arch     Version             Repository       Size
================================================================================
Installing:
 samba           x86_64 2:4.0.9-1.fc18    updates527 k

Transaction Summary
================================================================================
Install  1 Package

Total download size: 527 k
Installed size: 1.6 M
```

```
Is this ok [y/N]: y
Downloading Packages:
samba-4.0.9-1.fc18.x86_64.rpm                    | 527 kB  00:00:04
Running Transaction Check
Running Transaction Test
Transaction Test Succeeded
Running Transaction
  Installing : 2:samba-4.0.9-1.fc18.x86_64          1/1
  Verifying  : 2:samba-4.0.9-1.fc18.x86_64          1/1

Installed:
  samba.x86_642:4.0.9-1.fc18
Complete!
[root@computer-a ~]#
```

To install the samba client, type the following as root.

```
[root@computer-a ~]# yum install samba-client
Loaded plugins: langpacks, presto, refresh-packagekit
Resolving Dependencies
--> Running transaction check
---> Package samba-client.x86_64 2:4.0.9-1.fc18 will be installed
--> Finished Dependency Resolution

Dependencies Resolved

================================================================================
 Package         Arch     Version            Repository       Size
================================================================================
Installing:
 samba-client    x86_64 2:4.0.9-1.fc18    updates          455 k

Transaction Summary
================================================================================
Install  1 Package

Total download size: 455 k
Installed size: 1.2 M
```

```
Is this ok [y/N]: y
Downloading Packages:
samba-client-4.0.9-1.fc18.x86_64.rpm                | 455 kB  00:00:03
Running Transaction Check
Running Transaction Test
Transaction Test Succeeded
Running Transaction
  Installing : 2:samba-client-4.0.9-1.fc18.x86_64  1/1
  Verifying  : 2:samba-client-4.0.9-1.fc18.x86_64  1/1

Installed:
  samba-client.x86_64 2:4.0.9-1.fc18
Complete!
[root@computer-a ~]#
```

To check the status of the samba services, type the following as root.

```
[root@computer-a ~]# systemctl status smb.service
smb.service - Samba SMB Daemon
   Loaded: loaded (/usr/lib/systemd/system/smb.service; disabled)
   Active: inactive (dead)
```

```
[root@computer-a ~]# systemctl status nmb.service
nmb.service - Samba NMB Daemon
   Loaded: loaded (/usr/lib/systemd/system/nmb.service; disabled)
   Active: inactive (dead)
```

To start the Samba server, type the following command as root.

```
systemctl start smb.service
```

To stop the Samba server, type the following command as root.

```
systemctl stop smb.service
```

To restart the Samba server, type the following command as root.

```
systemctl restart smb.service
```

On computer-b (Ubuntu 12.04.2 LTS):

To install samba, type the following as root.

```
root@computer-b:~# apt-get install samba
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following extra packages will be installed:
  libwbclient0 samba-common smbclient tdb-tools
Suggested packages:
  openbsd-inetd inet-superserver smbldap-tools ldb-tools ctdb cifs-utils
The following NEW packages will be installed:
  samba tdb-tools
The following packages will be upgraded:
  libwbclient0 samba-common smbclient
3 upgraded, 2 newly installed, 0 to remove and 237 not upgraded.
Need to get 22.5 MB of archives.
After this operation, 23.5 MB of additional disk space will be used.
Do you want to continue [Y/n]? Y
Get:1 http://in.archive.ubuntu.com/ubuntu/ precise-updates/main libwbclient0 amd64 2:3.6.3-2ubuntu2.8 [29.9 kB]
Get:2 http://in.archive.ubuntu.com/ubuntu/ precise-updates/main smbclient amd64 2:3.6.3-2ubuntu2.8 [14.1 MB]
Get:3 http://in.archive.ubuntu.com/ubuntu/ precise-updates/main samba-common all 2:3.6.3-2ubuntu2.8 [326 kB]
Get:4 http://in.archive.ubuntu.com/ubuntu/ precise-updates/main samba amd64 2:3.6.3-2ubuntu2.8 [8,049 kB]
Get:5 http://in.archive.ubuntu.com/ubuntu/ precise/main tdb-tools amd64 1.2.9-4 [23.2 kB]
Fetched 22.5 MB in 2min 24s (156 kB/s)
Preconfiguring packages ...
(Reading database ... 201772 files and directories currently installed.)
Preparing to replace libwbclient0 2:3.6.3-2ubuntu2.6 (using .../libwbclient0_2%3a3.6.3-2ubuntu2.8_amd64.deb) ...
Unpacking replacement libwbclient0 ...
Preparing to replace smbclient 2:3.6.3-2ubuntu2.6 (using .../smbclient_2%3a3.6.3-2ubuntu2.8_amd64.deb) ...
Unpacking replacement smbclient ...
Preparing to replace samba-common 2:3.6.3-2ubuntu2.6 (using .../samba-common_2%3a3.6.3-2ubuntu2.8_all.deb) ...
Unpacking replacement samba-common ...
Selecting previously unselected package samba.
Unpacking samba (from .../samba_2%3a3.6.3-2ubuntu2.8_amd64.deb) ...
Selecting previously unselected package tdb-tools.
Unpacking tdb-tools (from .../tdb-tools_1.2.9-4_amd64.deb) ...
Processing triggers for man-db ...
Processing triggers for ureadahead ...
Processing triggers for ufw ...
Rules updated for profile 'Bind9'
```

```
Setting up libwbclient0 (2:3.6.3-2ubuntu2.8) ...
Setting up samba-common (2:3.6.3-2ubuntu2.8) ...
Setting up smbclient (2:3.6.3-2ubuntu2.8) ...
Setting up samba (2:3.6.3-2ubuntu2.8) ...
Generating /etc/default/samba...
Importing account for binoddeka...ok
Importing account for tapashi.kashyap...ok
Importing account for choudhurysmriti...ok
update-alternatives: using /usr/bin/smbstatus.samba3 to provide /usr/bin/smbstatus (smbstatus) in auto mode.
smbd start/running, process 30095
nmbd start/running, process 30129
Setting up tdb-tools (1.2.9-4) ...
update-alternatives: using /usr/bin/tdbbackup.tdbtools to provide /usr/bin/tdbbackup (tdbbackup) in auto mode.
Processing triggers for libc-bin ...
ldconfig deferred processing now taking place
root@computer-b:~#
```

To check the status of the samba services, type the following as root.

```
[root@computer-b ~]# service smbd status
smbd start/running, process 30095
```

```
[root@computer-b ~]# service nmbd status
nmbd start/running, process 30129
```

To start the samba server, type the following as root.

```
service smbd start
```

To stop the samba server, type the following as root.

```
service smbd stop
```

To restart the samba server, type the following as root.

```
service smbd restart
```

5.7.2 CONFIGURATION

On computer-a (Fedora 18):

The default samba configuration file located in "**/etc/samba/smb.conf**" allows users to view their home directories as a Samba share. It also shares all printers configured for the system as Samba shared printers. In other words, you can attach a printer to the system and print to it from the Windows machines on your network.

To specify the Windows workgroup and a brief description of the Samba server, we edit the following lines in our /etc/samba/smb.conf file.

```
#======================= Global Settings ==============================
[global]
# ---------------------- Network-Related Options -------------------------
workgroup = DOMAIN-A
server string = Samba Server Version %v

# --------------------------- Logging Options -----------------------------
log file = /var/log/samba/log.%m
max log size = 50

# ----------------------- Standalone Server Options ------------------------
security = user
passdb backend = tdbsam

# ---------------------------- Printing Options -----------------------------
load printers = yes
cups options = raw
```

To create a Samba share directory we add the following section to our /etc/samba/smb.conf file. In our example, we will allow the users **binoddeka**, **tapashi.kashyap** and **choudhurysmriti** to be able to access the share "**allusers**".

```
#========================= Share Definitions ========================
[allusers]
  comment = All Users
```

```
path = /home/shares/allusers
valid users = binoddeka, tapashi.kashyap, choudhurysmriti
create mask = 0660
directory mask = 0771
writable = yes
```

We can use the "testparm" program which checks the syntax of the /etc/samba/smb.conf file. The testparm program also displays a summary of your /etc/samba/smb.conf file and the server's role (stand-alone, domain, etc.) after testing. This is convenient when debugging as it excludes comments and concisely presents information for experienced administrators to read.

```
[root@computer-a ~]# testparm
Load smb config files from /etc/samba/smb.conf
rlimit_max: increasing rlimit_max (1024) to minimum Windows limit (16384)
Processing section "[homes]"
Processing section "[printers]"
Processing section "[allusers]"
Loaded services file OK.
Server role: ROLE_STANDALONE
Press enter to see a dump of your service definitions
[global]
        workgroup = DOMAIN-A
        server string = Samba Server Version %v on computer-a
        log file = /var/log/samba/log.%m
        max log size = 50
        idmap config * : backend = tdb
        cups options = raw
[homes]
        comment = Home Directories
        valid users = MYDOMAIN\%S
        read only = No
        browseable = No
[printers]
        comment = All Printers
        path = /var/spool/samba
        printable = Yes
        print ok = Yes
```

```
        browseable = No
[allusers]
  comment = All Users
  path = /home/shares/allusers
  valid users = binoddeka, tapashi.kashyap, choudhurysmriti
  create mask = 0660
  directory mask = 0771
  writable = yes
[root@computer-a ~]#
```

Next we will use the "smbpasswd" program that manages encrypted passwords. This program can be run by a superuser to change any user's password as well as by an ordinary user to change their own Samba password.

```
[root@computer-a ~]# smbpasswd -a binoddeka
New SMB password:
Retype new SMB password:
Added user binoddeka.
[root@computer-a ~]#
```

Similarly, using the "smbpasswd" program we will need to add the other users as well.

To connect to our share we will use the “smbclient” program, as shown below.

```
[root@computer-a ~]# smbclient //computer-a/allusers -U binoddeka
Enter binoddeka's password:
Domain=[DOMAIN-A] OS=[Unix] Server=[Samba 4.0.9]
smb: \>
```

At the “smb: \>” prompt you can type “?” to view the available commands. To view the files in the “allusers” shared folder you can either type “dir” or “ls”.

```
smb: \> ls
  .                   D        0  Tue Oct  1 15:20:19 2013
  ..                  D        0  Thu Sep 26 17:17:41 2013
  b.txt               N     7480  Tue Oct  1 15:20:19 2013
  a.txt               N     7436  Tue Oct  1 15:20:13 2013
                50380 blocks of size 8388608. 3286 blocks available
smb: \>
```

To exit from the “smb: \>” prompt you can type “quit” and hit the enter key on your keyboard.

On computer-b (Ubuntu 12.04.2 LTS):

The main Samba configuration file is located in “**/etc/samba/smb.conf**”. All the configurations will be similar to the Samba configuration discussed for Fedora 18. However, we will make the modifications with respect to **domain-b** on **computer-b**. So, that our “testparm” program out put will be as shown below.

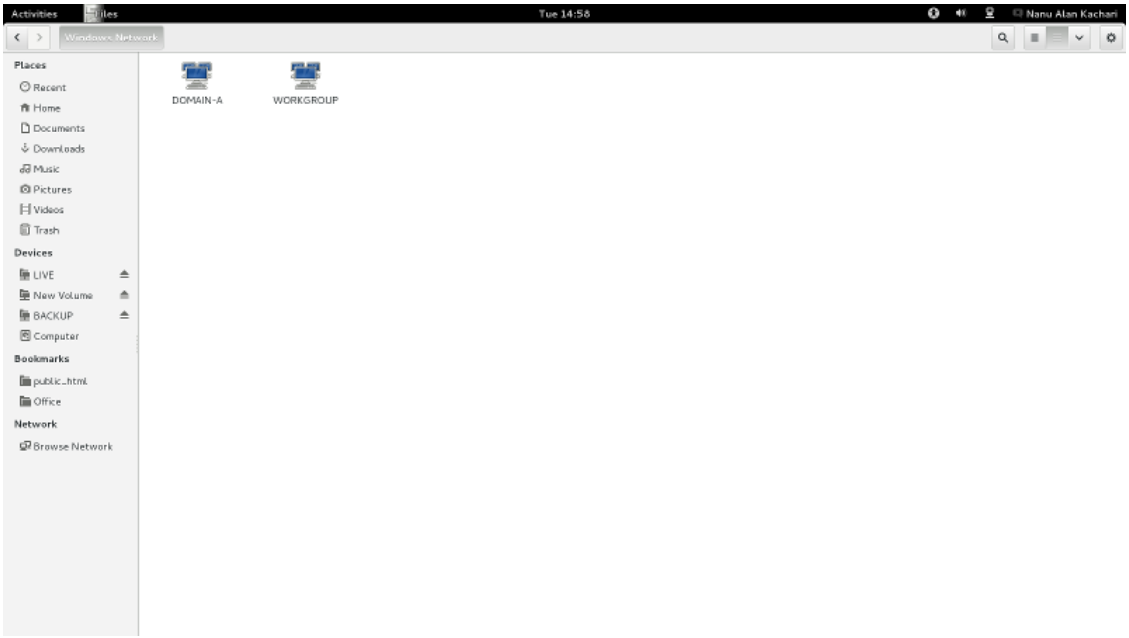
```
root@computer-b:/srv/samba/allusers# testparm
Load smb config files from /etc/samba/smb.conf
rlimit_max: increasing rlimit_max (1024) to minimum Windows limit (16384)
Processing section "[printers]"
Processing section "[print$]"
Processing section "[share]"
Loaded services file OK.
Server role: ROLE_STANDALONE
Press enter to see a dump of your service definitions
[global]
        workgroup = DOMAIN-B
        server string = %h server (Samba, Ubuntu)
        map to guest = Bad User
        obey pam restrictions = Yes
        pam password change = Yes
        passwd program = /usr/bin/passwd %u
        passwd chat = *Enter\snew\s*\spassword:* %n\n *Retype\snew\s*\spassword:* %n\n *password\supdated\ssuccessfully* .
        unix password sync = Yes
        syslog = 0
        log file = /var/log/samba/log.%m
        max log size = 1000
        dns proxy = No
        usershare allow guests = Yes
        panic action = /usr/share/samba/panic-action %d
        idmap config * : backend = tdb
[printers]
        comment = All Printers
```

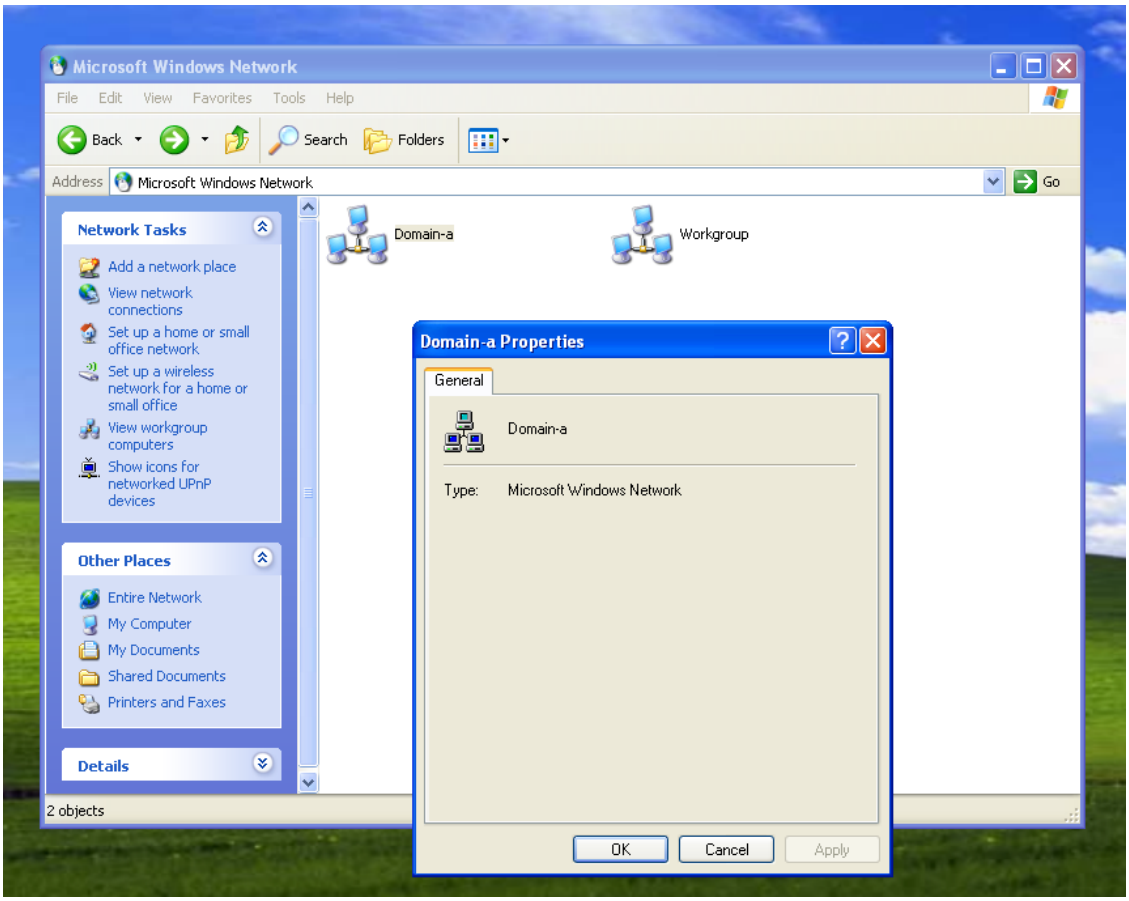
```
        path = /var/spool/samba
        create mask = 0700
        printable = Yes
        print ok = Yes
        browseable = No
[print$]
        comment = Printer Drivers
        path = /var/lib/samba/printers
[share]
        comment = Ubuntu File Server Share
        path = /srv/samba/allusers
        read only = No
        create mask = 0755
        guest ok = Yes
root@computer-b:/srv/samba/allusers#
```

Please do note that we require to restart the Samba Service on both Fedora 18 and Ubuntu 12.04.2 LTS system before we are able to use Samba. Therefore, ensure that after any configuration change to the “smb.conf” file you restart the samba services.

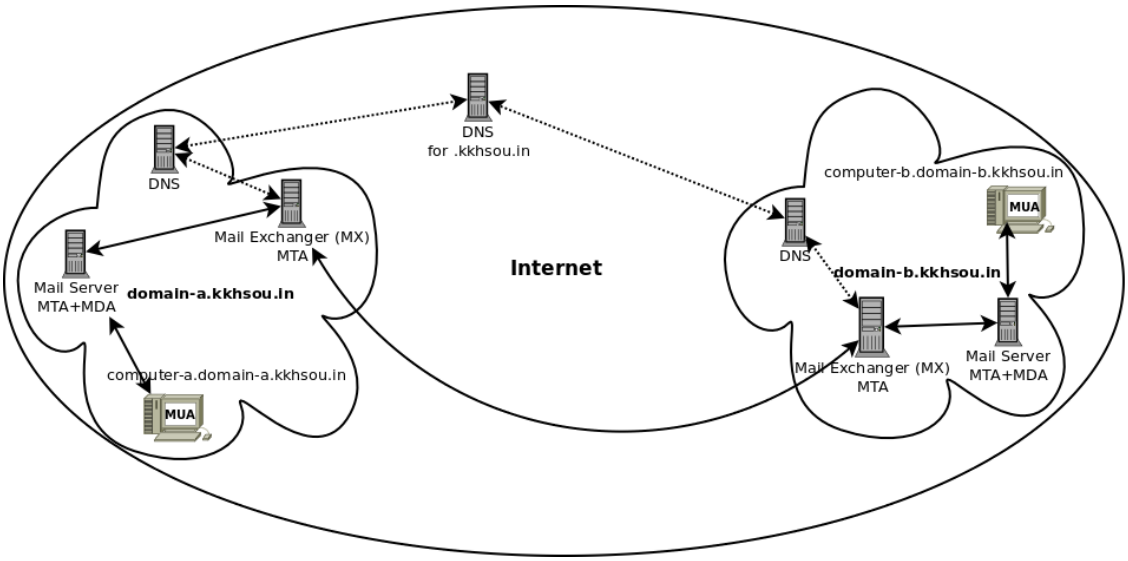
We now are done configuring a simple Samba server and you should be able to browse the shares created from both computers running Windows as well as from our Linux boxes.

Some screen-shots of our Samba configuration browse-able for Domain-A, from both Linux and Windows Clients.


Activities
Tue 14:58
Nanu Alan Kachari
Windows Network
Places
Recent
Home
Documents
Downloads
Music
Pictures
Videos
Trash
Devices
LIVE
New Volume
BACKUP
Computer
Bookmarks
public_html
Office
Network
Browse Network
DOMAIN-A
WORKGROUP

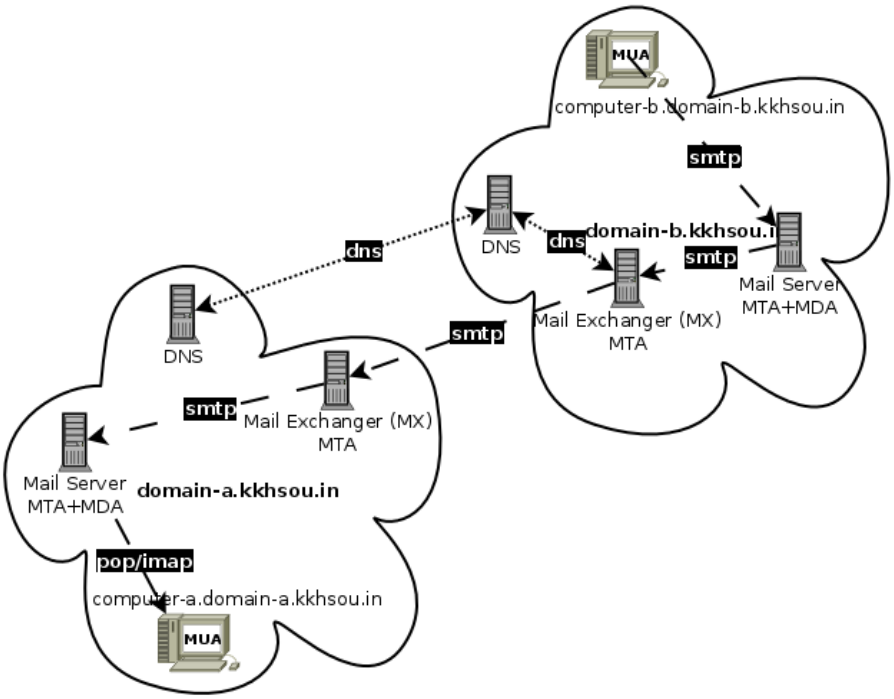

Microsoft Windows Network
File Edit View Favorites Tools Help
Back
Search
Folders
Address
Microsoft Windows Network
Go
Network Tasks
Add a network place
View network connections
Set up a home or small office network
Set up a wireless network for a home or small office
View workgroup computers
Show icons for networked UPnP devices
Other Places
Entire Network
My Computer
My Documents
Shared Documents
Printers and Faxes
Details
Domain-a
Workgroup
2 objects
Domain-a Properties
General
Domain-a
Type:
Microsoft Windows Network
OK
Cancel
Apply

5.8 INSTALLATION AND CONFIGURATION OF A MAIL SERVER – POSTFIX



The process of moving an email from one person to another over a network or the Internet involves many systems working together. Email is delivered using a client/server architecture. A typical mail system is depicted in the figure above, though we are using fictitious domain names here for our understanding.

A sender uses a Mail User Agent (**MUA**), meaning an email client like Evolution, Thunderbird, Mutt, etc., to send the message through one or more Mail Transfer Agents (**MTA**), meaning email servers like Exim, Postfix, Sendmail, etc., and ultimately will hand it off to a Mail Delivery Agent (**MDA**) like Procmail, mail, etc., for delivery to the recipient's mailbox. The recipient's email client will retrieve the email from the mailbox usually via a POP3 or IMAP server like Dovecot. The process of sending and receiving email between our example domains is depicted in the figure below.



Every email domain is required to have an MTA that would by default receive emails for it's domain. This MTA is known as the Mail Exchanger (MX) for the domain. Typically, a domain is required to also have a DNS server which is authoritative for it's domain. This DNS will maintain all the Domain Names for its domain including the MX Record which is the address of the MTA for the domain. Furthermore, the email mailboxes are generally stored on another MTA which could include an MDA as well. In our examples, we will configure all the components of a mail system on a single system.

We will install and configure postfix (as MTA), procmail (as MDA), mutt (as MUA) and dovecot (as pop server) on both Fedora 18 and Ubuntu 12.04.2 LTS operating systems. You will need to ensure that an active Internet connection is available on the system you are performing the installation steps mentioned below, as the software repositories being used for installation are located on the Internet.

5.8.1 INSTALLATION

MTA installation:

To install a Mail Transport Agent (MTA), we choose **postfix** in our examples, type the following as root.

On computer-a (Fedora 18):

```
yum install postfix
```

You can use the following to check the status, enable, start, stop or restart the postfix service.

systemctl status postfix.service
systemctl enable postfix.service
systemctl start postfix.service
systemctl stop postfix.service
systemctl restart postfix.service

On computer-b (Ubuntu 12.04.2 LTS):

```
apt-get install postfix
```

You can use the following to check the status, start, stop or restart the postfix service.

service postfix status
service postfix start
service postfix stop
service postfix restart

MDA installation:

To install the Mail Delivery Agent (MDA), we choose **procmail** in our examples, type the following as root.

On computer-a (Fedora 18):

```
yum install procmail
```

On computer-b (Ubuntu 12.04.2 LTS):

```
apt-get install procmail
```

MUA installation:

To install the Mail User Agent (MUA), we choose **mutt** (which is a text based email client) in our examples, type the following as root.

On computer-a (Fedora 18):

```
yum install mutt
```

On computer-b (Ubuntu 12.04.2 LTS):

```
apt-get install mutt
```

POP/IMAP Server installation:

This is optional and would only be required if you plan to grant access to user mailboxes via pop/imap. To install the **dovecot** (imap/pop) server, type the following as root.

On computer-a (Fedora 18):

```
yum install dovecot
```

You can use the following to check the status, enable, start, stop or restart the dovecot service.

systemctl status dovecot.service
systemctl enable dovecot.service
systemctl start dovecot.service
systemctl stop dovecot.service
systemctl restart dovecot.service

On computer-b (Ubuntu 12.04.2 LTS):

```
apt-get install dovecot-pop3d
```

You can use the following to check the status, start, stop or restart the dovecot service.

service dovecot status
service dovecot start
service dovecot stop
service dovecot restart

5.8.2 CONFIGURATION

MTA configuration:

The postfix configuration parameters are stored in "**/etc/postfix/main.cf**" file. Rather than editing the configuration file directly, you can use the "postconf" command to configure all postfix parameters. Check the man pages "man postconf" for more detailed options. We will not be using postconf for our examples since a simple configuration is sought.

On computer-a (Fedora 18):

For our simple example purpose, edit the "/etc/postfix/main.cf" file and make the following changes. Leave the rest as is default.

```
mydomain = domain-a.kkhsou.in
myorigin = $mydomain

myhostname = computer-a.domain-a.kkhsou.in
mydestination = $myhostname, localhost.$mydomain, localhost

mynetworks = 192.168.1.0/24, 127.0.0.0/8

inet_interfaces = all

home_mailbox = Maildir/
```

After adding the above to the "/etc/postfix/main.cf" file, we need to restart the postfix service for the changes to take effect. Type the following as root.

```
systemctl restart postfix.service
```

On computer-b (Ubuntu 12.04.2 LTS):

To configure postfix, run the command "dpkg-reconfigure postfix" as root or alternatively edit the "**/etc/postfix/main.cf**" and make the following changes as root. Leave the rest as default.

```
myhostname = computer-b.domain-b.kkhsou.in
myorigin = /etc/mailname

mydestination = computer-b.domain-b.kkhsou.in, localhost.domain-b.kkhsou.in, localhost

mynetworks = 127.0.0.0/8 192.168.1.0/24

inet_interfaces = all

home_mailbox = Maildir/
```

After adding the above to the "/etc/postfix/main.cf" file, we need to restart the postfix service for the changes to take effect. Type the following as root.

```
service postfix restart
```

By default Postfix will use mbox for the mailbox format. The "home_mailbox = Maildir/" setting will place new mails in a directory named "Maildir" of the respective users' home directory. We will need to configure our Mail Delivery Agent (MDA) to use the same path.

MDA configuration:

A Mail Delivery Agent (MDA) is invoked by the MTA to file incoming email in the proper user's mailbox. Any program that actually handles a message for delivery to the point where it can be read by an email client application can be considered an MDA. For this reason, some MTAs (such as Sendmail and Postfix) can fill the role of an MDA when they append new email messages to a local user's mail spool file. For our example purposes we will use the default configurations.

MUA configuration:

Since we are using mutt as our MUA, we need to add the following to the "**~/.muttrc**" file, i.e. the .muttrc file within the users' home directory. You will need to create this file if it does not already exist.

```
set mbox_type=Maildir
set folder="~/Maildir"
set mask="!^\\.[^.]"
set mbox="~/Maildir"
set record="+.Sent"
set postponed="+.Drafts"
set spoolfile="~/Maildir"
mailboxes `echo -n "+ "; find ~/Maildir -maxdepth 1 -type d -name ".*" -printf "+'%f' "`
```

POP/IMAP Server configuration:

The main configuration file for dovecot is located at "**/etc/dovecot/dovecot.conf**".

Now add the following to the dovecot configuration specifying the mailbox we are using.

```
mail_location = maildir:~/Maildir
```

Finally, we need to also add our mailserver that we have just configured as an MX Record in our DNS. If you refer to the section of this UNIT which deals with the installation and configuration of BIND as a DNS (Section 5.5), you will notice that we had added an MX for each of our example domains.

If you followed the examples in this section without running into an errors, you should have a working email server. However, prior to deployment in any live environment you should consult the documentation of each of these thoroughly and read about all the security issues related to email. Since, our aim was to setup a simple email system we did not cover most of the configurations.

5.9 INSTALLATION AND CONFIGURATION OF A DHCP SERVER

The Dynamic Host Configuration Protocol (DHCP) is a network protocol that automatically assigns TCP/IP information to client machines. In a DHCP environment each DHCP client connects to the centrally located DHCP server, which provides all

network related configuration including the IP address, gateway, and DNS servers. This reduces the administrative overhead of having to assign network related configuration manually to hosts in a network.

In this section, we will install and configure a simple DHCP server on Fedora 18 and Ubuntu 12.04.2 LTS.

5.9.1 INSTALLATION

On computer-a (Fedora 18):

To install the dhcp server type "**yum install dhcp**", as root.

```
[root@computer-a ~]# yum install dhcp
Loaded plugins: langpacks, presto, refresh-packagekit
updates/18/x86_64/metalink                              | 8.5 kB  00:00:00
updates                                                 | 4.7 kB  00:00:00
updates/primary_db                                      |  11 MB  00:00:28
Resolving Dependencies
--> Running transaction check
---> Package dhcp.x86_64 12:4.2.5-15.fc18 will be installed
--> Finished Dependency Resolution

Dependencies Resolved

================================================================================
 Package          Arch      Version                     Repository         Size
================================================================================
Installing:
 dhcp             x86_64    12:4.2.5-15.fc18   updates             506 k

Transaction Summary
================================================================================
Install  1 Package

Total download size: 506 k
Installed size: 1.4 M
Is this ok [y/N]: y
Downloading Packages:
dhcp-4.2.5-15.fc18.x86_64.rpm                           | 506 kB  00:00:03
```

```
Running Transaction Check
Running Transaction Test
Transaction Test Succeeded
Running Transaction
  Installing : 12:dhcp-4.2.5-15.fc18.x86_64     1/1
  Verifying  : 12:dhcp-4.2.5-15.fc18.x86_64     1/1

Installed:
  dhcp.x86_64 12:4.2.5-15.fc18

Complete!
[root@computer-a ~]#
```

To check the status of the dhcpd service, type the following as root.

```
[root@computer-a ~]# systemctl status dhcpd.service
```

To enable the dhcpd service, type the following as root.

```
[root@computer-a ~]# systemctl enable dhcpd.service
```

To start the dhcpd service, type the following as root.

```
[root@computer-a ~]# systemctl start dhcpd.service
```

To stop the dhcpd service, type the following as root.

```
[root@computer-a ~]# systemctl stop dhcpd.service
```

To restart the dhcpd service, type the following as root.

```
[root@computer-a ~]# systemctl restart dhcpd.service
```

On computer-b (Ubuntu 12.04.2 LTS):

To install the dhcp server type "**apt-get install isc-dhcp-server** ", as root.

```
root@computer-b:~# apt-get install isc-dhcp-server
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following extra packages will be installed:
  isc-dhcp-client isc-dhcp-common
Suggested packages:
  isc-dhcp-server-ldap
The following NEW packages will be installed:
```

```
  isc-dhcp-server
The following packages will be upgraded:
  isc-dhcp-client isc-dhcp-common
2 upgraded, 1 newly installed, 0 to remove and 253 not upgraded.
Need to get 1,066 kB of archives.
After this operation, 1,010 kB of additional disk space will be used.
Do you want to continue [Y/n]? Y
Get:1 http://in.archive.ubuntu.com/ubuntu/ precise-updates/main isc-dhcp-client amd64 4.1.ESV-R4-0ubuntu5.9 [290 kB]
Get:2 http://in.archive.ubuntu.com/ubuntu/ precise-updates/main isc-dhcp-common amd64 4.1.ESV-R4-0ubuntu5.9 [347 kB]
Get:3 http://in.archive.ubuntu.com/ubuntu/ precise-updates/main isc-dhcp-server amd64 4.1.ESV-R4-0ubuntu5.9 [428 kB]
Fetched 1,066 kB in 4s (233 kB/s)
Preconfiguring packages ...
(Reading database ... 201832 files and directories currently installed.)
Preparing to replace isc-dhcp-client 4.1.ESV-R4-0ubuntu5.8 (using .../isc-dhcp-client_4.1.ESV-R4-0ubuntu5.9_amd64.deb) ...
Unpacking replacement isc-dhcp-client ...
Preparing to replace isc-dhcp-common 4.1.ESV-R4-0ubuntu5.8 (using .../isc-dhcp-common_4.1.ESV-R4-0ubuntu5.9_amd64.deb) ...
Unpacking replacement isc-dhcp-common ...
Selecting previously unselected package isc-dhcp-server.
Unpacking isc-dhcp-server (from .../isc-dhcp-server_4.1.ESV-R4-0ubuntu5.9_amd64.deb) ...
Processing triggers for man-db ...
Processing triggers for ureadahead ...
Setting up isc-dhcp-common (4.1.ESV-R4-0ubuntu5.9) ...
Setting up isc-dhcp-client (4.1.ESV-R4-0ubuntu5.9) ...
Setting up isc-dhcp-server (4.1.ESV-R4-0ubuntu5.9) ...
Generating /etc/default/isc-dhcp-server...
isc-dhcp-server start/running, process 31631
isc-dhcp-server6 stop/waiting
root@computer-b:~#
```

To check the status of the dhcpd service, type the following as root.

```
[root@computer-b ~]# service isc-dhcp-server status
```

To start the dhcpd service, type the following as root.

```
[root@computer-b ~]# service isc-dhcp-server start
```

To stop the dhcpd service, type the following as root.

```
[root@computer-b ~]# service isc-dhcp-server stop
```

To restart the dhcpd service, type the following as root.

```
[root@computer-b ~]# service isc-dhcp-server restart
```

5.9.2 CONFIGURATION

On both Fedora and Ubuntu, the default location for the main DHCP configuration file is "**/etc/dhcp/dhcpd.conf**". A sample configuration file can be found at "**/usr/share/doc/dhcp-version/dhcpd.conf.sample**" on Fedora 18 and for Ubuntu 12.04.2 LTS at "**/usr/share/doc/isc-dhcp-server/examples/dhcpd.conf**". You can refer to this file while configuring "/etc/dhcp/dhcpd.conf".

We will configure a simple dhcp server for the example network we have setup.

```
#set the default and the maximum lease time in seconds
default-lease-time 600;
max-lease-time 7200;

#our example network configuration details
option subnet-mask 255.255.255.0;
option broadcast-address 192.168.1.255;
option routers 192.168.1.254;
option domain-name-servers 192.168.1.1, 192.168.1.2;
option domain-search "kkhsou.in";
subnet 192.168.1.0 netmask 255.255.255.0 {
  range 192.168.1.10 192.168.1.20;
}
```

You are required to restart the dhcp service after any changes to the configuration file.

Though we have a working dhcp server using the above configuration. In practice you may require more parameters to be included in the dhcp configuration. Therefore,

you are encouraged to read the man pages “man dhcpd.conf” before attempting to install, configure and use on a live network.

5.10 INSTALLATION AND CONFIGURATION OF A SSH SERVER AND CLIENT

The SSH (Secure Shell) is a protocol which facilitates secure communications between two systems using a client/server architecture and allows users to log into server host systems remotely. Both Fedora 18 and Ubuntu 12.04.2 LTS includes the general OpenSSH package (openssh) as well as the OpenSSH server (openssh-server) and client (openssh-clients) packages. Note that the OpenSSH packages require the OpenSSL package (openssl), which installs several important cryptographic libraries, enabling OpenSSH to provide encrypted communications. The SSH server listens on port 22 by default.

In this section, we will install and configure a simple SSH server "openssh" on Fedora 18 and Ubuntu 12.04.2 LTS.

5.10.1 INSTALLATION

During the Operating System installation, be it Fedora or Ubuntu, the openssh-client will be installed by default. Though the openssh-server may not be installed by default. We will be assuming that the openssh is not installed on our system and perform the installation on both Fedora and Ubuntu.

On computer-a (Fedora 18):

To install openssh type the following, as root.

```
[root@computer-a ~]# yum install openssh openssh-server openssh-clients openssl
```

To check the status of the openssh daemon, type the following as root.

```
[root@computer-a ~]# systemctl status sshd.service
```

To start the openssh daemon, type the following as root.

```
[root@computer-a ~]# systemctl start sshd.service
```

To stop the openssh daemon, type the following as root.

```
[root@computer-a ~]# systemctl stop sshd.service
```

To restart the openssh daemon, type the following as root.

```
[root@computer-a ~]# systemctl restart sshd.service
```

On computer-b (Ubuntu 12.04.2 LTS):

To install openssh type the following, as root.

```
root@computer-b:~# apt-get install openssh-server openssh-client openssl
```

To check the status of the openssh daemon, type the following as root.

```
root@computer-b:~# service ssh status
```

To start the openssh daemon, type the following as root.

```
root@computer-b:~# service ssh start
```

To stop the openssh daemon, type the following as root.

```
root@computer-b:~# service ssh stop
```

To restart the openssh daemon, type the following as root.

```
root@computer-b:~# service ssh restart
```

5.10.2 CONFIGURATION

In most cases, the default configuration for SSH should suffice. However, it is possible to configure the SSH server to suite your requirements.

The default configuration file for the ssh daemon is located at "**/etc/ssh/sshd_config**". Though we will not discuss it at length, you are encouraged to

use the man pages by typing “**man sshd_config**” at the terminal to read about it and have a clear understanding before attempting to configure it on a live system.

You should now be able to connect using any ssh client with the ssh server we just setup.

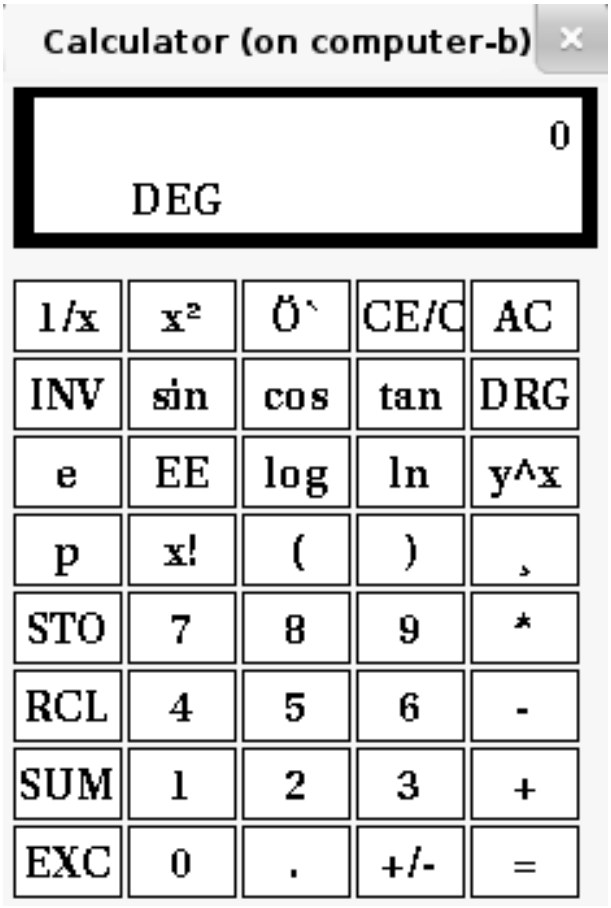
```
[binoddeka@computer-a ~]$ ssh binoddeka@computer-b
binoddeka@computer-b's password:
Last login: Fri Oct  4 21:51:18 2013 from 192.168.1.1
[binoddeka@computer-b ~]$
```

Another interesting option you should know about is the X11 forwarding, which can be used to connect to remote Linux system via ssh and work with the graphical desktop environment applications available on the remote system. To use ssh with this option simple type in the following while initiating the ssh connection.

```
[binoddeka@computer-a ~]$ ssh -Y binoddeka@computer-b
```

Now, for example if you type “xcalc” you should get a pop-up window of the graphical calculator.

```
[binoddeka@computer-b ~]$ xcalc
```



5.11 INSTALLATION AND CONFIGURATION OF A FTP SERVER AND CLIENT

The File Transfer Protocol (FTP) is one of the oldest and most commonly used protocols found on the Internet today. Its purpose is to reliably transfer files between computer hosts on a network without requiring the user to log directly into the remote host or have knowledge of how to use the remote system. It allows users to access files on remote systems using a standard set of simple commands. The ftp server listens on port 21 by default.

In this section, we will install and configure a simple FTP server "vsftpd" on Fedora 18 and Ubuntu 12.04.2 LTS. **vsftpd** (Very Secure FTP Daemon) is a fast, stable and secure FTP server.

5.11.1 INSTALLATION

On computer-a (Fedora 18):

To install **vsftpd** on Fedora 18, type the following as root.

```
[root@computer-a ~]# yum install vsftpd
Loaded plugins: langpacks, presto, refresh-packagekit
Resolving Dependencies
--> Running transaction check
---> Package vsftpd.x86_64 0:3.0.2-2.fc18 will be installed
--> Finished Dependency Resolution

Dependencies Resolved

================================================================
 Package          Arch      Version             Repository        Size
================================================================
Installing:
 vsftpd           x86_64    3.0.2-2.fc18        updates           166 k

Transaction Summary
================================================================
Install  1 Package

Total download size: 166 k
Installed size: 355 k
Is this ok [y/N]: y
Downloading Packages:
vsftpd-3.0.2-2.fc18.x86_64.rpm                          | 166 kB  00:00:02
Running Transaction Check
Running Transaction Test
Transaction Test Succeeded
Running Transaction
  Installing : vsftpd-3.0.2-2.fc18.x86_64          1/1
  Verifying  : vsftpd-3.0.2-2.fc18.x86_64          1/1
```

```
Installed:
  vsftpd.x86_64 0:3.0.2-2.fc18
  Complete!
[root@computer-a ~]#
```

To check the status of the “vsftpd” service, type the following as root.

```
[root@computer-a ~]# systemctl status vsftpd.service
vsftpd.service - Vsftpd ftp daemon
   Loaded: loaded (/usr/lib/systemd/system/vsftpd.service; disabled)
   Active: inactive (dead)
```

To enable the “vsftpd” service, type the following as root.

```
[root@computer-a ~]# systemctl enable vsftpd.service
ln -s '/usr/lib/systemd/system/vsftpd.service' '/etc/systemd/system/multi-user.target.wants/vsftpd.service'
```

To start the “vsftpd” service, type the following as root.

```
[root@computer-a ~]# systemctl start vsftpd.service
```

To stop the “vsftpd” service, type the following as root.

```
[root@computer-a ~]# systemctl stop vsftpd.service
```

To restart the “vsftpd” service, type the following as root.

```
[root@computer-a ~]# systemctl restart vsftpd.service
```

On computer-b (Ubuntu 12.04.2 LTS):

To install **vsftpd** on Ubuntu, type the following as root.

```
root@computer-b:~# apt-get install vsftpd
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following package was automatically installed and is no longer required:
  tdb-tools
Use 'apt-get autoremove' to remove them.
The following NEW packages will be installed:
  vsftpd
0 upgraded, 1 newly installed, 0 to remove and 256 not upgraded.
Need to get 124 kB of archives.
After this operation, 342 kB of additional disk space will be used.
Get:1 http://in.archive.ubuntu.com/ubuntu/ precise/main vsftpd amd64 2.3.5-1ubuntu2 [124 kB]
Fetched 124 kB in 2s (50.7 kB/s)
```

```
Preconfiguring packages ...
Selecting previously unselected package vsftpd.
(Reading database ... 201789 files and directories currently installed.)
Unpacking vsftpd (from .../vsftpd_2.3.5-1ubuntu2_amd64.deb) ...
Processing triggers for man-db ...
Processing triggers for ureadahead ...
Setting up vsftpd (2.3.5-1ubuntu2) ...
vsftpd start/running, process 29005
root@computer-b:~#
```

To check the status of the “vsftpd” service, type the following as root.

```
root@computer-b:~# service vsftpd status
```

To start the “vsftpd” service, type the following as root.

```
root@computer-b:~# service vsftpd start
```

To stop the “vsftpd” service, type the following as root.

```
root@computer-b:~# service vsftpd stop
```

To restart the “vsftpd” service, type the following as root.

```
root@computer-b:~# service vsftpd restart
```

5.11.2 CONFIGURATION

In most cases, the default configuration for **vsftp** should suffice. However, it is possible to configure vsftp to suite your requirements.

On computer-a (Fedora 18):

The main configuration file for vsftpd is located at "**/etc/vsftpd/vsftpd.conf**".

The list of users not allowed to log into vsftpd is located at "**/etc/vsftpd/ftpusers**". By default, this list includes the root, bin, and daemon users, among others.

The "**/etc/vsftpd/user_list**" file can be configured to either deny or allow access to the users listed, depending on whether the userlist_deny directive is set to YES (default) or

NO in /etc/vsftpd/vsftpd.conf. If /etc/vsftpd/user_list is used to grant access to users, the usernames listed must not appear in /etc/vsftpd/ftpusers.

The folder "**/var/ftp/**" contains the files served by vsftpd. It also contains the /var/ftp/pub/ directory for anonymous users. Both directories are world-readable, but writable only by the root user.

You should now be able to connect to the ftp server we just installed. We will be loging in as an anonymous user without a password.

```
[binoddeka@computer-a ~]$ ftp
ftp> open localhost
Connected to localhost (127.0.0.1).
220 (vsFTPd 3.0.2)
Name (localhost:binoddeka): anonymous
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ls -l
227 Entering Passive Mode (127,0,0,1,184,157).
150 Here comes the directory listing.
drwxr-xr-x    2 0        0            4096 Sep 10 07:37 pub
226 Directory send OK.
ftp>
```

On computer-b (Ubuntu 12.04.2 LTS):

The main configuration file for vsftpd is located at "**/etc/vsftpd.conf**".

The "**/etc/ftpusers**" file contains the list of users that are disallowed for FTP access. The default list includes root, daemon, nobody, etc. To disable FTP access for additional users simply add them to the list.

During installation a user “ftp” is created with a home directory “**/srv/ftp**”. This is the default FTP directory.

You should now be able to connect to the ftp server we just installed. We will be loging in as an anonymous user without a password.

```
[binoddeka@computer-b ~]$ ftp localhost
Connected to localhost (127.0.0.1).
220 (vsFTPd 2.3.5)
Name (localhost:binoddeka): anonymous
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ls -la
227 Entering Passive Mode (172,16,24,1,42,218).
150 Here comes the directory listing.
drwxr-xr-x    2 0        133          4096 Oct 07 12:58 .
drwxr-xr-x    2 0        133          4096 Oct 07 12:58 ..
-rw-r--r--    1 0        0               0 Oct 07 12:58 test.txt
226 Directory send OK.
ftp>
```

Since we are setting up a simple ftp server we will go with the default configuration that is available after installation although you should be able to perform custom configurations based on your specific requirements.

CHECK YOUR PROGRESS

Q1.Give three examples of passive components.

Q2.What is Squid proxy server?

Q3.What do you mean by BIND?

Q4.What is a resource record?

Q5.What is Apache web server?