LAB MANUAL OF

ADVANCED COMPUTER NETWORKS LAB
ETCS 457

Maharaja Agrasen Institute of Technology, PSP area,
Sector – 22, Rohini, New Delhi – 110085

(Affiliated to Guru Gobind Singh Indraprastha University, New Delhi)

INDEX OF THE CONTENTS

Introduction to Advanced Computer NetworksLab

This course covers a set of advanced topics in computer networks. The focus is on principles, architectures, and protocols used in modern networked systems. The goals of the course is to build basic networking and understanding of the tradeoffs and existing technology in building large, complex networked systems, and provide concrete experience of the challenges through a series of lab exercises.

2. LAB REQUIREMENTS

H/W Detail		24 Nos.
	Intel i3/C2D Processor/2 GB RAM/500GB HDD/MB/Lan Card/ Key Board/ Mouse/CD Drive/15" Color Monitor/ UPS	
	LaserJet Printer	1 No.
S/W Detail	CentOS/Fedora Linux	

ADVANCED COMPUTER NETWORKS LAB

Paper Code: ETCS-457(ELECTIVE)
Paper: Advanced Computer Network Lab

List of Experiments:
(As prescribed by G.G.S.I.P.U)

1. Configuration and logging to a CISCO Router and introduction to the basic user Interfaces. Introduction

to the basic router configuration and basic commands.

2. Configuration of IP addressing for a given scenario for a given set of topologies.

3. Configure a DHCP Server to serve contiguous IP addresses to a pool of four IP devices with a default

gateway and a default DNS address. Integrate the DHCP server with a BOOTP demon to automatically

serve Windows and Linux OS Binaries based on client MAC address.

4. Configure, implement and debug the following: Use open source tools for debugging and diagnostics.

a. ARP/RARP protocols

b. RIP routing protocols

c. BGP routing

d. OSPF routing protocols

e. Static routes (check using netstat)

5. Configure DNS: Make a caching DNS client, and a DNS Proxy; implement reverse DNS and forward

DNS, using TCP dump/Wireshark characterise traffic when the DNS server is up and when it is down.

6. Configure FTP Server on a Linux/Windows machine using a FTP client/SFTP client characterise file

transfer rate for a cluster of small files 100k each and a video file of 700mb.Use a TFTP client and repeat

the experiment.

7. Configure a mail server for IMAP/POP protocols and write a simple SMTP client in C/C++/Java client to

send and receive mails.

8. Implement Open NMS+ SNMPD for checking Device status of devices in community MIB of a linux

PC. Using yellow pages and NIS/NFS protocols implement Network Attached Storage Controller (NAS).

Extend this to serve a windows client using SMB. Characterise the NAS traffic using wireshark.

NOTE: At least 8 Experiments out of the list must be done in the semester.

3. LIST OF EXPERIMENTS (As prescribed by G.G.S.I.P.U)

1. Configuration and logging to a CISCO Router and introduction to the basic user Interfaces. Introduction to the basic router configuration and basic commands.
2. Configuration of IP addressing for a given scenario for a given set of topologies.
3. Configure a DHCP Server to serve contiguous IP addresses to a pool of four IP devices with a default gateway and a default DNS address. Integrate the DHCP server with a BOOTP demon to automatically serve Windows and Linux OS Binaries based on client MAC address.
4. Configure, implement and debug the following: Use open source tools for debugging and diagnostics.
   a. ARP/RARP protocols
   b. RIP routing protocols
   c. BGP routing
   d. OSPF routing protocols
   e. Static routes (check using netstat)
5. Configure DNS: Make a caching DNS client, and a DNS Proxy; implement reverse DNS and forward DNS, using TCP dump/Wireshark characterise traffic when the DNS server is up and when it is down.
6. Configure FTP Server on a Linux/Windows machine using a FTP client/SFTP client characterise file transfer rate for a cluster of small files 100k each and a video file of 700mb.Use a TFTP client and repeat the experiment.
7. Configure a mail server for IMAP/POP protocols and write a simple SMTP client in C/C++/Java client to send and receive mails.
8. Implement Open NMS+ SNMPD for checking Device status of devices in community MIB of a linux PC. Using yellow pages and NIS/NFS protocols implement Network Attached Storage Controller (NAS).Extend this to serve a windows client using SMB. Characterise the NAS traffic using wireshark.

4. FORMAT OF THE LAB RECORD TO BE PREPARED BY THE STUDENTS

The front page of the lab record prepared by the students should have a cover page as displayed below.

NAME OF THE LAB

Font should be (Size 20", italics bold, Times New Roman)

Faculty name

Student name
Roll No.:
Semester:
Group:

Font should be (12", Times Roman)

Maharaja Agrasen Institute of Technology, PSP Area,
Sector – 22, Rohini, New Delhi – 110085

Font should be (18", Times Roman)

The second page in the record should be the index as displayed below.

LAB NAME
PRACTICAL RECORD

PAPER CODE **:**

Name of the student :

University Roll No. :

Branch :

Section/ Group :

PRACTICAL DETAILS

Experiments according to the lab syllabus prescribed by GGSIPU

Exp. no	Experiment Name	Date of performance	Date of checking	Remarks	Marks

5. MARKING SCHEME FOR THE PRACTICAL EXAMS

There will be two practical exams in each semester.

- Internal Practical Exam
- External Practical Exam

INTERNAL PRACTICAL EXAM

It is taken by the concerned lecturer of the batch.

MARKING SCHEME FOR THIS EXAM IS:

Total Marks: 40

Division of 40 marks is as follows

1. Regularity: 30

    - Performing program in each turn of the lab
    - Attendance of the lab
    - File

2. Viva Voice: 10

NOTE: For the regularity, marks are awarded to the student out of 5 for each experiment performed in the lab and at the end the average marks are giving out of 30.

EXTERNAL PRACTICAL EXAM

It is taken by the concerned lecturer of the batch and by an external examiner. In this exam student needs to perform the experiment allotted at the time of the examination, a sheet will be given to the student in which some details asked by the examiner needs to be written and at the last viva will be taken by the external examiner.

MARKING SCHEME FOR THIS EXAM IS:

Total Marks: 60

Division of 60 marks is as follows

1. Sheet filled by the student:	20
2. Viva Voice:	15
3. Experiment performance:	15
4. File submitted:	10

NOTE:

- Internal marks + External marks = Total marks given to the students
  (40 marks) (60 marks) (100 marks)

- Experiments given to perform can be from any section of the lab.

6. DETAILS OF EACH SECTION

ALONGWITH

EXAMPLES, EXERCISES

&

EXPECTED VIVA QUESTIONS

1. INTRODUCTION

What is Router?

A router is a networking device that forwards data packets between computer networks. Routers perform the "traffic directing" functions on the Internet. A data packet is typically forwarded from one router to another through the networks that constitute the internetwork until it reaches its destination node.

What is DHCP Server?

The **Dynamic Host Configuration Protocol** (**DHCP**) is a standardized network protocol used on Internet Protocol (IP) networks for dynamically distributing network configuration parameters, such as IP addresses for interfaces and services. With DHCP, computers request IP addresses and networking parameters automatically from a DHCP server, reducing the need for a network administrator or a user to configure these settings manually.

What is ARP/RARP?

The**Address Resolution Protocol** (**ARP**) is a telecommunication protocol used for resolution of Internet layer addresses into link layer addresses, a critical function in multiple-access networks.

ARP is used for mapping a network address (e.g. an IPv4 address) to a physical address like an Ethernet address (also named a MAC address). ARP has been implemented with many combinations of network and data link layer technologies, like IPv4, FDDI, X.25, Frame Relay and Asynchronous Transfer Mode (ATM). IPv4 over IEEE 802.3 and IEEE 802.11 is the most common case.

The **Reverse Address Resolution Protocol** (**RARP**) is an obsolete computer networking protocol used by a client computer to request its Internet Protocol (IPv4) address from a computer network, when all it has available is its link layer or hardware address, such as a MAC address. The client broadcasts the request, and does not need prior knowledge of the network topology or the identities of servers capable of fulfilling its request.

RARP requires one or more server hosts to maintain a database of mappings of Link Layer addresses to their respective protocol addresses. Media Access Control (MAC) addresses need to be individually configured on the servers by an administrator. RARP is limited to serving only IP addresses.

What is DNS?

The **Domain Name System** (**DNS**) is a hierarchical decentralized naming system for computers, services, or any resource connected to the Internet or a private network. It associates various information with domain names assigned to each of the participating entities. Most prominently, it translates more readily memorized domain names to the numerical IP addresses needed for the purpose of locating and identifying computer services and devices with the underlying network protocols. By providing a worldwide, distributed directory service, the Domain Name System is an essential component of the functionality of the Internet.

The Domain Name System delegates the responsibility of assigning domain names and mapping those names to Internet resources by designating authoritative name servers for each domain. Network administrators may delegate authority over sub-domains of their allocated name space to other name servers. This mechanism provides distributed and fault tolerant service and was designed to avoid a single large central database.

The Domain Name System also specifies the technical functionality of the database service which is at its core. It defines the DNS protocol, a detailed specification of the data structures and data communication exchanges used in the DNS, as part of the Internet Protocol Suite. Historically, other directory services preceding DNS were not scalable to large or global directories as they were originally based on text files, prominently the HOSTS.TXT resolver. The Domain Name System has been in use since the 1980s.

What is FTP?

The **File Transfer Protocol** (**FTP**) is a standard network protocol used to transfer computer files between a client and server on a computer network.

FTP is built on a client-server model architecture and uses separate control and data connections between the client and the server. FTP users may authenticate themselves with a clear-text sign-in protocol, normally in the form of a username and password, but can connect anonymously if the server is configured to allow it. For secure transmission that protects the username and password, and encrypts the content, FTP is often secured with SSL/TLS (FTPS). SSH File Transfer Protocol (SFTP) is sometimes also used instead, but is technologically different.

What is IMAP?

The **Internet Message Access Protocol** (**IMAP**) is an Internet standard protocol used by e-mail clients to retrieve e-mail messages from a mail server over a TCP/IP connection.

IMAP was designed with the goal of permitting complete management of an email box by multiple email clients, therefore clients generally leave messages on the server until the user explicitly deletes them. An IMAP server typically listens on port number 143. IMAP over SSL (**IMAPS**) is assigned the port number 993.

Virtually all modern e-mail clients and servers support IMAP. IMAP and the earlier POP3 (Post Office Protocol) are the two most prevalent standard protocols for email retrieval,with many webmail service providers such as Gmail and Yahoo Mail also providing support for either IMAP or POP3.

What is POP/SMTP?

The **Post Office Protocol** (**POP**) is an application-layer Internet standard protocol used by local e-mail clients to retrieve e-mail from a remote server over a TCP/IP connection. POP has been developed through several versions, with version 3 (**POP3**) being the last standard in common use before largely being made obsolete by the more advanced IMAP. In POP3, e-mails are downloaded from the server's inbox to your computer. E-mails are available when you are not connected.

Simple Mail Transfer Protocol (**SMTP**) is an Internet standard for electronic mail (email) transmission. SMTP defined in 1982, it was last updated in 2008 with the Extended SMTP additions.

SMTP by default uses TCP port 25. The protocol for mail submission is the same, but uses port 587. SMTP connections secured by SSL, known as SMTPS, default to port 465 (nonstandard, but sometimes used for legacy reasons).

Although electronic mail servers and other mail transfer agents use SMTP to send and receive mail messages, user-level client mail applications typically use SMTP only for sending messages to a mail server for relaying. For retrieving messages, client applications usually use either POP3 or IMAP.

1. **CONFIGURING A DNS SERVER**

DHCP, or **Dynamic Host Configuration Protocol,** allows an administrator to configure network settings for all clients on a central server.

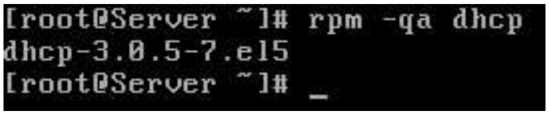
The DHCP clients request an IP address and other network settings from the **DHCP server** on the network. The **DHCP server** in turn leases the client an IP address within a given range or leases the client an IP address based on the MAC address of the client's network interface card (NIC). The information includes its IP address, along with the network's name server, gateway, and proxy addresses,including the netmask.

Nothing has to be configured manually on the local system, except to specify the **DHCP server** it should get its network configuration from. If an IP address is assigned according to the MAC address of the client's NIC, the same IP address can be leased to the client every time the client requests one. DHCP makes network administration easier and less prone to error.

Configure DHCPserver

We are using three systems one linux server one linux clients and one window clients.

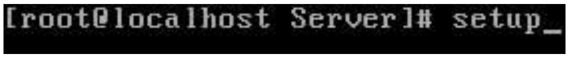
The dhcp package contains an Internet Systems Consortium (ISC) DHCP server. First, install the package as the superuser:



```
[root@Server ~]# rpm -qa dhcp
dhcp-3.0.5-7.el5
[root@Server ~]# _
```

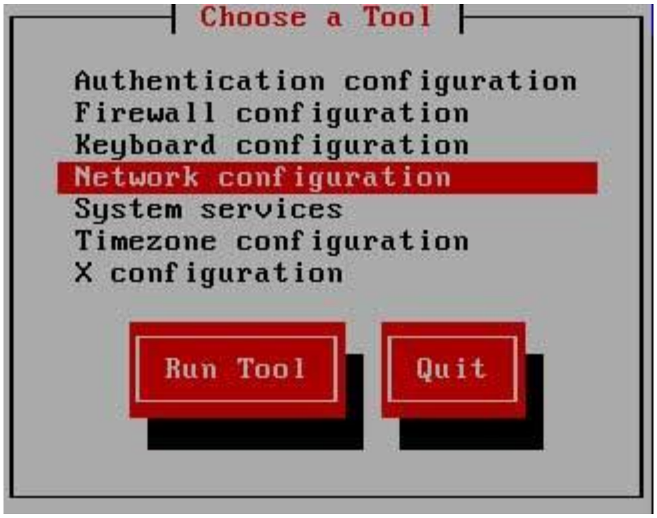
DHCP server have a static a ip address. First configure the ip address **192.168.0.254** with netmask of **255.255.255.0** on server.

Run **setup** command form root user

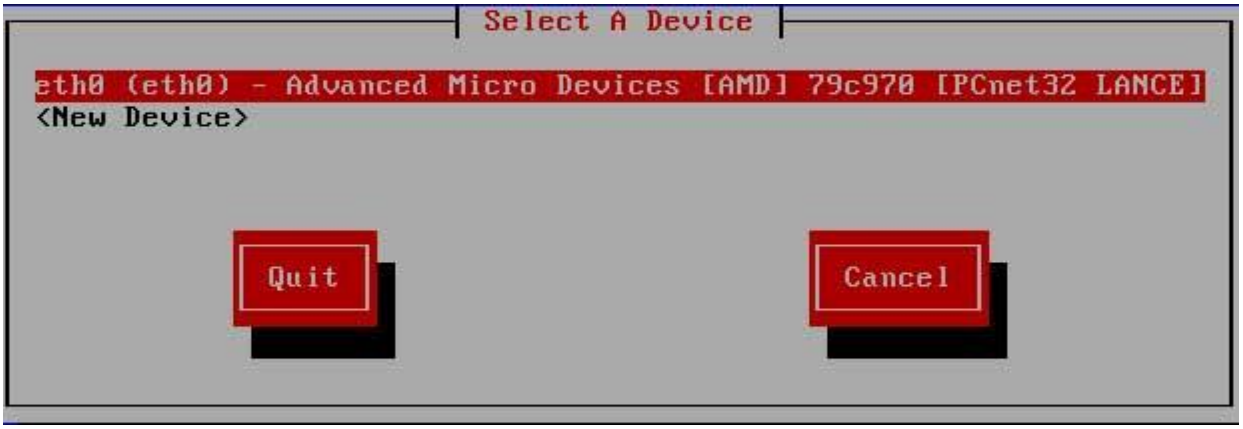


```
[root@localhost Server]# setup_
```

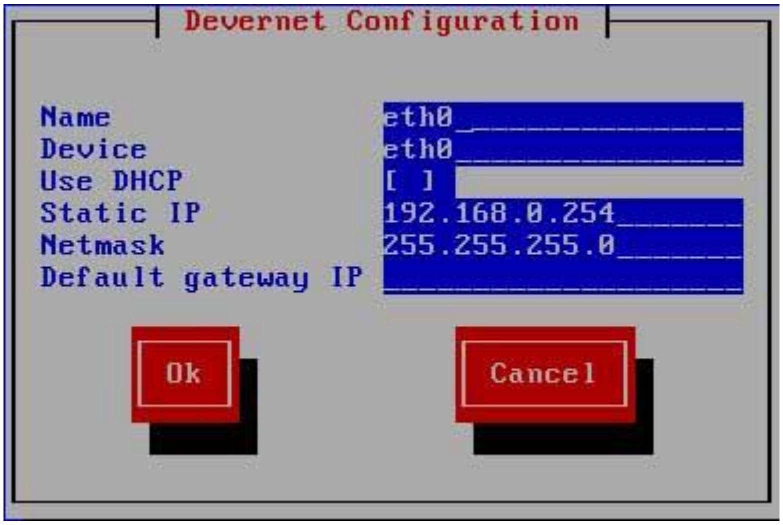
this will launch a new window select **network configuration**



now a new window will show you all available LAN card select your LAN card (if you don't see any LAN card here mean you don't have install driver)


Select A Device
eth0 (eth0) - Advanced Micro Devices [AMD] 79c970 [PCnet32 LANCE]
<New Device>
Quit
Cancel

assign IP in this box and click ok


Devernet Configuration

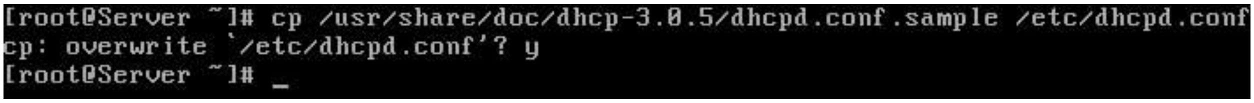
```
Name                eth0
Device              eth0
Use DHCP            [ ]
Static IP           192.168.0.254
Netmask             255.255.255.0
Default gateway IP
```

click on ok, quit and again quit to come back on root prompt.

restart the **network service** so new ip address can take place on LAN card

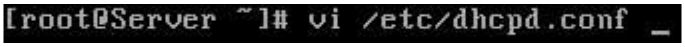
```
 #service network restart
```

main configuration file of dhcp server is**dhcpd.conf.** This file located on **/etc** directory. If this file is not present there or you have corrupted this file, then copy new file first, if ask for overwrite press **y**



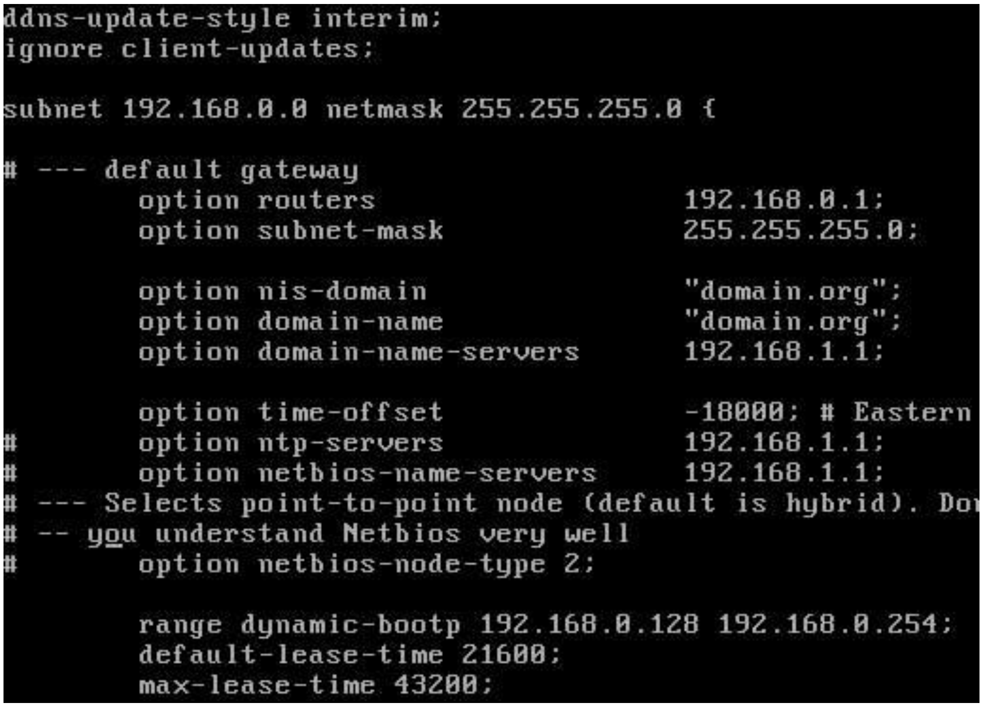
```
[root@Server ~]# cp /usr/share/doc/dhcp-3.0.5/dhcpd.conf.sample /etc/dhcpd.conf
cp: overwrite `/etc/dhcpd.conf'? y
[root@Server ~]# _
```

now open **/etc/dhcpd.conf**



```
[root@Server ~]# vi /etc/dhcpd.conf _
```

Default entry in this file look like this

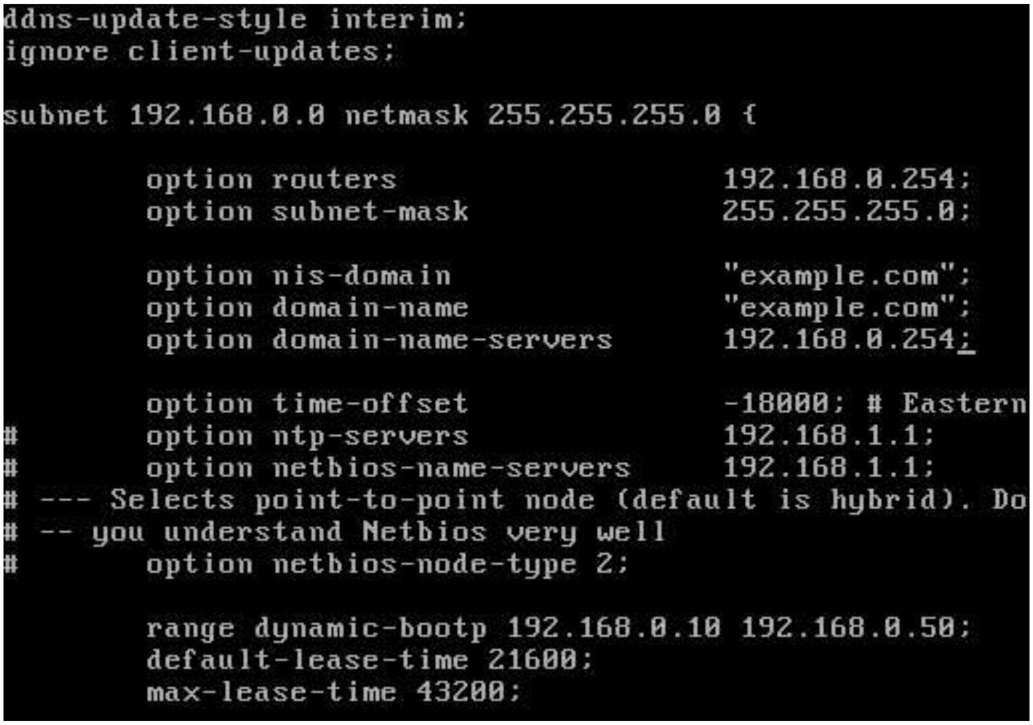


```
ddns-update-style interim;
ignore client-updates;

subnet 192.168.0.0 netmask 255.255.255.0 {

# --- default gateway
        option routers                  192.168.0.1;
        option subnet-mask              255.255.255.0;

        option nis-domain               "domain.org";
        option domain-name              "domain.org";
        option domain-name-servers      192.168.1.1;

        option time-offset              -18000; # Eastern
#       option ntp-servers              192.168.1.1;
#       option netbios-name-servers     192.168.1.1;
# --- Selects point-to-point node (default is hybrid). Do
# -- you understand Netbios very well
#       option netbios-node-type 2;

        range dynamic-bootp 192.168.0.128 192.168.0.254;
        default-lease-time 21600;
        max-lease-time 43200;
```

Make **these change** in this file to configure **dhcp server**

```
remove this line # - - - default gateway
set option routers to 192.168.0.254
set option subnet-mask to 255.255.255.0
optionnis domain to example.com
option domain-name to example.com
option domain-name-servers to 192.168.0.254
range dynamic-bootp to 192.168.0.10 192.168.0.50;
```

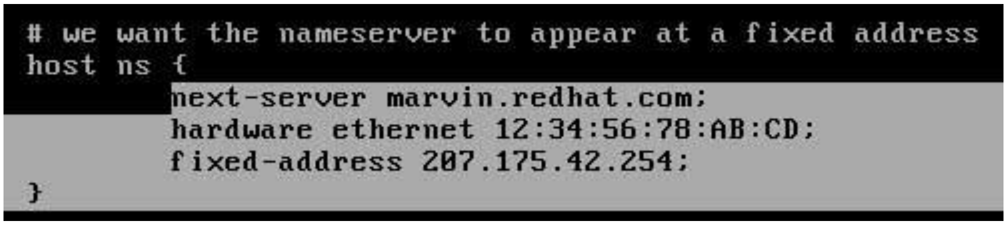
After change this file should look like this



```
ddns-update-style interim;
ignore client-updates;

subnet 192.168.0.0 netmask 255.255.255.0 {

        option routers                  192.168.0.254;
        option subnet-mask              255.255.255.0;

        option nis-domain               "example.com";
        option domain-name              "example.com";
        option domain-name-servers      192.168.0.254;

        option time-offset              -18000; # Eastern
#       option ntp-servers              192.168.1.1;
#       option netbios-name-servers     192.168.1.1;
# --- Selects point-to-point node (default is hybrid). Do
# -- you understand Netbios very well
#       option netbios-node-type 2;

        range dynamic-bootp 192.168.0.10 192.168.0.50;
        default-lease-time 21600;
        max-lease-time 43200;
```

Assign fix ip address to any host

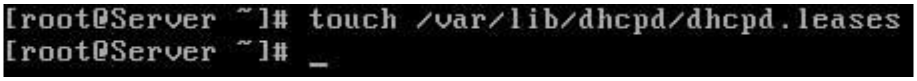
locate this paragraph and change **hardware Ethernet** to client's **mac address** and **fixed -address** to **ip address** which you want to provide that host



```
 # we want the nameserver to appear at a fixed address
 host ns {
         next-server marvin.redhat.com;
         hardware ethernet 12:34:56:78:AB:CD;
         fixed-address 207.175.42.254;
 }
```

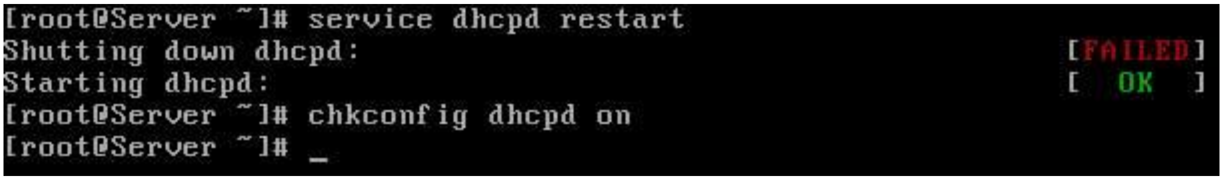
After making necessary change **save** file and exit

Now create a **blank file** use to store the allocated **ip address information**



```
[root@Server ~]# touch /var/lib/dhcpd/dhcpd.leases
[root@Server ~]# _
```

Now **restart dhcpd service** and on it with **chkconfig** commands



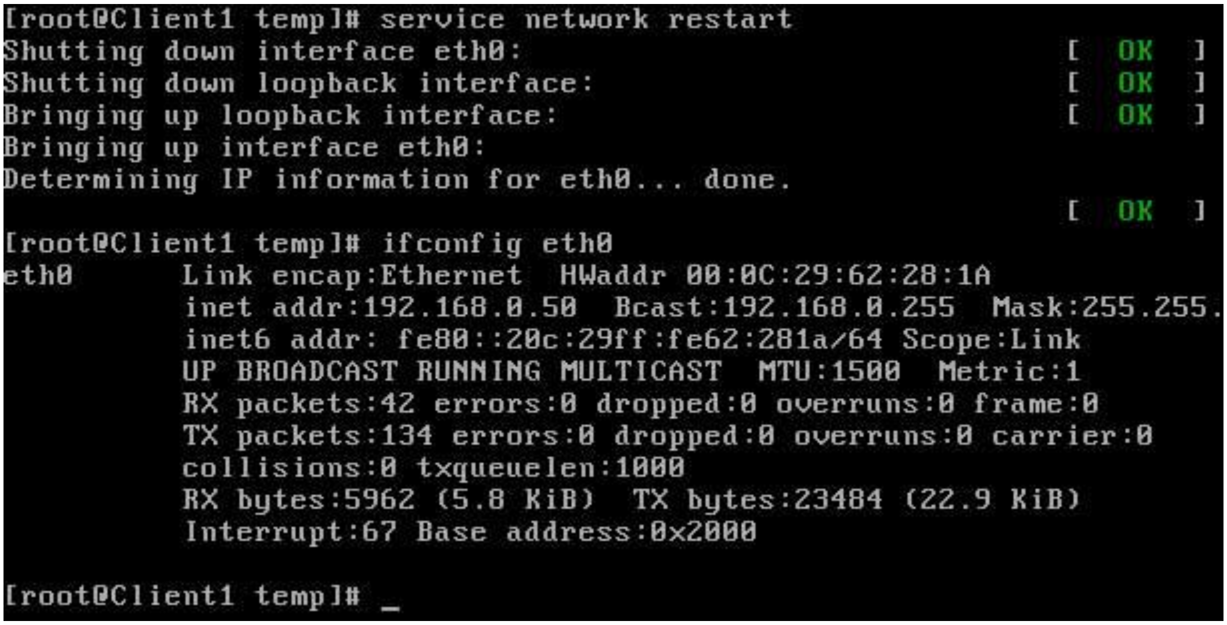
```
[root@Server ~]# service dhcpd restart
Shutting down dhcpd:                                     [FAILED]
Starting dhcpd:                                          [  OK  ]
[root@Server ~]# chkconfig dhcpd on
[root@Server ~]# _
```

Linux Client configuration

Client configuration is very easy and straightforward. All you need to do is set ip address to dynamic in the properties of lan card. In linux

```
#setup
select  network configuration from menu list
Select lan card and enter on ok
Select  USE DHCP and enter on ok
Now click on  quit
and  quit to come back on root prompt
```

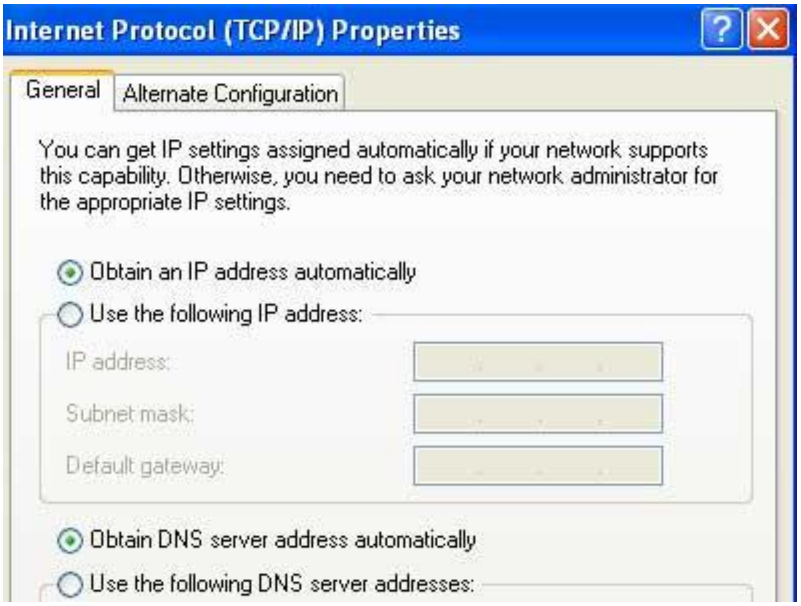
Now restart the **network service** to obtain ip from **dhcp server**



```
[root@Client1 temp]# service network restart
Shutting down interface eth0:                            [  OK  ]
Shutting down loopback interface:                        [  OK  ]
Bringing up loopback interface:                          [  OK  ]
Bringing up interface eth0:
Determining IP information for eth0... done.
                                                         [  OK  ]
[root@Client1 temp]# ifconfig eth0
eth0      Link encap:Ethernet  HWaddr 00:0C:29:62:28:1A
          inet addr:192.168.0.50  Bcast:192.168.0.255  Mask:255.255.
          inet6 addr: fe80::20c:29ff:fe62:281a/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:42 errors:0 dropped:0 overruns:0 frame:0
          TX packets:134 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:5962 (5.8 KiB)  TX bytes:23484 (22.9 KiB)
          Interrupt:67 Base address:0x2000

[root@Client1 temp]# _
```

Window Client configuration

To configure windows system as **dhcp clients** open lan card **properties** and select **tcp/ip** and click on properties and set **obtain ip address automatically**



Internet Protocol (TCP/IP) Properties

General | Alternate Configuration

You can get IP settings assigned automatically if your network supports this capability. Otherwise, you need to ask your network administrator for the appropriate IP settings.

● Obtain an IP address automatically

○ Use the following IP address:

IP address:

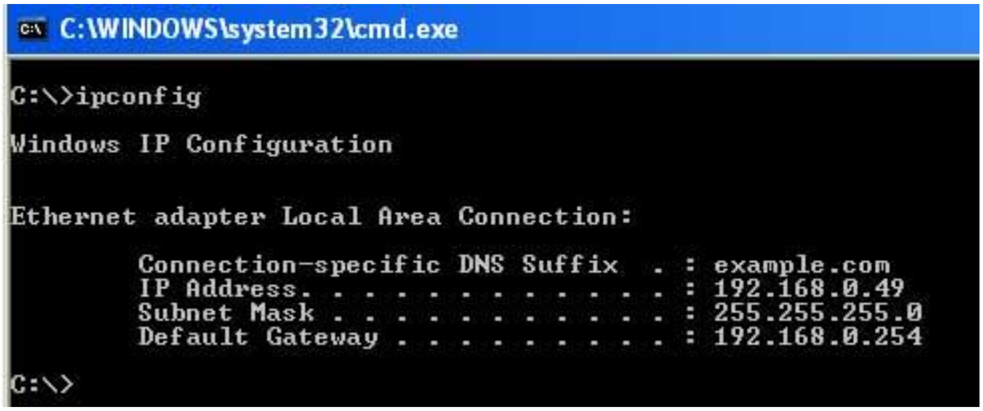
Subnet mask:

Default gateway:

● Obtain DNS server address automatically

○ Use the following DNS server addresses:

Go on **command prompt** and check new **ip address**

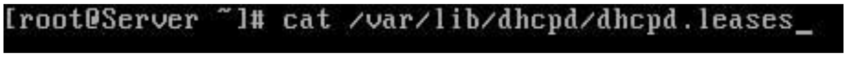


C:\WINDOWS\system32\cmd.exe

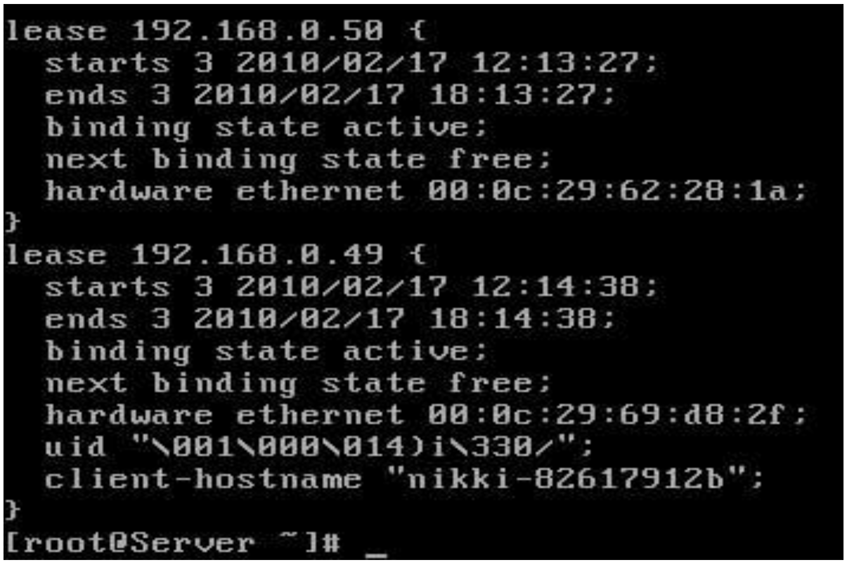
```
C:\>ipconfig

Windows IP Configuration


Ethernet adapter Local Area Connection:

        Connection-specific DNS Suffix  . : example.com
        IP Address. . . . . . . . . . . . : 192.168.0.49
        Subnet Mask . . . . . . . . . . . : 255.255.255.0
        Default Gateway . . . . . . . . . : 192.168.0.254

C:\>
```

Check lease on DHCP server

You can check **allocated address** on server.



```
[root@Server ~]# cat /var/lib/dhcpd/dhcpd.leases_
```



```
lease 192.168.0.50 {
  starts 3 2010/02/17 12:13:27;
  ends 3 2010/02/17 18:13:27;
  binding state active;
  next binding state free;
  hardware ethernet 00:0c:29:62:28:1a;
}
lease 192.168.0.49 {
  starts 3 2010/02/17 12:14:38;
  ends 3 2010/02/17 18:14:38;
  binding state active;
  next binding state free;
  hardware ethernet 00:0c:29:69:d8:2f;
  uid "\001\000\014)i\330/";
  client-hostname "nikki-82617912b";
}
[root@Server ~]# _
```

Sample /etc/dhcp.conf file

```
ddns-update-style interim;                    # Required for dhcp 3.0+ / Red Hat 8.0+
ignore client-updates;
subnet 192.168.1.0 netmask 255.255.255.0 {
range 192.168.1.128 192.168.1.254;            # Range of IP addresses to be issued to DHCP clients
option subnet-mask           255.255.255.0;   # Default subnet mask to be used by DHCP clients
option broadcast-address     192.168.1.255;   # Default broadcastaddress to be used by DHCP clients
option routers             192.168.1.1;       # Default gateway to be used by DHCP clients
option domain-name           "your-domain.org";
option domain-name-servers     40.175.42.254, 40.175.42.253;   # Default DNS to be used by DHCP
clients
optionnetbios-name-servers    192.168.1.100;  # Specify a WINS server for MS/Windows clients.
                                              # (Optional. Specify if used on your network)
# DHCP requests are not forwarded. Applies when there is more than one ethernet device and
forwarding is configured.
optionipforwarding off;
default-lease-time 21600;                     # Amount of time in seconds that a client may keep the
IP address
max-lease-time 43200;
option time-offset          -18000;           # Eastern Standard Time
optionntp-servers          192.168.1.1;       # Default NTP server to be used by DHCP clients
#      optionnetbios-name-servers    192.168.1.1;
# --- Selects point-to-point node (default is hybrid). Don't change this unless you understand Netbios
very well
#      optionnetbios-node-type 2;
# We want the nameserver "ns2" to appear at a fixed address.
# Name server with this specified MAC address will recieve this IP.
host ns2 {
```

```
next-server ns2.your-domain.com;
hardwareethernet 00:02:c3:d0:e5:83;
fixed-address 40.175.42.254;
}
# Laser printer obtains IP address via DHCP. This assures that the
# printer with this MAC address will get this IP address every time.
host laser-printer-lex1 {
hardwareethernet 08:00:2b:4c:a3:82;
fixed-address 192.168.1.120;
}
}
```

2. **CONFIGURING DNS SERVER**

A DNS server (BIND), or name server, is used to resolve an IP address to a hostname or vice versa.

You can set up four different types of DNS servers:

- A **master DNS server for your domain(s),** which stores authoritative records for your domain.
- A **slave DNS server,** which relies on a master DNS server for data.
- A **caching-only DNS server,** which stores recent requests like a proxy server. It otherwise refers to other DNS servers.
- A **forwarding-only DNS server,** which refers all requests to other DNS servers.

Before configuring BIND to create a DNS server, you must understand some basic DNS concepts.

The entire hostname with its domain such as ***server.example.com*** is called a fully qualified domain name (FQDN). The right-most part of the FQDN such as .com or .net is called the ***top level domain,*** with the remaining parts of the FQDN, which are separated by periods, being sub-domains.

These sub-domains are used to divide FQDNs into zones, with the DNS information for each zone being maintained by at least one ***authoritative name server.***

The authoritative server that contains the master zone file, which can be modified to update DNS information about the zone, is called the ***primary master server,*** or just ***master server.***

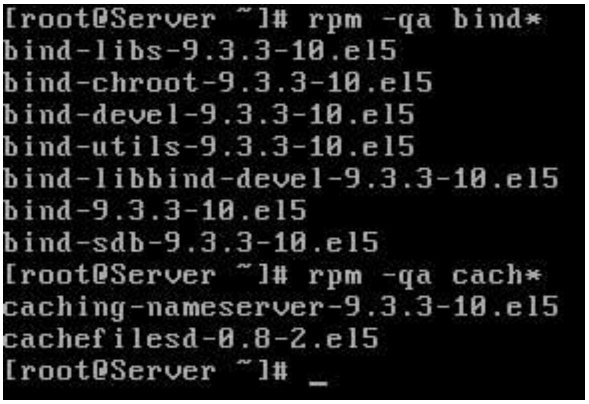
The additional name servers for the zone are called ***secondary servers*** or ***slave servers.*** Secondary servers retrieve information about the zone through a zone transfer from the master server or from another secondary server. DNS information about a zone is never modified directly on the secondary server

Configure dns server

In this example we will configure a dns server and will test from client side.

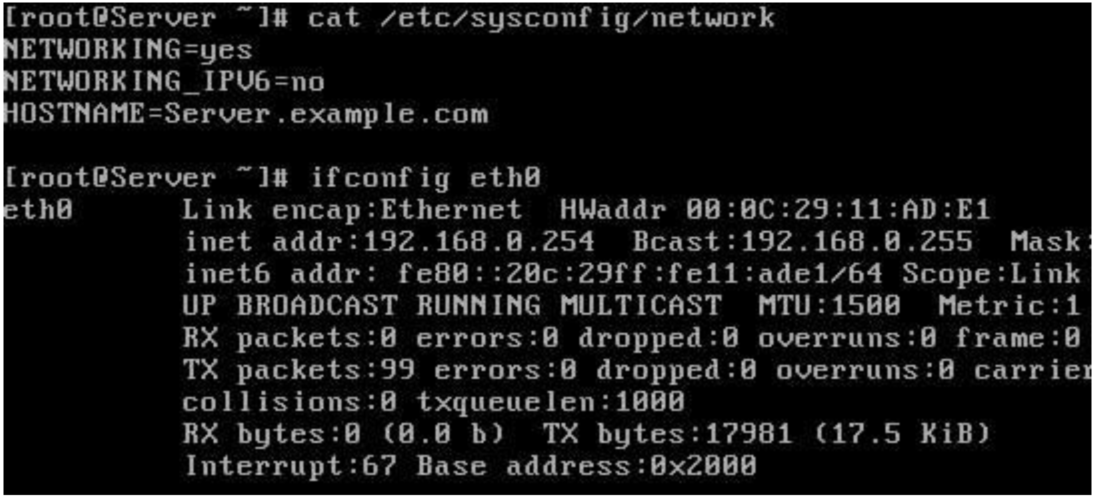
For this example we are using three systems one linux server one linux clients and one window clients.

bind and **caching-nameserver** rpm is required to configure dns. check them for install if not found install them.



```
[root@Server ~]# rpm -qa bind*
bind-libs-9.3.3-10.el5
bind-chroot-9.3.3-10.el5
bind-devel-9.3.3-10.el5
bind-utils-9.3.3-10.el5
bind-libbind-devel-9.3.3-10.el5
bind-9.3.3-10.el5
bind-sdb-9.3.3-10.el5
[root@Server ~]# rpm -qa cach*
caching-nameserver-9.3.3-10.el5
cachefilesd-0.8-2.el5
[root@Server ~]# _
```

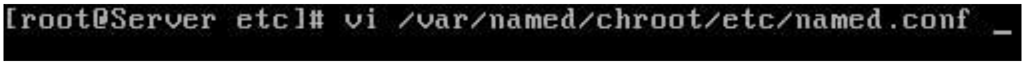
set hostname to **server.example.com** and ip address to **192.168.0.254**



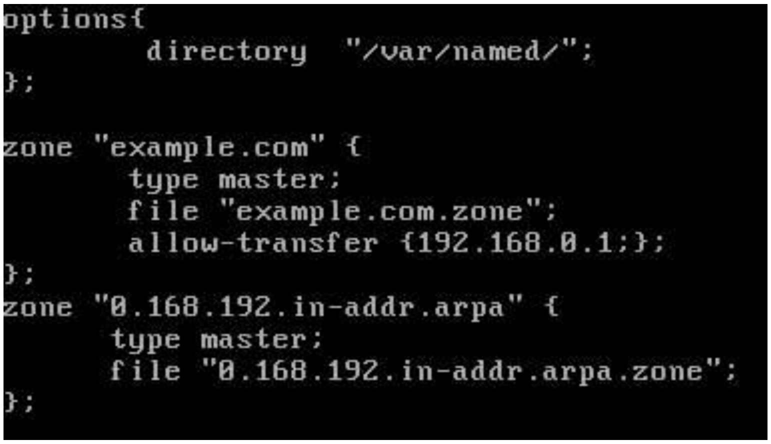
```
[root@Server ~]# cat /etc/sysconfig/network
NETWORKING=yes
NETWORKING_IPV6=no
HOSTNAME=Server.example.com

[root@Server ~]# ifconfig eth0
eth0      Link encap:Ethernet  HWaddr 00:0C:29:11:AD:E1
          inet addr:192.168.0.254  Bcast:192.168.0.255  Mask:
          inet6 addr: fe80::20c:29ff:fe11:ade1/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:99 errors:0 dropped:0 overruns:0 carrier
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 b)  TX bytes:17981 (17.5 KiB)
          Interrupt:67 Base address:0x2000
```

main configuration file for dns server is **named.conf.** By default this file is not created in **/var/named/chroot/etc/** directory. Instead of named.conf a sample file **/var/named/chroot/etc/named.caching-nameserver.conf**is created. This file is use to make a caching only name server. You can also do editing in this file after changing its name to **named.conf** to configure master dns server or you can manually create a new **named.conf** file.

In this example we are creating a new named.conf file



```
[root@Server etc]# vi /var/named/chroot/etc/named.conf _
```



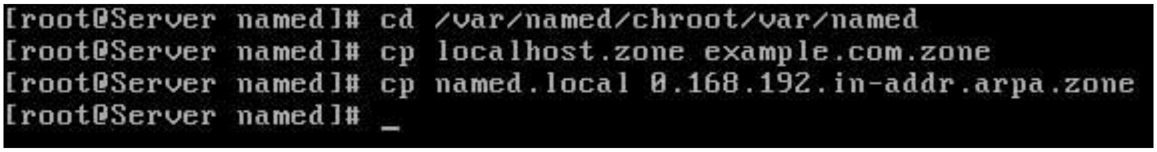
```
options{
        directory  "/var/named/";
};

zone "example.com" {
       type master;
       file "example.com.zone";
       allow-transfer {192.168.0.1;};
};
zone "0.168.192.in-addr.arpa" {
      type master;
      file "0.168.192.in-addr.arpa.zone";
};
```

save this file with **:wq** and exit

Configure zone file

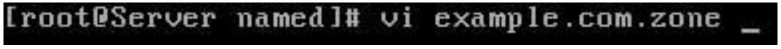
We have defined two zone files **example.com.zone** for forward zone and **0.168.192.in-addr.arpa** for reverse zone. These files will be store in **/var/named/chroot/var/named/** location. We will use two sample files for creating these files.

Change directory to **/var/named/chroot/var/named** and copy the sample files to name which we have set in named.conf



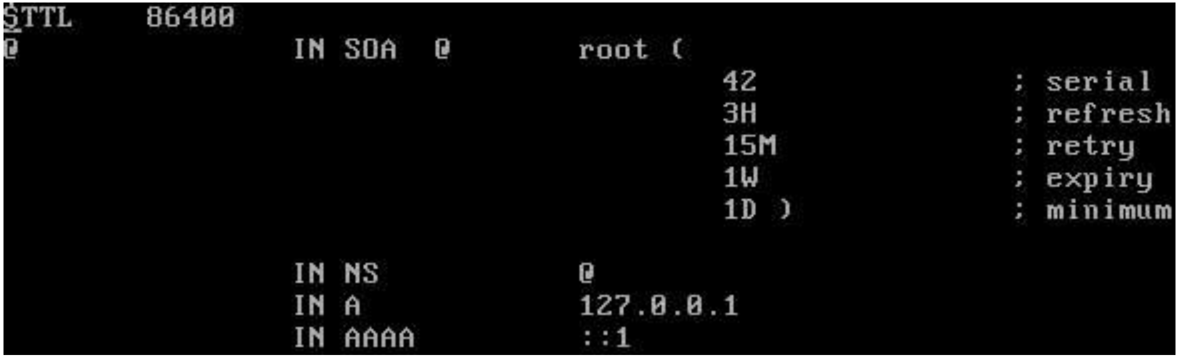
```
[root@Server named]# cd /var/named/chroot/var/named
[root@Server named]# cp localhost.zone example.com.zone
[root@Server named]# cp named.local 0.168.192.in-addr.arpa.zone
[root@Server named]# _
```

Now open forward zone file **example.com.zone**



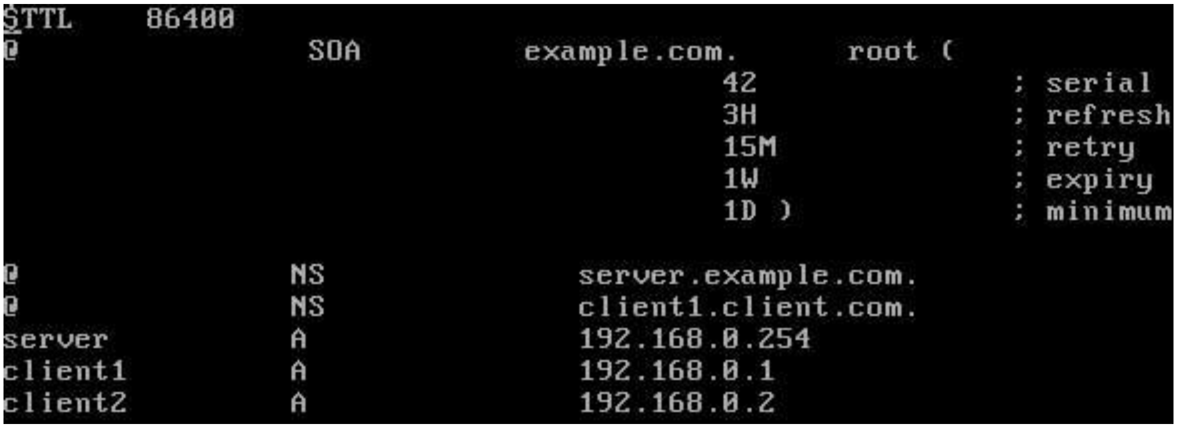
```
[root@Server named]# vi example.com.zone _
```

By default this file will look like this



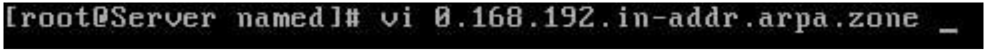
```
$TTL    86400
@               IN SOA  @       root (
                                        42              ; serial
                                        3H              ; refresh
                                        15M             ; retry
                                        1W              ; expiry
                                        1D )            ; minimum

                IN NS           @
                IN A            127.0.0.1
                IN AAAA         ::1
```

Change this file exactly as shown in image below



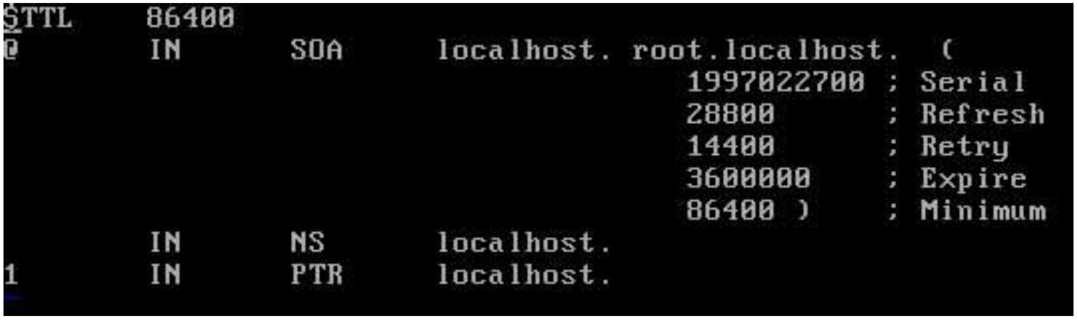
```
$TTL    86400
@                SOA         example.com.      root (
                                       42                ; serial
                                       3H                ; refresh
                                       15M               ; retry
                                       1W                ; expiry
                                       1D )              ; minimum

@               NS              server.example.com.
@               NS              client1.client.com.
server          A               192.168.0.254
client1         A               192.168.0.1
client2         A               192.168.0.2
```

Now open reverse lookup zone file **0.168.192.in-addr.arpa**



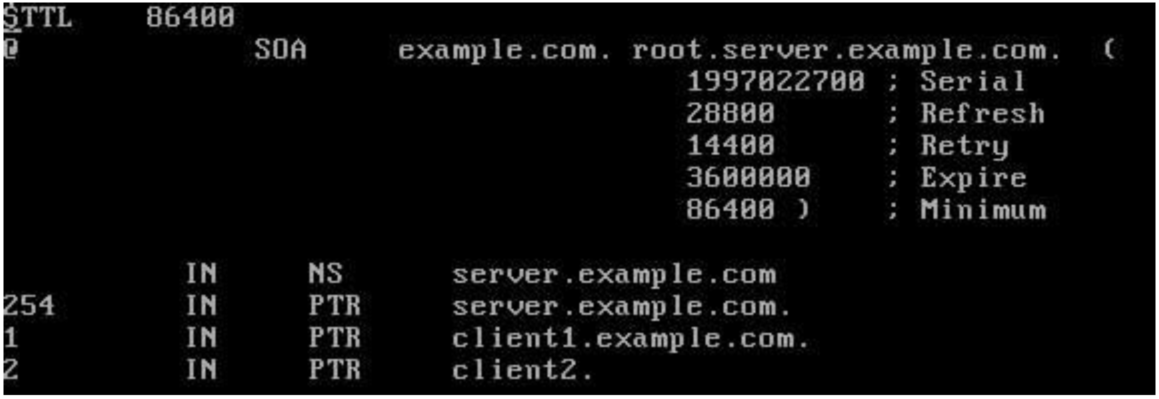
```
[root@Server named]# vi 0.168.192.in-addr.arpa.zone _
```

By default this file will look like this



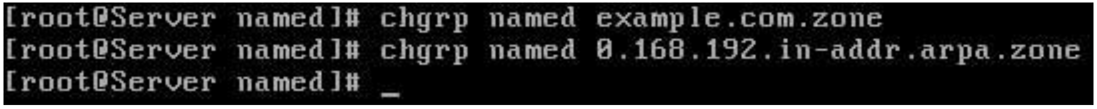
```
$TTL    86400
@       IN      SOA     localhost. root.localhost.  (
                                    1997022700 ; Serial
                                    28800      ; Refresh
                                    14400      ; Retry
                                    3600000    ; Expire
                                    86400 )    ; Minimum
        IN      NS      localhost.
1       IN      PTR     localhost.
```

Change this file exactly as shown in image below



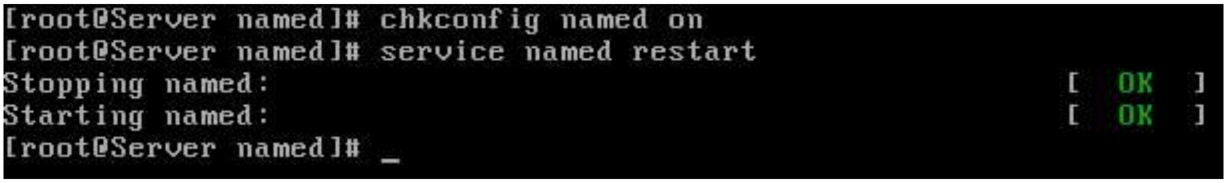
```
$TTL    86400
@             SOA     example.com. root.server.example.com.  (
                                    1997022700 ; Serial
                                    28800      ; Refresh
                                    14400      ; Retry
                                    3600000    ; Expire
                                    86400 )    ; Minimum

          IN      NS      server.example.com
254       IN      PTR     server.example.com.
1         IN      PTR     client1.example.com.
2         IN      PTR     client2.
```

Now changed the ownership of these zone files to **named** group



```
[root@Server named]# chgrp named example.com.zone
[root@Server named]# chgrp named 0.168.192.in-addr.arpa.zone
[root@Server named]# _
```

Now start the named service



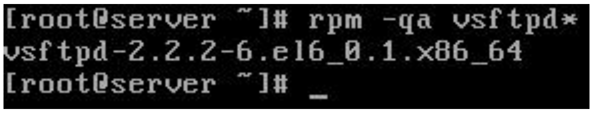
```
[root@Server named]# chkconfig named on
[root@Server named]# service named restart
Stopping named:                                            [  OK  ]
Starting named:                                            [  OK  ]
[root@Server named]# _
```

3. CONFIGURING FTP SERVER

FTP stand for **F**ile **T**ransfer **P**rotocol. As the name suggest this network protocol allows you to transfer files or directories from one host to another over the network whether it is your LAN or Internet.

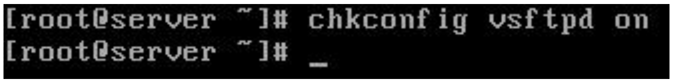
Configure FTP Server

vsftpd package is required for FTP Server. Check whether package is installed or not. If package is missing install it first.



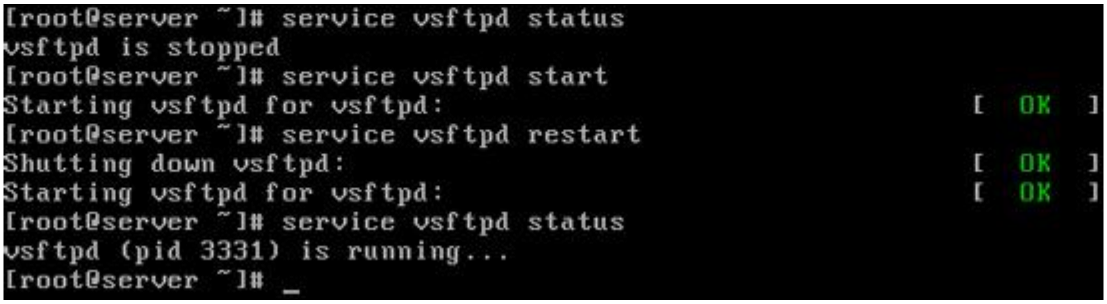
```
[root@server ~]# rpm -qa vsftpd*
vsftpd-2.2.2-6.el6_0.1.x86_64
[root@server ~]# _
```

Configure **vsftpd** service to start at boot



```
[root@server ~]# chkconfig vsftpd on
[root@server ~]# _
```

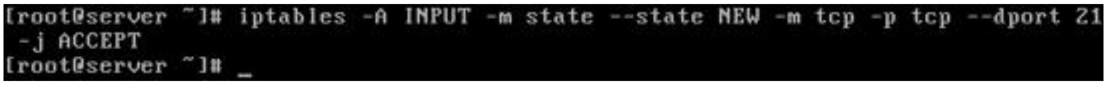
Current status of **vsftpd** service must be running. Start if it is stopped. Restart **vsftpd** service whenever you made any change in configuration file.



```
[root@server ~]# service vsftpd status
vsftpd is stopped
[root@server ~]# service vsftpd start
Starting vsftpd for vsftpd:                                [  OK  ]
[root@server ~]# service vsftpd restart
Shutting down vsftpd:                                      [  OK  ]
Starting vsftpd for vsftpd:                                [  OK  ]
[root@server ~]# service vsftpd status
vsftpd (pid 3331) is running...
[root@server ~]# _
```

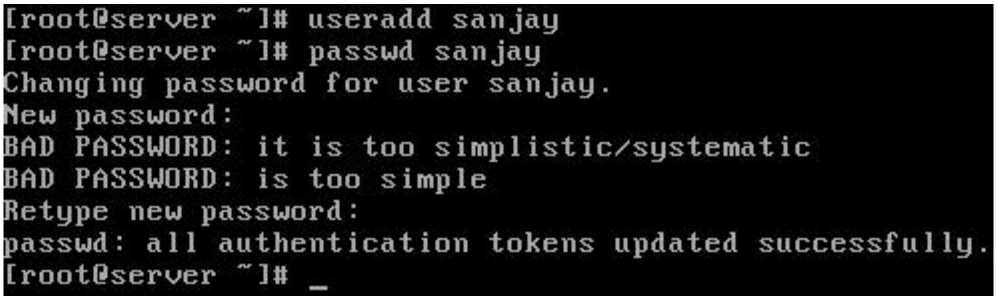
FTP Server is by default configured to listen on port 21. Port 21 must be opened if you have configured firewall. The configuration of a firewall for an FTP server is a relatively simple process.

```
#iptables -A INPUT -m state --state NEW -m tcp -p tcp --dport 21 -j ACCEPT
```



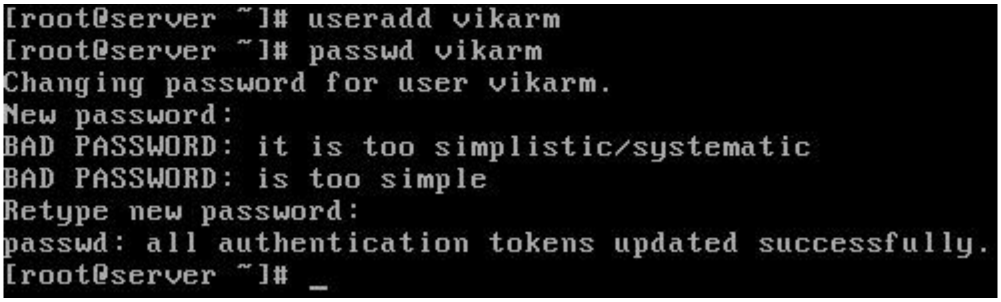
```
[root@server ~]# iptables -A INPUT -m state --state NEW -m tcp -p tcp --dport 21
 -j ACCEPT
[root@server ~]# _
```

Create 2 normal user accounts for testing. Create a normal user



```
[root@server ~]# useradd sanjay
[root@server ~]# passwd sanjay
Changing password for user sanjay.
New password:
BAD PASSWORD: it is too simplistic/systematic
BAD PASSWORD: is too simple
Retype new password:
passwd: all authentication tokens updated successfully.
[root@server ~]# _
```

Create another normal user



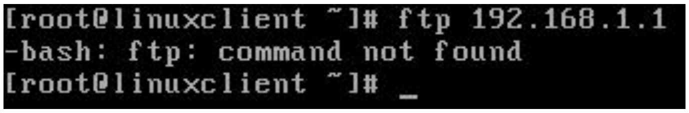
```
[root@server ~]# useradd vikarm
[root@server ~]# passwd vikarm
Changing password for user vikarm.
New password:
BAD PASSWORD: it is too simplistic/systematic
BAD PASSWORD: is too simple
Retype new password:
passwd: all authentication tokens updated successfully.
[root@server ~]# _
```

That's all configure we need on server right now.

Configure FTP client

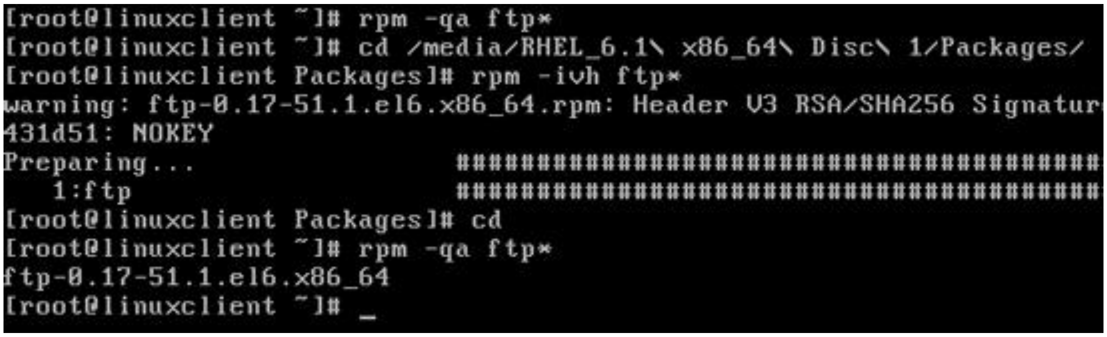
You will not be able to run ftp command. By default you will get following error

```
-bash: ftp: command not found error
```



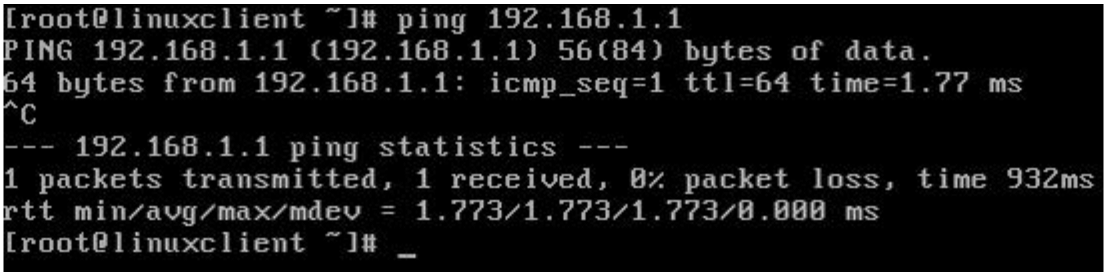
```
[root@linuxclient ~]# ftp 192.168.1.1
-bash: ftp: command not found
[root@linuxclient ~]# _
```

To run **ftp** command **ftp** package is required. Install it if it is not installed.



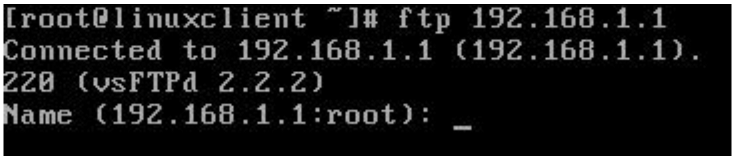
```
[root@linuxclient ~]# rpm -qa ftp*
[root@linuxclient ~]# cd /media/RHEL_6.1\ x86_64\ Disc\ 1/Packages/
[root@linuxclient Packages]# rpm -ivh ftp*
warning: ftp-0.17-51.1.el6.x86_64.rpm: Header V3 RSA/SHA256 Signatur
431d51: NOKEY
Preparing...                ##########################################
   1:ftp                    ##########################################
[root@linuxclient Packages]# cd
[root@linuxclient ~]# rpm -qa ftp*
ftp-0.17-51.1.el6.x86_64
[root@linuxclient ~]# _
```

Check connectivity with FTP Server.



```
[root@linuxclient ~]# 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_seq=1 ttl=64 time=1.77 ms
^C
--- 192.168.1.1 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 932ms
rtt min/avg/max/mdev = 1.773/1.773/1.773/0.000 ms
[root@linuxclient ~]# _
```

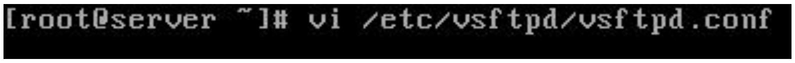
Now try again to run **ftp** command



```
[root@linuxclient ~]# ftp 192.168.1.1
Connected to 192.168.1.1 (192.168.1.1).
220 (vsFTPd 2.2.2)
Name (192.168.1.1:root): _
```

We have successfully connected with FTP server.

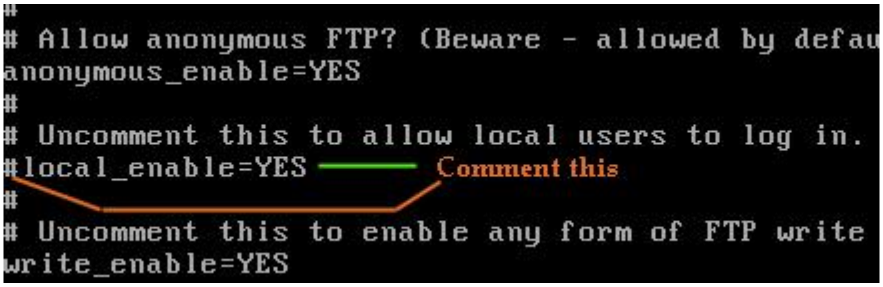
Go on Server system and open main ftp configuration file **/etc/vsftpd/vsftpd.conf**



```
[root@server ~]# vi /etc/vsftpd/vsftpd.conf
```

vsftpd.conf is the main configuration file of FTP server and it contains lot of directives. Configuration of an anonymous-only download is relatively simple. Default configuration of **vsftpd.conf** already supports anonymous-only download. But it also supports access from local users. All you need to do is disable the directive which allows locally configured users to login with their accounts.

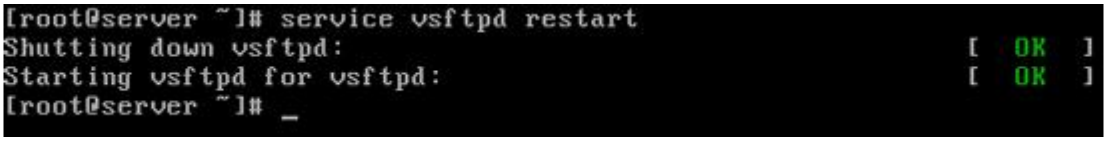
Comment following directives and save the file



```
#
# Allow anonymous FTP? (Beware - allowed by defau
anonymous_enable=YES
#
# Uncomment this to allow local users to log in.
#local_enable=YES
#
# Uncomment this to enable any form of FTP write
write_enable=YES
```

Comment this

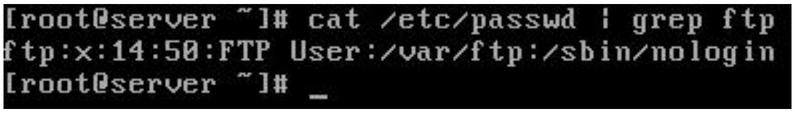
Restart the **vsftpd** service



```
[root@server ~]# service vsftpd restart
Shutting down vsftpd:                                      [  OK  ]
Starting vsftpd for vsftpd:                                [  OK  ]
[root@server ~]# _
```

When a user connects on the FTP server with anonymous **username**, actually that user connects on the server as a user named **ftp**. RHEL6 automatically create this account with following setting.

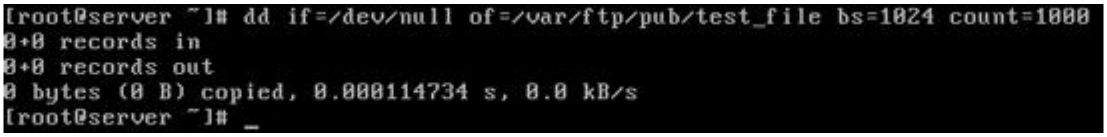
```
ftp:x:14:50:FTP User:/var/ftp:/sbin/nologin
```



```
[root@server ~]# cat /etc/passwd | grep ftp
ftp:x:14:50:FTP User:/var/ftp:/sbin/nologin
[root@server ~]# _
```

With these setting users are not allowed to login as the user named **ftp**. So they need to use **anonymous** as user name. So whenever an **anonymous** user logged in, he is taken to ftp user's home directory **/var/ftp**. So if you want to change the default directory associated with anonymous logins, change the home directory associated with the local user named **ftp**. Create a file on the root of the ftp directory **/var/ftp/pub**. This file will be downloaded by anonymous user.

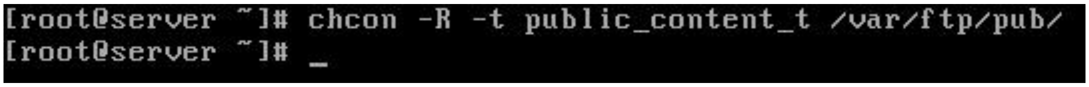
```
# dd if=/dev/null of=/var/ftp/pub/file bs=1024 count=1000
```



```
[root@server ~]# dd if=/dev/null of=/var/ftp/pub/test_file bs=1024 count=1000
0+0 records in
0+0 records out
0 bytes (0 B) copied, 0.000114734 s, 0.0 kB/s
[root@server ~]# _
```

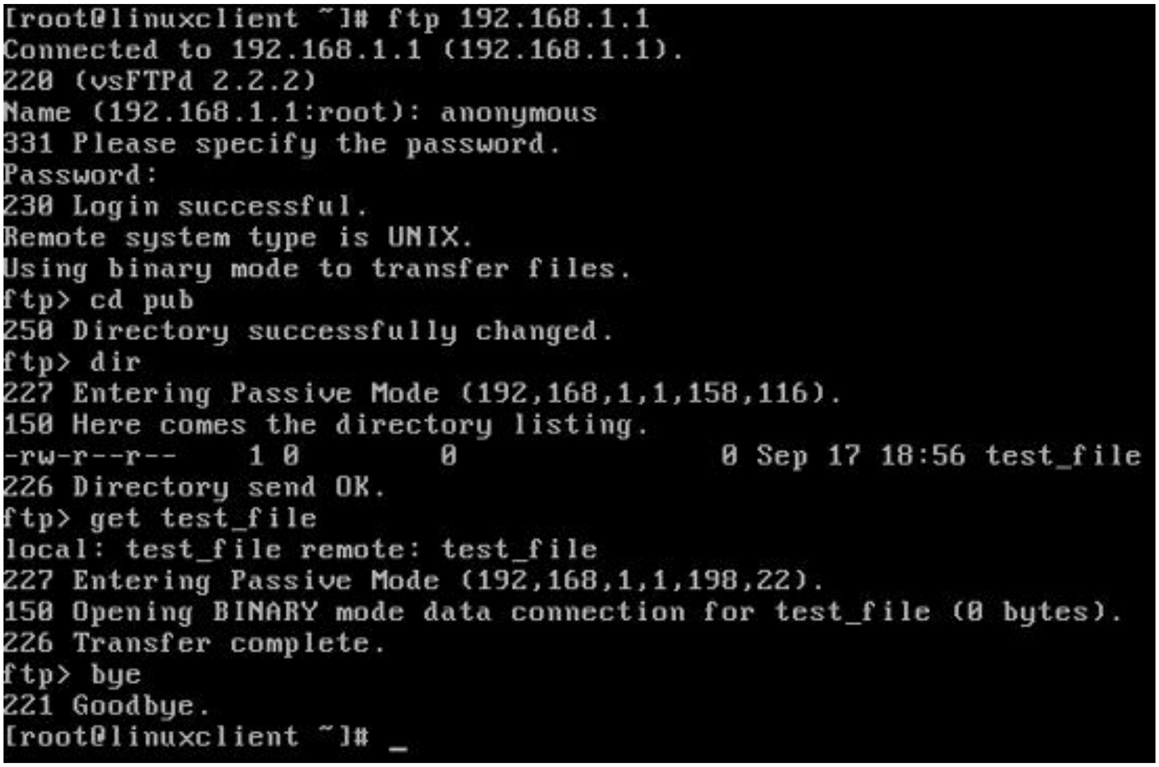
If you are running Linux without SELinux that's all setting which we need for this exercise. SELinux is listed in RHCE6 exam objective. So if you have configured SELinux, also configure following boolean option.

```
# chcon -R -t public_content_t /var/ftp/pub/
```



```
[root@server ~]# chcon -R -t public_content_t /var/ftp/pub/
[root@server ~]# _
```

Go on **linuxclient** system and login to the FTP server as anonymous user and download **test_file**

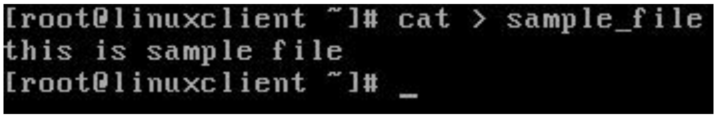


```
[root@linuxclient ~]# ftp 192.168.1.1
Connected to 192.168.1.1 (192.168.1.1).
220 (vsFTPd 2.2.2)
Name (192.168.1.1:root): anonymous
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> cd pub
250 Directory successfully changed.
ftp> dir
227 Entering Passive Mode (192,168,1,1,158,116).
150 Here comes the directory listing.
-rw-r--r--    1 0        0               0 Sep 17 18:56 test_file
226 Directory send OK.
ftp> get test_file
local: test_file remote: test_file
227 Entering Passive Mode (192,168,1,1,198,22).
150 Opening BINARY mode data connection for test_file (0 bytes).
226 Transfer complete.
ftp> bye
221 Goodbye.
[root@linuxclient ~]# _
```

Most commonly commands used on ftp prompt are

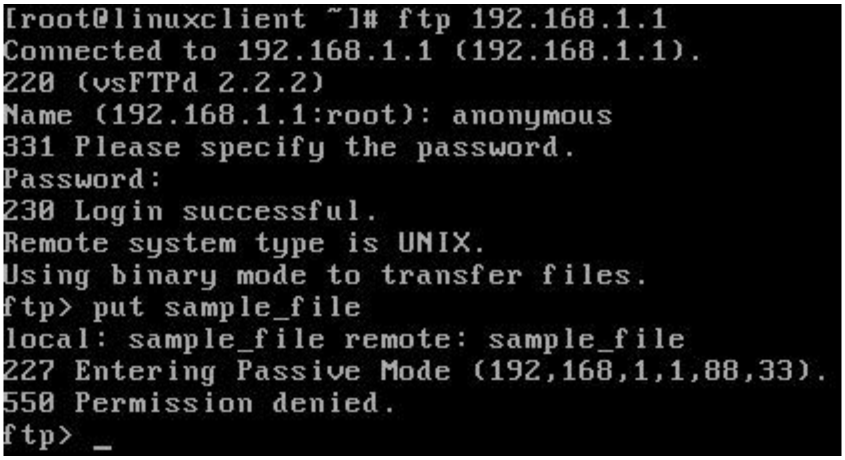
```
put  To upload files on server
get  To download files from server
mput To upload all files
mget To download all files
?    To see all available command on ftp prompts
cd   To change remote directory
lcd  To change local directory.
```

Create a sample file



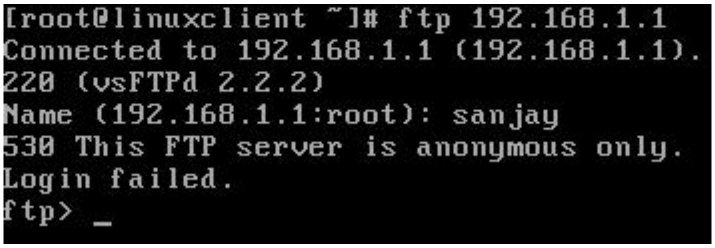
```
[root@linuxclient ~]# cat > sample_file
this is sample file
[root@linuxclient ~]# _
```

Login from **anonymous** again and try to upload



```
[root@linuxclient ~]# ftp 192.168.1.1
Connected to 192.168.1.1 (192.168.1.1).
220 (vsFTPd 2.2.2)
Name (192.168.1.1:root): anonymous
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> put sample_file
local: sample_file remote: sample_file
227 Entering Passive Mode (192,168,1,1,88,33).
550 Permission denied.
ftp> _
```

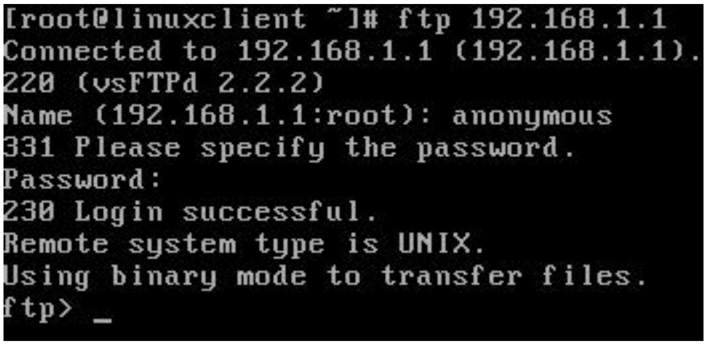
Try to login form normal user



```
[root@linuxclient ~]# ftp 192.168.1.1
Connected to 192.168.1.1 (192.168.1.1).
220 (vsFTPd 2.2.2)
Name (192.168.1.1:root): sanjay
530 This FTP server is anonymous only.
Login failed.
ftp> _
```

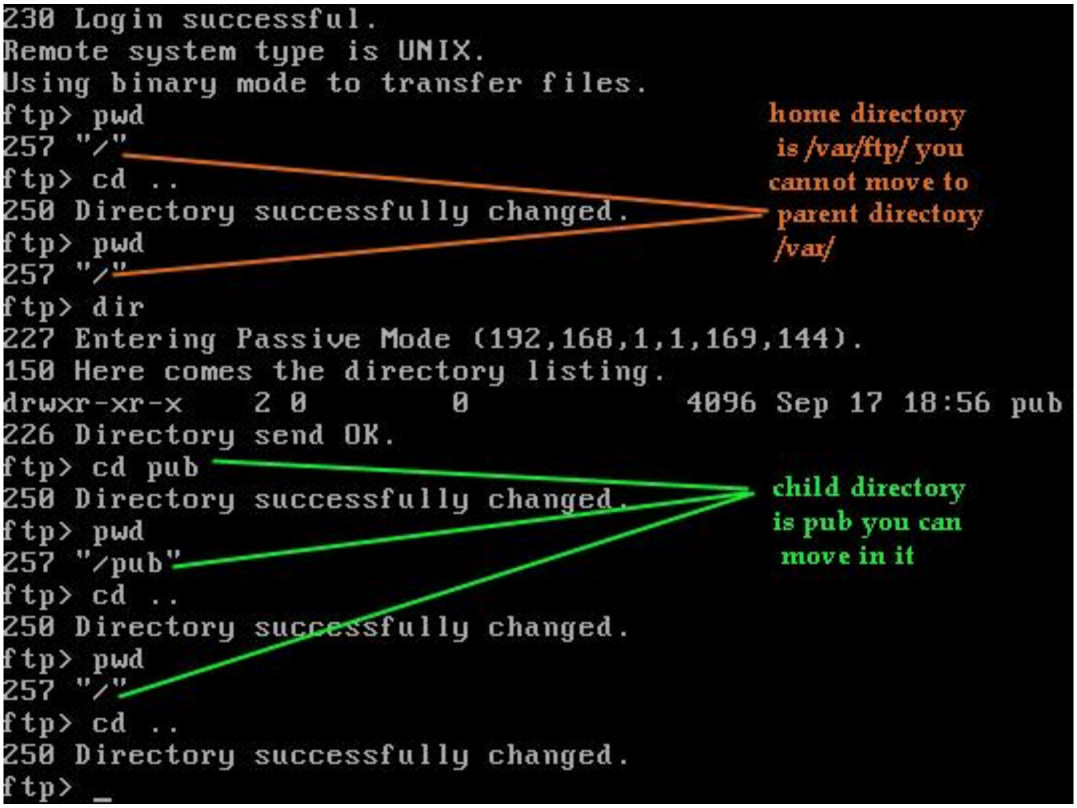
Restrict anonymous user to ftp directory.

To test this login form anonymous user again



```
[root@linuxclient ~]# ftp 192.168.1.1
Connected to 192.168.1.1 (192.168.1.1).
220 (vsFTPd 2.2.2)
Name (192.168.1.1:root): anonymous
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> _
```

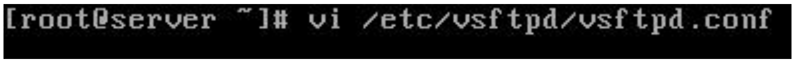
Try to change parent directory



```
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> pwd
257 "/"
ftp> cd ..
250 Directory successfully changed.
ftp> pwd
257 "/"
ftp> dir
227 Entering Passive Mode (192,168,1,1,169,144).
150 Here comes the directory listing.
drwxr-xr-x    2 0        0            4096 Sep 17 18:56 pub
226 Directory send OK.
ftp> cd pub
250 Directory successfully changed.
ftp> pwd
257 "/pub"
ftp> cd ..
250 Directory successfully changed.
ftp> pwd
257 "/"
ftp> cd ..
250 Directory successfully changed.
ftp> _
```

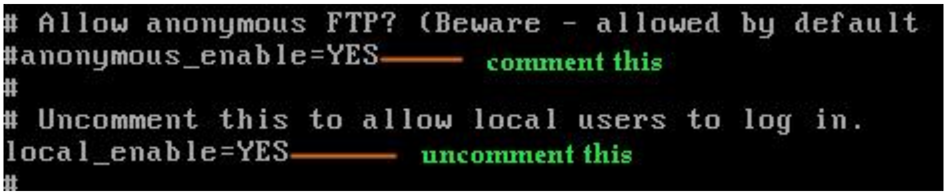
FTP non-anonymous server

In this exercise we will configure FTP server that allow local users logins to their home directories. Download/upload must be allowed for these users. Go on server system and open **/etc/vsftpd/vsftpd.conf** file



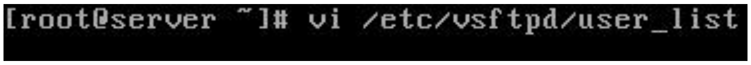
```
[root@server ~]# vi /etc/vsftpd/vsftpd.conf
```

Comment **anonymous_login=YES**, uncomment **local_enable** and save the file



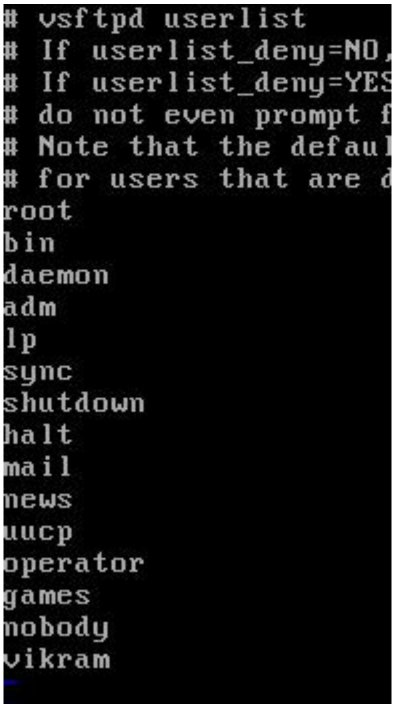
```
# Allow anonymous FTP? (Beware - allowed by default
#anonymous_enable=YES
#
# Uncomment this to allow local users to log in.
local_enable=YES
#
```

open**/etc/vsftpd/user_list** file



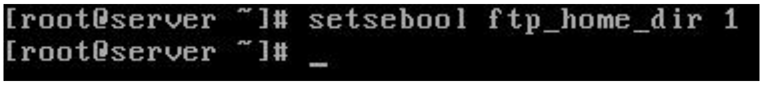
```
[root@server ~]# vi /etc/vsftpd/user_list
```

Users listed on **/etc/vsftpd/user_list** are not allowed to login on FTP server. Add user **vikarm** in it. This file also have an entry for root user that why root user is denied from FTP login. If you want to enable root user for ftp session just remove its entry from this file [Enable root for FTP session is not recommended in any circumstances, change at your own risk].



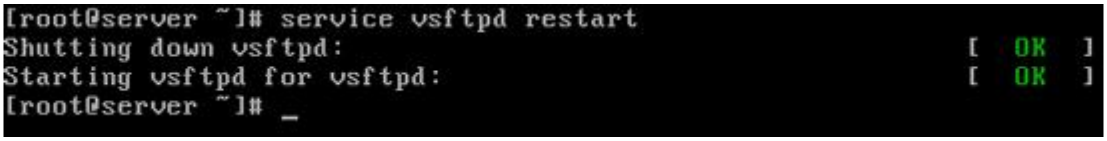
```
# vsftpd userlist
# If userlist_deny=NO,
# If userlist_deny=YES
# do not even prompt f
# Note that the defaul
# for users that are d
root
bin
daemon
adm
lp
sync
shutdown
halt
mail
news
uucp
operator
games
nobody
vikram
_
```

Configure SELinux to allow upload/download in user's home directory



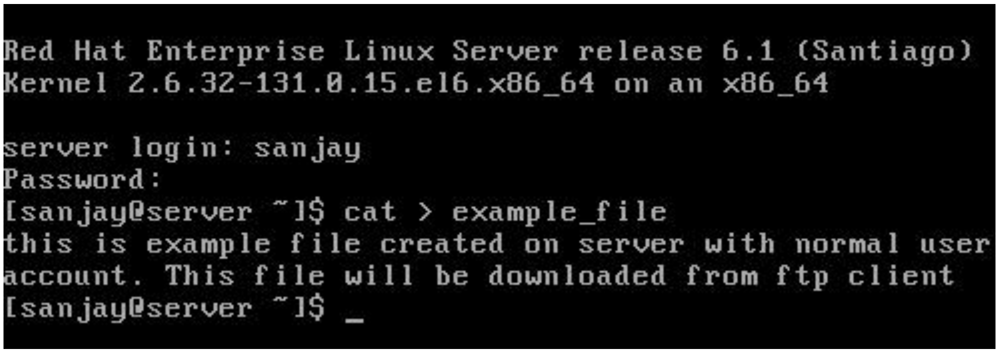
```
[root@server ~]# setsebool ftp_home_dir 1
[root@server ~]# _
```

Restart the **vsftpd** service



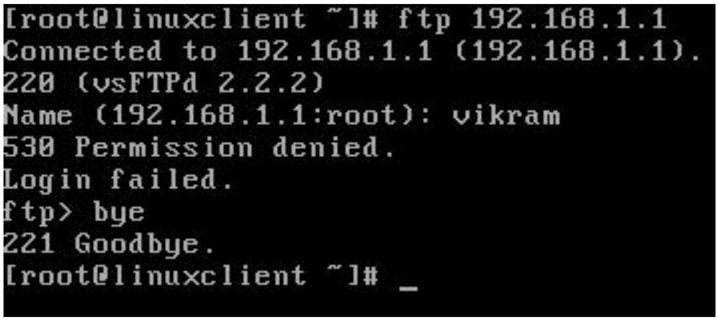
```
[root@server ~]# service vsftpd restart
Shutting down vsftpd:                                      [  OK  ]
Starting vsftpd for vsftpd:                                [  OK  ]
[root@server ~]# _
```

Login from normal user **sanjay** and create a**example_file**



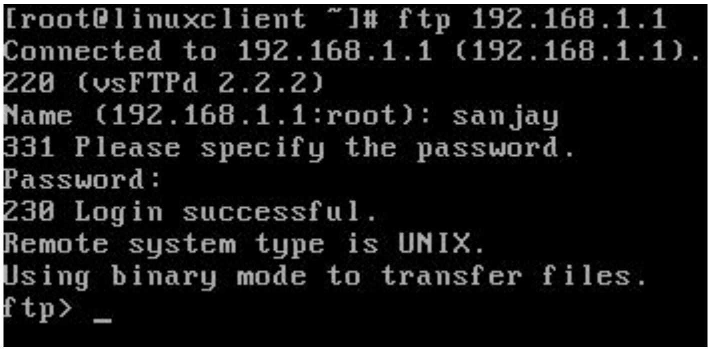
```
Red Hat Enterprise Linux Server release 6.1 (Santiago)
Kernel 2.6.32-131.0.15.el6.x86_64 on an x86_64

server login: sanjay
Password:
[sanjay@server ~]$ cat > example_file
this is example file created on server with normal user
account. This file will be downloaded from ftp client
[sanjay@server ~]$ _
```

Come back on **linuxclient** system and try to login from user **vikram**



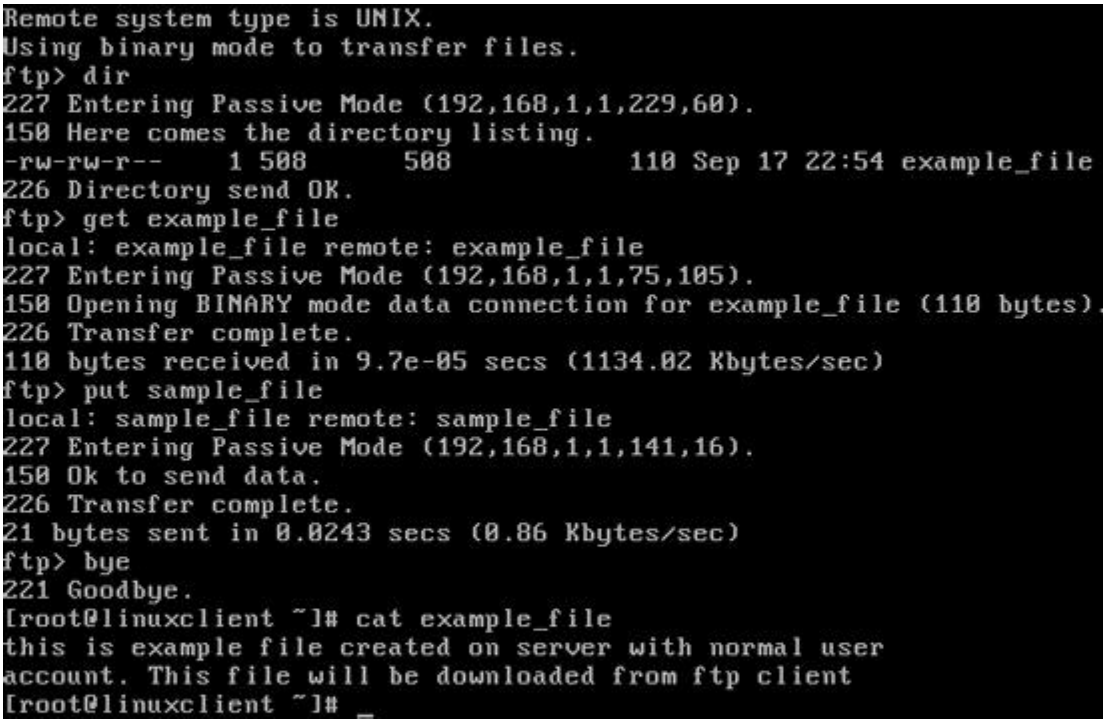
```
[root@linuxclient ~]# ftp 192.168.1.1
Connected to 192.168.1.1 (192.168.1.1).
220 (vsFTPd 2.2.2)
Name (192.168.1.1:root): vikram
530 Permission denied.
Login failed.
ftp> bye
221 Goodbye.
[root@linuxclient ~]# _
```

Now try to login from user **sanjay**



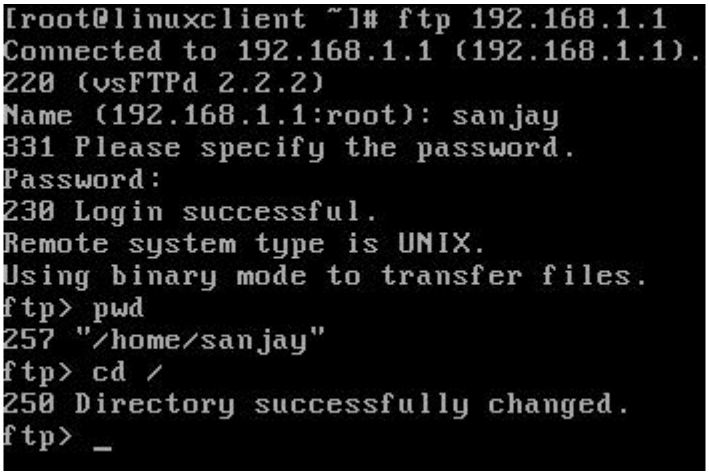
```
[root@linuxclient ~]# ftp 192.168.1.1
Connected to 192.168.1.1 (192.168.1.1).
220 (vsFTPd 2.2.2)
Name (192.168.1.1:root): sanjay
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> _
```

upload/download file



```
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> dir
227 Entering Passive Mode (192,168,1,1,229,60).
150 Here comes the directory listing.
-rw-rw-r--    1 508      508           110 Sep 17 22:54 example_file
226 Directory send OK.
ftp> get example_file
local: example_file remote: example_file
227 Entering Passive Mode (192,168,1,1,75,105).
150 Opening BINARY mode data connection for example_file (110 bytes).
226 Transfer complete.
110 bytes received in 9.7e-05 secs (1134.02 Kbytes/sec)
ftp> put sample_file
local: sample_file remote: sample_file
227 Entering Passive Mode (192,168,1,1,141,16).
150 Ok to send data.
226 Transfer complete.
21 bytes sent in 0.0243 secs (0.86 Kbytes/sec)
ftp> bye
221 Goodbye.
[root@linuxclient ~]# cat example_file
this is example file created on server with normal user
account. This file will be downloaded from ftp client
[root@linuxclient ~]# _
```

Login again from normal user and try to change parent directory

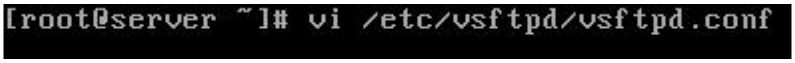


```
[root@linuxclient ~]# ftp 192.168.1.1
Connected to 192.168.1.1 (192.168.1.1).
220 (vsFTPd 2.2.2)
Name (192.168.1.1:root): sanjay
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> pwd
257 "/home/sanjay"
ftp> cd /
250 Directory successfully changed.
ftp> _
```

It allows you to navigate the / directory which serious security issue.

Configure FTP to chroot local users in their home directory

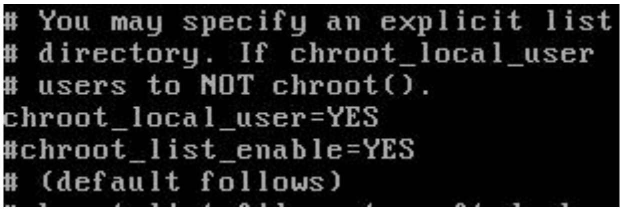
Go on server and open **/etc/vsftpd/vsftpd.conf** file



```
[root@server ~]# vi /etc/vsftpd/vsftpd.conf
```

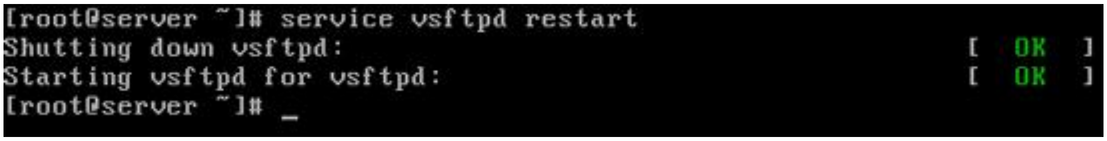
Uncomment following directive and save the file

```
chroot_local_user=YES
```



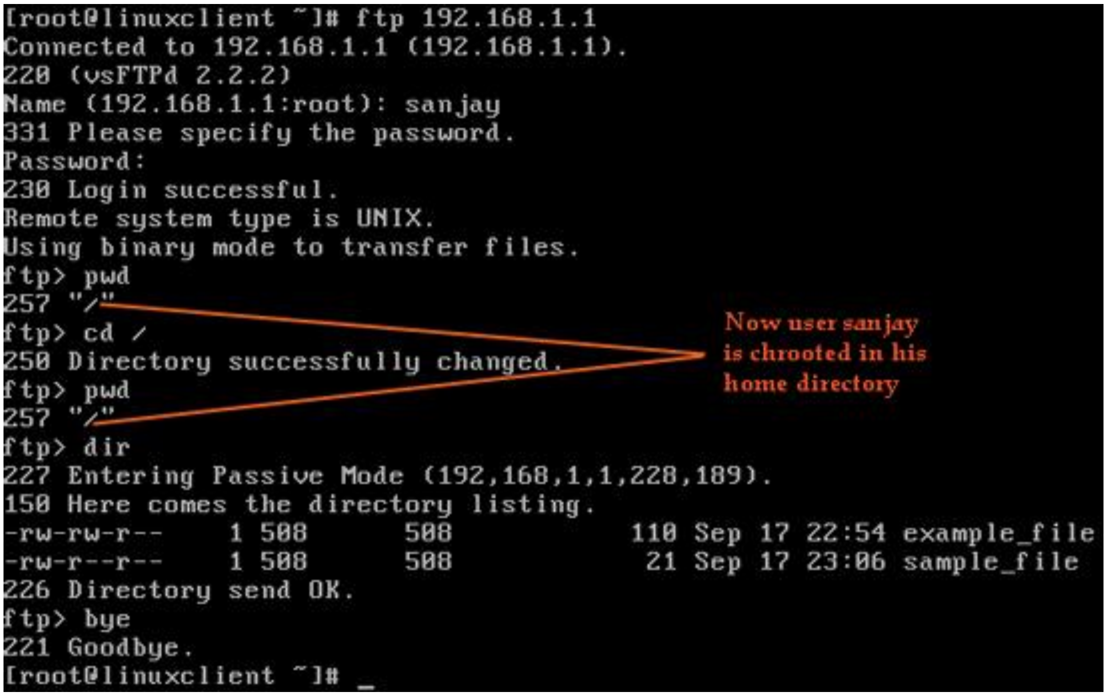
```
# You may specify an explicit list
# directory. If chroot_local_user
# users to NOT chroot().
chroot_local_user=YES
#chroot_list_enable=YES
# (default follows)
```

Restart the **vsftpd** restart



```
[root@server ~]# service vsftpd restart
Shutting down vsftpd:                                      [  OK  ]
Starting vsftpd for vsftpd:                                [  OK  ]
[root@server ~]# _
```

Come back on linux client system and login form sanjay and try again to change directory to /



```
[root@linuxclient ~]# ftp 192.168.1.1
Connected to 192.168.1.1 (192.168.1.1).
220 (vsFTPd 2.2.2)
Name (192.168.1.1:root): sanjay
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> pwd
257 "/"
ftp> cd /
250 Directory successfully changed.
ftp> pwd
257 "/"
ftp> dir
227 Entering Passive Mode (192,168,1,1,228,189).
150 Here comes the directory listing.
-rw-rw-r--    1 500      500           110 Sep 17 22:54 example_file
-rw-r--r--    1 500      500            21 Sep 17 23:06 sample_file
226 Directory send OK.
ftp> bye
221 Goodbye.
[root@linuxclient ~]# _
```

Now normal user will not be able to navigate outside the home directory.

4. SMTP CLIENT

```
/* Copyright (c) 2006-2008 Dovecot authors */

#include "lib.h"
#include "deliver.h"
#include "smtp-client.h"

#include <unistd.h>
#include <sys/wait.h>

structsmtp_client {
	FILE *f;
	pid_tpid;
};

staticstructsmtp_client *smtp_client_devnull(FILE **file_r)
{
	structsmtp_client *client;

	client = i_new(structsmtp_client, 1);
	client->f = *file_r = fopen("/dev/null", "w");
	if (client->f == NULL)
		i_fatal("fopen() failed: %m");
	client->pid = (pid_t)-1;
	return client;
}

static void ATTR_NORETURN
smtp_client_run_sendmail(const char *destination,
			 const char *return_path, intfd)
{
	const char *argv[7], *sendmail_path;

	/* deliver_set's contents may point to environment variables.
	deliver_env_clean() cleans them up, so they have to be copied. */
	sendmail_path = t_strdup(deliver_set->sendmail_path);

	argv[0] = sendmail_path;
	argv[1] = "-i"; /* ignore dots */
	argv[2] = "-f";
	argv[3] = return_path != NULL && *return_path != '\0' ?
		return_path : "<>";
	argv[4] = "--";
	argv[5] = destination;
	argv[6] = NULL;

	if (dup2(fd, STDIN_FILENO) < 0)
		i_fatal("dup2() failed: %m");

	deliver_env_clean();

	(void)execv(sendmail_path, (void *)argv);
	i_fatal("execv(%s) failed: %m", sendmail_path);
}

structsmtp_client *smtp_client_open(const char *destination,
				const char *return_path, FILE **file_r)
{
	structsmtp_client *client;
	intfd[2];
	pid_tpid;

	if (pipe(fd) < 0) {
		i_error("pipe() failed: %m");
		returnsmtp_client_devnull(file_r);
	}

	if ((pid = fork()) == (pid_t)-1) {
		i_error("fork() failed: %m");
		(void)close(fd[0]); (void)close(fd[1]);
		returnsmtp_client_devnull(file_r);
	}
	if (pid == 0) {
		/* child */
		(void)close(fd[1]);
		smtp_client_run_sendmail(destination, return_path, fd[0]);
	}
	(void)close(fd[0]);

	client = i_new(structsmtp_client, 1);
	client->f = *file_r = fdopen(fd[1], "w");
	if (client->f == NULL)
		i_fatal("fdopen() failed: %m");
	return client;
}

intsmtp_client_close(structsmtp_client *client)
{
	int ret = EX_TEMPFAIL, status;
```

```
        fclose(client->f);
        if (client->pid == (pid_t)-1) {
                /* smtp_client_open() failed already */
        } else if (waitpid(client->pid, &status, 0) < 0)
                i_error("waitpid() failed: %m");
        else if (WIFEXITED(status)) {
                ret = WEXITSTATUS(status);
                if (ret != 0) {
                        i_error("Sendmail process terminated abnormally, "
                                "exit status %d", ret);
                }
        } else if (WIFSIGNALED(status)) {
                i_error("Sendmail process terminated abnormally, "
                                "signal %d", WTERMSIG(status));
        } else if (WIFSTOPPED(status)) {
                i_error("Sendmail process stopped, signal %d",
                        WSTOPSIG(status));
        } else {
                i_error("Sendmail process terminated abnormally, "
                        "return status %d", status);
        }

        i_free(client);
        return ret;
}
```

QUESTION BANK

1. How can I prevent unauthorized laptops from using a network that uses DHCP for dynamic addressing?
2. Can a BOOTP client boot from a DHCP server?
3. What is DHCP’s purpose?
4. Can DHCP support remote access?
5. What is a DHCP lease?
6. What is DHCP Spoofing?
7. What is a MAC address?
8. Can a DHCP server back up another DHCP server?
9. What protocol and port does DHCP use?
10. How many DHCP packets are exchanged between a client and a server before the client receives an IP address?
11. What type of packet is a DHCP Discover packet?
12. What is an IP Helper address feature and why is it required in a DHCP environment?
13. What is the role of DNS ?
14. Which are the important configuration files for DNS server ?
15. On which port DNS server works ?
16. What is round robin DNS?
17. What is Primary name server or primary master server?
18. What is Root name server?
19. What do you mean by "Resource Records"?
20. Explain "Resource Records/ TTL/SOA/A/NS/CNAME/SOA/PTR/MX/HINFO/PTR”?
21. What are the defaults ports used in ftp server ?
22. What is default directory for ftp/Anonymous user ?
23. How to enable chroot environment in vsftpdserver ?
24. How to enable only limited/allowed users are able to login via ftp ?
25. How To limit the data transfer rate, number of clients & connectionsper IP for local users ?
26. What is the difference between TFTP and FTP servers?
27. I want to copy multiple files with out prompting for any info, how can I do that one?
28. How to disable certain FTP commands?
29. What is MTA and it’s role in mailing system ?
30. What is MDA&MUA ?
31. How to send a test mail from command line ?
32. What is an Open mail relay ?
33. What is Greylisting ?
34. What is the importance of SPF records in mail servers ?
35. What is the use of Domain Keys(DKIM) in mail servers ?
36. What is the role of Anti-Spam SMTP Proxy (ASSP) in mail server ?