

TELE5330- LINUX PROJECT

Our Project aims at implementing a Network of our startup company DMS. We three team members have planned our design comprising of three main Servers: DNS, DHCP, Web-firewall Backup Server. When any client enters the network, it obtains an IP address from our DHCP server and can access a web page whose address is resolved by the DNS and the web server loads the web page onto the client machine.

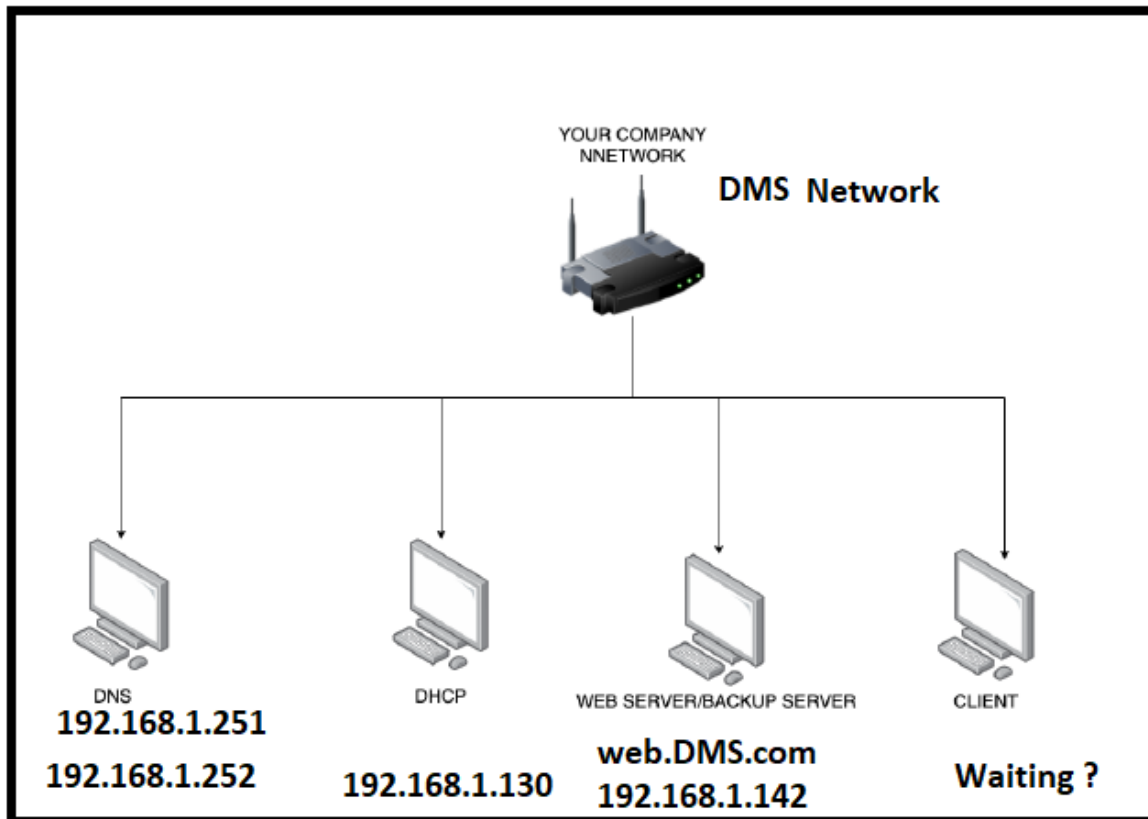
Parts Implemented and Configured by team members as below:

Samrudhi Prabhulkar: DHCP Server and ARP Spoofing

Miloni Kapadia: DNS (Master-Slave) and IPSEC VPN tunneling

Divya Jadhav: Web-Firewall-Backup Server and NFS Server

1. HIGH LEVEL DIAGRAM



2. BEHAVIOR OF THE PROTOCOL

PART 1: DHCP SERVER

The Dynamic Host Configuration Protocol provides configuration parameters to the hosts. It consists of two components: server and host.

There are three types of DHCP: automatic, dynamic and manual.

The main function of the DHCP server is to assign IP addresses to the hosts in the network. DHCP server processes the DHCP messages that a client sends. The following messages are received by the server:

DHCPDISCOVER
DHCPREQUEST
DHCPDECLINE
DHCPRELEASE
DHCPINFORM

PART 2: DOMAIN NAME SYSTEM (DNS)

- DNS: The Domain Name System is nothing but a hierarchical and decentralized naming system which translates memorized domain names to the numerical IP addresses to locate computer/server. So the main task of DNS is to assign domain name to the respective server IPs as it is difficult for the client to remember the IP addresses of every server. So DNS make it easier by assigning domain name to every server. The types of record stored in the DNS database are Start of Authority (SOA), name server (NS), IP addresses (A for IPv4 and AAAA for IPv6), SMTP mail exchanges (MX), pointers for reverse DNS lookups (PTR) and domain name aliases (CNAME).
- BIND9: Berkely Internet Name Domain software is mainly used in Linux systems for translating domain names into IP addresses by storing zonal files such as forward and reverse zones in the format of DNS database.
- Forward Zone: This zonal file of BIND9 is mainly used for translating domain names to IP addresses.
- Reverse Zone: This zonal file of BIND9 is mainly used for translating IP addresses to domain names which is nothing but the reversing process of forward zone.
- Now, if we consider our example then as soon as the client enters into our start up i.e, DMS.com on its web-browser, translation process from domain names to IP addresses has been done by DNS and IP 192.168.1.142 has been return for the same. So now client can be able to access the web-page of DMS.com by record name “web” or IP address 192.168.1.142

PART 3: WEB SERVER

- We have designed an html page hosting on our Webserver with IP address: 192.168.1.142
- Whenever a client wants to access our webpage, it can be accessed using web.DMS.com our domain via any web browsers like Firefox.
- DNS resolves our fully qualified domain name into our IP address 192.168.1.142 and the client can access our hosted webpage web.DMS.com

PART 4: FIREWALL

- Firewall is a network security technology providing the security to our webpage by applying different security technologies.
- It can be used for securing our data on webpage by limiting the users by methods like SSH, preventing the client from accessing files using FTP or TELNET.

PART 5: BACKUP SERVER

- BACKUP SERVER is a server that provides the backup services. It basically ensures the data remains intact in case of any failures.
- In our project we have implemented backing up of for our webpage apache services and sending the backup file in a tar format with current time stamp (date wise) to our DHCP server.
- We have used rsync command for copying files and sending via ssh to the dhcp server.

PART 6: NFS SERVER

- It is a network file system server which helps to share files and folders over a network
- We can share the files by mounting the nfs share either temporarily or permanently.

PART 7: VPN

Concept of IPsec VPN tunnel is nothing but with tunnel mode the entire packet will be protected by IPsec as IPsec will wraps the original packet, encrypts it and send it to the other side of the VPN tunnel

PART 8: ARP SPOOFING

It is a protocol used for resolution of network layer addresses into Link layer addresses.

3. HIERARCHY

- The Client will first access into the network of our startup company DMS.
- The DHCP assigns the IP addresses from the pool of IP Addresses:
- After an IP address is assigned by the DHCP server. The client will browse the domain name web.DMS.com
- Then the DNS resolves the domain name to IP address giving the IP address of our webbrowser i.e 192.168.1.142
- The client can now access the web page web.DMS.com.
- The DNS can also map the IP address to the domain name as we have created reverse entries for it.
- Hence the client can now browse the web page the either way.
- A firewall is configured at the network. If the client tries to share any files or transfer any files by accessing through telnet or ftp then those services are blocked at our end.
- If there is any update of the web page files then they will be backed up and send to DHCP server as a backup.
- The backup is scheduled every day around 9am in the morning

4. COMMANDS USED

PART 1: DHCP:

- Install the isc-dhcp by entering the command `sudo apt install isc-dhcp`
- Statically assign IPs to the DHCP server. Enter `sudo nano /etc/netplan/50-cloud-init.yaml` and configure the following

```
network:
    ethernet:
        ens33:
            addresses: [192.168.1.130/24, '2001:1200:1100:1000::99/64']
            gateway4: 192.168.1.1
            gateway6: 2001:1200:1100:1000::1
            dhcp4: false
            dhcp6: false
```

- Save the file and run it with `sudo netplan apply`
- Open the `dhcpd.conf` file and do the following changes for IPv4 addresses.
- There are IP addresses assigned to the respective MAC addresses for the servers.

```
dhcserver@ubuntu: ~
File Edit View Search Terminal Help
GNU nano 2.9.3 /etc/dhcp/dhcpd.conf

# dhcpd.conf
#
# Sample configuration file for ISC dhcpd
#
# Attention: If /etc/ltsp/dhcpd.conf exists, that will be used as
# configuration file instead of this file.
#
# option definitions common to all supported networks...
option domain-name "DMS.com";
option domain-name-servers 192.168.1.252, 192.168.1.251;

default-lease-time 600;
max-lease-time 7200;

# The ddns-updates-style parameter controls whether or not the server will
# attempt to do a DNS update when a lease is confirmed. We default to the
# behavior of the version 2 packages ('none', since DHCP v2 didn't
# have support for DDNS.)
ddns-update-style none;

# If this DHCP server is the official DHCP server for the local
# network, the authoritative directive should be uncommented.
authoritative;
```

```
# This declaration allows BOOTP clients to get dynamic addresses,
# which we don't really recommend.

#subnet 10.254.239.32 netmask 255.255.255.224 {
#  range dynamic-bootp 10.254.239.40 10.254.239.60;
#  option broadcast-address 10.254.239.31;
#  option routers rtr-239-32-1.example.org;
#}

# A slightly different configuration for an internal subnet.
subnet 192.168.1.0 netmask 255.255.255.0 {
  range 192.168.1.100 192.168.1.200;
  option domain-name-servers 192.168.1.252, 192.168.1.251;
  option domain-name "DMS.com";
  option subnet-mask 255.255.255.0;
  option routers 192.168.1.1;
  option broadcast-address 192.168.1.254;
  default-lease-time 600;
  max-lease-time 7200;
}

# Hosts which require special configuration options can be listed in
# host statements.  If no address is specified, the address will be
# allocated dynamically (if possible), but the host-specific information
# will still come from the host declaration.

#web server, define static IP
host webbrowser {
  hardware ethernet 00:0c:29:72:a3:b2;
  fixed-address 192.168.1.142;
}
```

```
#client, define static IP
host client1 {
  hardware ethernet 00:0c:29:ce:a2:9c;
  fixed-address 192.168.1.138;
}

#dns server, define static IP
host Server.DMS.com {
  hardware ethernet 00:0c:29:4c:96:b8;
  fixed-address 192.168.1.252;
}

#dns server, define static IP
host Client.DMS.com {
  hardware ethernet 00:0c:29:aa:1e:d4;
  fixed-address 192.168.1.251;
}

#host passacaglia {
#  hardware ethernet 0:0:c0:5d:bd:95;
#  filename "vmunix.passacaglia";
#  server-name "toccata.example.com";
#}
```

For IPv6 addresses the configuration is as follows:

```
# The subnet where the server is attached
# (i.e., the server has an address in this subnet)
subnet6 2001:1200:1100:1000::/64 {
#     # Two addresses available to clients
#     # (the third client should get NoAddrsAvail)
    range6 2001:1200:1100:1000::200 2001:1200:1100:1000::300;
#
#     # Use the whole /64 prefix for temporary addresses
#     # (i.e., direct application of RFC 4941)
    range6 2001:1200:1100:1000::/64 temporary;
#
#     # Some /64 prefixes available for Prefix Delegation (RFC 3633)
    prefix6 2001:1200:1100:1000:: 2001:1200:1100:1000::/64;
    option dhcp6.name-servers 2001:1200:1100:1000::252, 2001:1200:1100:1000::251;
    option dhcp6.domain-search "DMS.com";
}

# A second subnet behind a relay agent
#subnet6 3ffe:501:ffff:101::/64 {
#    range6 3ffe:501:ffff:101::10 3ffe:501:ffff:101::11;
#
#    # Override of the global definitions,
#    # works only when a resource (address or prefix) is assigned
#    option dhcp6.name-servers 3ffe:501:ffff:101:200:ff:fe00:3f3e;
#
#
```

Save the file and then run the following commands

```
sudo systemctl restart isc-dhcp-server
sudo systemctl restart isc-dhcp-server6
```

PART 2: DOMAIN NAME SYSTEM (DNS)

- We are preparing Master- Slave hierarchy for DNS. So we will be requiring two Virtual machines for that.
 - DNS Server(master):
 - IPv4: 192.168.1.252
 - IPv6: 2001:1200:1100:1000::252
 - DNS Slave:
 - IPv4: 192.168.1.252
 - IPv6: 2001:1200:1100:1000::252
- After that our first task would be to install bind9 package on each machines for DNS configuration.
- After that we need to create zone entries under configuration file /etc/bind/named.conf.local so that any request for domain name “DMS.com” can be handled by zone files created by this file. Here, we have created three zonal files one is forward and two is reverse (IPv4 & IPv6). We have included default zone file to the our local file so we are commenting that line from named.conf file.

```
GNU nano 2.9.3                                named.conf.local
//
// Do any local configuration here
//
// Consider adding the 1918 zones here, if they are not used in your
// organization
//include "/etc/bind/zones.rfc1918";

view "internal" {
    match-clients {
        localhost;
        192.168.1.0/24;
    };

    zone "DMS.com" {
        type master;
        file "/etc/bind/for.DMS.com";
        allow-update { none; };
        also-notify { 192.168.1.251; 2001:1200:1100:1000::251; };
    };

    zone "1.168.192.in-addr.arpa" {
        type master;
        file "/etc/bind/rev.DMS.com";
        allow-update { none; };
        also-notify { 192.168.1.251; 2001:1200:1100:1000::251; };
    };

    zone "0.0.0.1.0.0.1.1.0.0.2.1.1.0.0.2.ip6.arpa" {
        type master;
        file "/etc/bind/rev6.DMS.com";
        allow-update { none; };
        also-notify { 192.168.1.251; 2001:1200:1100:1000::251; };
    };

include "/etc/bind/named.conf.default-zones";
};
```

Read 38 lines

```
GNU nano 2.9.3                                named.conf
// This is the primary configuration file for the BIND DNS server named.
//
// Please read /usr/share/doc/bind9/README.Debian.gz for information on the
// structure of BIND configuration files in Debian, *BEFORE* you customize
// this configuration file.
//
// If you are just adding zones, please do that in /etc/bind/named.conf.local

include "/etc/bind/named.conf.options";
include "/etc/bind/named.conf.local";
//include "/etc/bind/named.conf.default-zones";

GNU nano 2.9.3                                named.conf.options
options {
    directory "/var/cache/bind";

    // If there is a firewall between you and nameservers you want
    // to talk to, you may need to fix the firewall to allow multiple
    // ports to talk.  See http://www.kb.cert.org/vuls/id/800113

    // If your ISP provided one or more IP addresses for stable
    // nameservers, you probably want to use them as forwarders.
    // Uncomment the following block, and insert the addresses replacing
    // the all-0's placeholder.

    forwarders {
        8.8.8.8;
        8.8.4.4;
    };

    allow-query { localhost; 192.168.1.0/24; 2001:1200:1100:1000::/64; };
    allow-transfer { localhost; 192.168.1.251; 2001:1200:1100:1000::251; };
    allow-recursion { localhost; 192.168.1.0; 2001:1200:1100:1000::/64; };

    //=====
    // If BIND logs error messages about the root key being expired,
    // you will need to update your keys.  See https://www.isc.org/bind-keys
    //=====
    dnssec-validation auto;

    auth-nxdomain no;    # conform to RFC1035
    listen-on-v6 { any; };
};

Read 32 lines
```

into

After that we have created forward zone file for.DMS.com under path /etc/bind/ which is already specified in named.conf.local file. Enter name server entries for both master and slave. Then enter IPv4 and IPv6 entries for each domain and the web-browser as below.

```
GNU nano 2.9.3                                for.DMS.com
$TTL 86400
@      IN      SOA      Server.DMS.com. root.DMS.com. (
        2018070801
        3600
        1800
        604800
        86400
)

        IN      NS       Server.DMS.com.
        IN      NS       Client.DMS.com.

Server IN      A         192.168.1.252
Client IN      A         192.168.1.251
web     IN      A         192.168.1.142
Server  IN      AAAA      2001:1200:1100:1000::252
Client  IN      AAAA      2001:1200:1100:1000::251
web     IN      AAAA      2001:1200:1100:1000::300
```


Then create reverse zone file for IPv4. Point IP address to respective entities as below.

```
GNU nano 2.9.3 rev.DMS.com
$TTL 86400
@      IN      SOA      Server.DMS.com. root.DMS.com.  (
                                2018070801
                                3600
                                1800
                                604800
                                86400
)

      IN      NS       Server.DMS.com.
      IN      NS       Client.DMS.com.

252    IN      PTR      Server.
251    IN      PTR      Client.
142    IN      PTR      web.
```

Then create reverse zone file for IPv6. Point IPv6 to respective entities as below.

```
GNU nano 2.9.3 rev6.DMS.com
$TTL 86400
@      IN      SOA      Server.DMS.com. root.DMS.com.  (
                                2018070801
                                3600
                                1800
                                604800
                                86400
)

      IN      NS       Server.DMS.com.
      IN      NS       Client.DMS.com.

2.5.2.0.0.0.0.0.0.0.0.0.0.0.0.0 IN      PTR      Server.
1.5.2.0.0.0.0.0.0.0.0.0.0.0.0.0 IN      PTR      Client.
0.0.3.0.0.0.0.0.0.0.0.0.0.0.0.0 IN      PTR      web.
```

After that restart BIND9 service and go to Slave machine. In Slave machine add below entities in named.conf.local file for establishing of zones. Also, we have specified path for zone files as /var/cache/bind/ where files from master DNS will be auto transferred. Here also we have added default zone files so we need to comment it at named.conf file.

```
GNU nano 2.9.3                                named.conf.local
//
// Do any local configuration here
//
// Consider adding the 1918 zones here, if they are not used in your
// organization
//include "/etc/bind/zones.rfc1918";

view "internal" {
    match-clients {
        localhost;
        192.168.1.0/24;
        2001:1200:1100:1000::/64;
    };

    zone "DMS.com" {
        type slave;
        file "/var/cache/bind/for.DMS.com";
        masters { 192.168.1.252; 2001:1200:1100:1000::252; };
    };

    zone "1.168.192.in-addr.arpa" {
        type slave;
        file "/var/cache/bind/rev.DMS.com";
        masters { 192.168.1.252; 2001:1200:1100:1000::252; };
    };

    zone "0.0.0.1.0.0.1.1.0.0.2.1.1.0.0.2.ip6.arpa" {
        type slave;
        file "/var/cache/bind/rev6.DMS.com";
        masters { 192.168.1.252; 2001:1200:1100:1000::252; };
    };

include "/etc/bind/named.conf.default-zones";
};

GNU nano 2.9.3                                named.conf
// This is the primary configuration file for the BIND DNS server named.
//
// Please read /usr/share/doc/bind9/README.Debian.gz for information on the
// structure of BIND configuration files in Debian, *BEFORE* you customize
// this configuration file.
//
// If you are just adding zones, please do that in /etc/bind/named.conf.local

include "/etc/bind/named.conf.options";
include "/etc/bind/named.conf.local";
//include "/etc/bind/named.conf.default-zones";
```

Allow forwarders as google's DNS into file named.conf.options

```
GNU nano 2.9.3                                named.conf.options
options {
    directory "/var/cache/bind";

    // If there is a firewall between you and nameservers you want
    // to talk to, you may need to fix the firewall to allow multiple
    // ports to talk.  See http://www.kb.cert.org/vuls/id/800113

    // If your ISP provided one or more IP addresses for stable
    // nameservers, you probably want to use them as forwarders.
    // Uncomment the following block, and insert the addresses replacing
    // the all-0's placeholder.

    forwarders {
        8.8.8.8;
        8.8.4.4;
    };

    //=====
    // If BIND logs error messages about the root key being expired,
    // you will need to update your keys.  See https://www.isc.org/bind-keys
    //=====
    dnssec-validation auto;

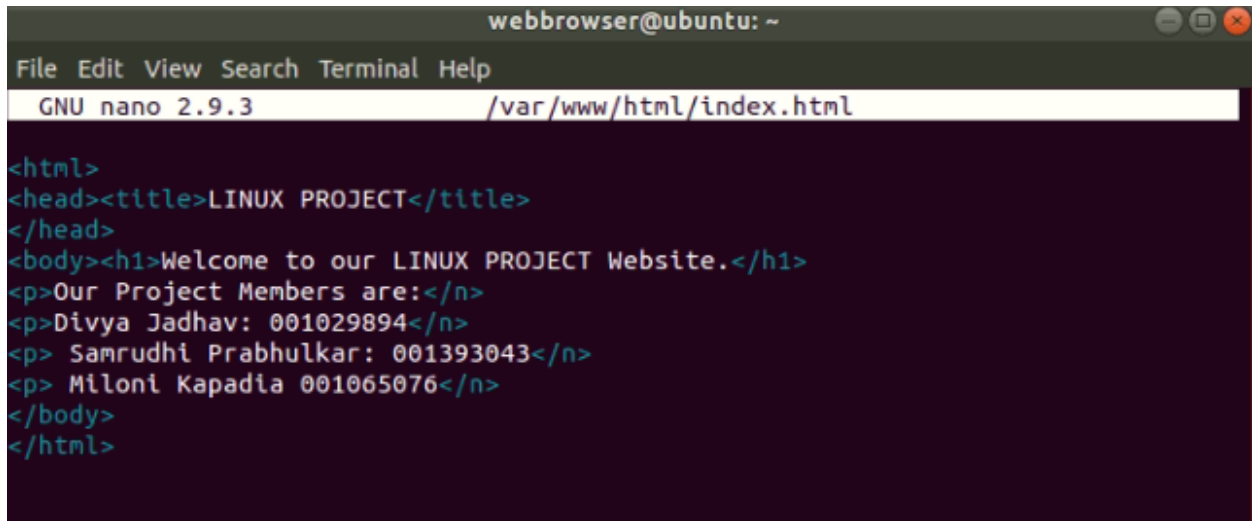
    auth-nxdomain no;      # conform to RFC1035
    listen-on-v6 { any; };
};
```

Then restart bind9 service at both master and slave side. By doing this zone files from master DNS will be transfer to Slave at specified location /var/cache/bind. Then check for the same as below.

```
mkapadia@Client:~$ sudo ls -l /var/cache/bind/
total 28
-rw-r--r-- 1 bind bind 501 Apr 15 13:38 for.DNS.com
-rw-r--r-- 1 bind bind 1349 Apr 15 13:38 internal.mkeys
-rw-r--r-- 1 bind bind 512 Apr 15 13:38 internal.mkeys.jnl
-rw-r--r-- 1 bind bind 821 Mar 16 15:45 managed-keys.bind
-rw-r--r-- 1 bind bind 512 Mar 16 15:45 managed-keys.bind.jnl
-rw-r--r-- 1 bind bind 476 Apr 15 13:38 rev6.DNS.com
-rw-r--r-- 1 bind bind 348 Apr 15 13:38 rev.DNS.com
mkapadia@Client:~$
```

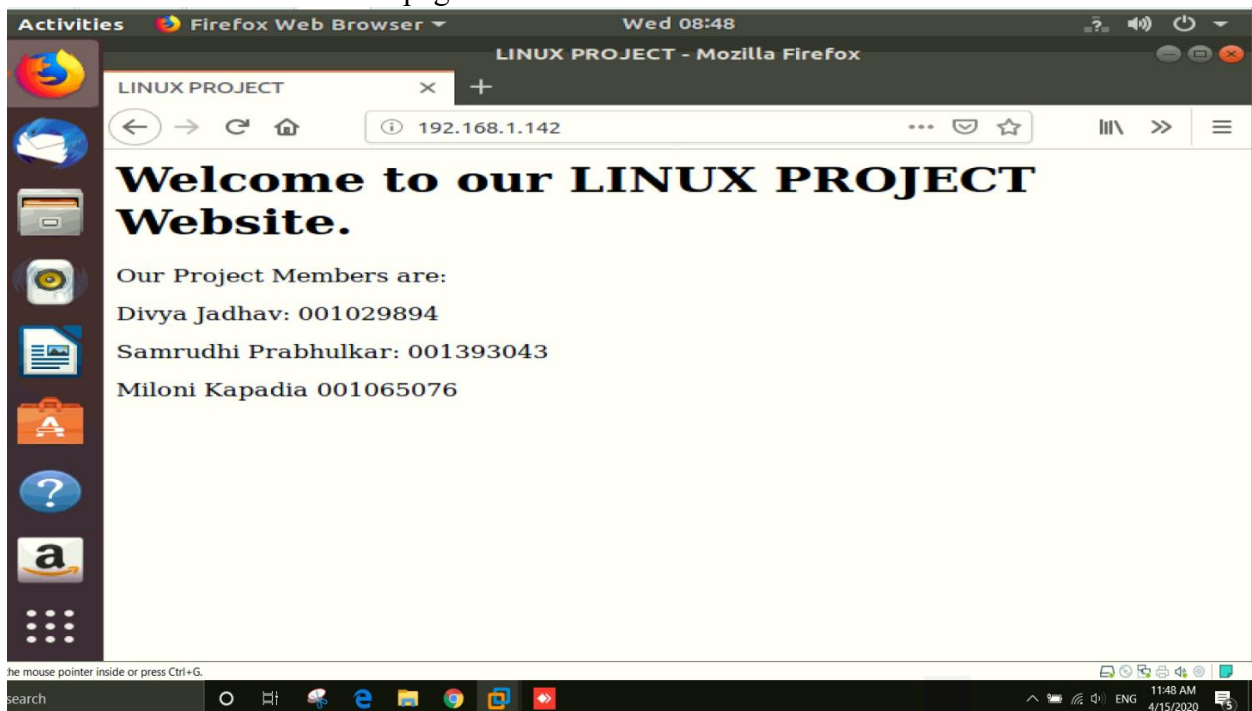
Part 3: Web Server

We have written an html script for the index.html page at the below path /var/www/html/index.html:

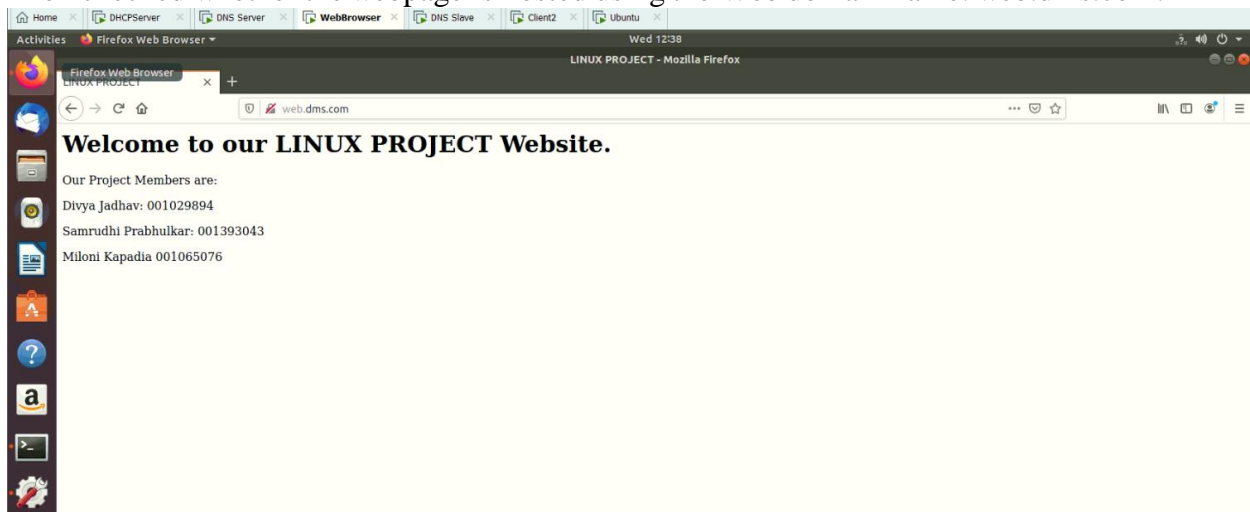


```
webbrowser@ubuntu: ~  
File Edit View Search Terminal Help  
GNU nano 2.9.3 /var/www/html/index.html  
  
<html>  
<head><title>LINUX PROJECT</title>  
</head>  
<body><h1>Welcome to our LINUX PROJECT Website.</h1>  
<p>Our Project Members are:</p>  
<p>Divya Jadhav: 001029894</p>  
<p> Samrudhi Prabhulkar: 001393043</p>  
<p> Miloni Kapadia 001065076</p>  
</body>  
</html>
```

Then checked whether the webpage is hosted on the Webbrowser's IP:

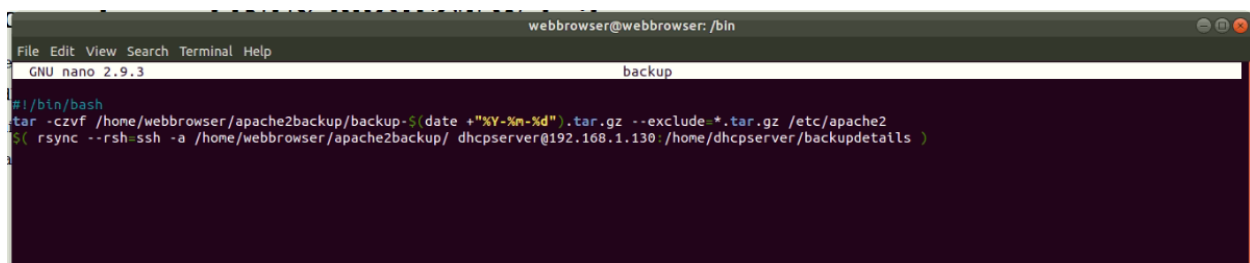


Then checked whether the webpage is hosted using the Web domain name: web.dms.com:



Part 4: Backup Server

- Written below a BASH script where the /etc/apache web server conf files are backed up and saved to the /bin/backup file
- I have used tar command for creating a zip format of the file with czvf options
 - c: create a archive file, -z: to create a compressed gzip file, -v: show the progress of archive file, -f: to enter filename of archive file.
- The tar file created is of current time stamp with date Year-Month-Date format.
- Rsync protocol is used for copying the files to/from the other host over a remote shell (rsh)
- The current files are sent using secured shell protocol ssh by installing openssh-server at the server side and openssh-client at client side.



Scheduled the backup at every 9am in the morning:

```
webbrowser@webbrowser: ~
File Edit View Search Terminal Help
GNU nano 2.9.3 /tmp/crontab.8mwqEf/crontab

# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h dom mon dow command
0 9 * * * /bin/backup
```

Checked in DHCP server if the files transferred are present:

```
dhcpserver@ubuntu:~$ cd backupdetails/
dhcpserver@ubuntu:~/backupdetails$ ls
apache2backup  backup-2020-04-12.tar.gz  backup-2020-04-14.tar.gz
backup         backup-2020-04-13.tar.gz  backup-2020-04-15.tar.gz
dhcpserver@ubuntu:~/backupdetails$
```

Part 5: FIREWALL

- `sudo iptables -L` is for viewing IP tables
- `sudo iptables -A INPUT -m conntrack --ctstate ESTABLISHED,RELATED -j ACCEPT`: For setting up current connections.
- `SUDO IPTABLES -A INPUT -p tcp --dport ssh -j ACCEPT`
- `SUDO IPTABLES -A INPUT -p tcp --dport 80 -j ACCEPT`: For establishing tcp port connection and 80 through which packet is transmitted
- `SUDO IPTABLES -A INPUT -p icmp --icmp-type 8 -j ACCEPT`
- For accepting echo request: `SUDO IPTABLES -A OUTPUT -p icmp --icmp-type 0 -j ACCEPT`
- `SUDO IPTABLES -A INPUT -p tcp --dport 20 -j DROP` // for ftp data
- `SUDO IPTABLES -A INPUT -p tcp --dport 21 -j DROP` // for ftp control :For blocking both ftp data and control ports
- `SUDO IPTABLES -A INPUT -d -p tcp --dport 23 -j DROP` // for blocking TELNET services.

```
webbrowser@webbrowser: ~  
File Edit View Search Terminal Help  
target prot opt source destination  
webbrowser@webbrowser:~$ sudo iptables -A INPUT -m conntrack --ctstate ESTABLISHED,RELATED -j ACCEPT  
webbrowser@webbrowser:~$ sudo iptables -A INPUT -p tcp --dport ssh -j ACCEPT  
webbrowser@webbrowser:~$ sudo iptables -A INPUT -p tcp --dport 80 -j ACCEPT  
webbrowser@webbrowser:~$ sudo iptables -A INPUT -p icmp --icmp-type 8 -j ACCEPT  
webbrowser@webbrowser:~$ sudo iptables -A INPUT -p icmp --icmp-type 0 -j ACCEPT  
webbrowser@webbrowser:~$ sudo iptables -A INPUT -p tcp --dport 20 -j DROP  
webbrowser@webbrowser:~$ sudo iptables -A INPUT -p tcp --dport 21 -j DROP  
webbrowser@webbrowser:~$ sudo iptables -A INPUT -d -p tcp --dport 23 -j DROP  
Bad argument 'tcp'  
Try 'iptables -h' or 'iptables --help' for more information.  
webbrowser@webbrowser:~$ sudo iptables -A INPUT -p tcp --dport 23 -j DROP  
webbrowser@webbrowser:~$ sudo iptables -L  
Chain INPUT (policy DROP)  
target prot opt source destination  
ufw-before-logging-input all -- anywhere anywhere  
ufw-before-input all -- anywhere anywhere  
ufw-after-input all -- anywhere anywhere  
ufw-after-logging-input all -- anywhere anywhere  
ufw-reject-input all -- anywhere anywhere  
ufw-track-input all -- anywhere anywhere  
ACCEPT all -- anywhere anywhere ctstate RELATED,ESTABLISHED  
ACCEPT tcp -- anywhere anywhere tcp dpt:ssh  
ACCEPT tcp -- anywhere anywhere tcp dpt:http  
ACCEPT icmp -- anywhere anywhere icmp echo-request  
ACCEPT icmp -- anywhere anywhere icmp echo-reply  
DROP tcp -- anywhere anywhere tcp dpt:ftp-data  
DROP tcp -- anywhere anywhere tcp dpt:ftp  
DROP tcp -- anywhere anywhere tcp dpt:telnet  
Chain FORWARD (policy DROP)  
target prot opt source destination  
ufw-before-logging-forward all -- anywhere anywhere  
ufw-before-forward all -- anywhere anywhere  
ufw-after-forward all -- anywhere anywhere  
ufw-after-logging-forward all -- anywhere anywhere
```

Part 6: NFS(Network File Share)

Installed NFS packages at server as well as client side:

```
webbrowser@webbrowser: ~  
File Edit View Search Terminal Help  
webbrowser@webbrowser:~$ sudo apt-get install nfs-kernel-server  
Reading package lists... Done  
Building dependency tree  
Reading state information... Done  
The following additional packages will be installed:  
  keyutils libnfsidmap2 libtirpc1 nfs-common rpcbind  
Suggested packages:  
  open-iscsi watchdog  
The following NEW packages will be installed:  
  keyutils libnfsidmap2 libtirpc1 nfs-common nfs-kernel-server rpcbind  
0 upgraded, 6 newly installed, 0 to remove and 140 not upgraded.  
Need to get 490 kB of archives.  
After this operation, 1,703 kB of additional disk space will be used.  
Do you want to continue? [Y/n] Y  
Get:1 http://us.archive.ubuntu.com/ubuntu bionic/main amd64 keyutils amd64 1.5.9-9.2ubuntu2 [47.9 kB]  
Get:2 http://us.archive.ubuntu.com/ubuntu bionic/main amd64 libnfsidmap2 amd64 0.25-5.1 [27.2 kB]  
Get:3 http://us.archive.ubuntu.com/ubuntu bionic-updates/main amd64 libtirpc1 amd64 0.2.5-1.2ubuntu0.1 [75.7 kB]  
Get:4 http://us.archive.ubuntu.com/ubuntu bionic/main amd64 rpcbind amd64 0.2.3-0.6 [40.6 kB]  
Get:5 http://us.archive.ubuntu.com/ubuntu bionic-updates/main amd64 nfs-common amd64 1:1.3.4-2.1ubuntu5.2 [205 kB]
```



```
client1@ubuntu: ~  
File Edit View Search Terminal Help  
client1@ubuntu:~$ sudo apt-get install nfs-common  
Reading package lists... Done  
Building dependency tree  
Reading state information... Done  
The following additional packages will be installed:  
  keyutils libnfsidmap2 libtirpc1 rpcbind  
Suggested packages:  
  open-iscsi watchdog  
The following NEW packages will be installed:  
  keyutils libnfsidmap2 libtirpc1 nfs-common rpcbind  
0 upgraded, 5 newly installed, 0 to remove and 309 not upgraded.  
Need to get 397 kB of archives.  
After this operation, 1,358 kB of additional disk space will be used.  
Do you want to continue? [Y/n] Y  
Get:1 http://us.archive.ubuntu.com/ubuntu bionic/main amd64 keyutils amd64 1.5.9-9.2ubuntu2 [47.9 kB]  
Get:2 http://us.archive.ubuntu.com/ubuntu bionic/main amd64 libnfsidmap2 amd64 0.2.5-1 [27.2 kB]  
Get:3 http://us.archive.ubuntu.com/ubuntu bionic-updates/main amd64 libtirpc1 amd64 0.2.5-1.2ubuntu0.1 [75.7 kB]  
Get:4 http://us.archive.ubuntu.com/ubuntu bionic/main amd64 rpcbind amd64 0.2.3-0.6 [40.6 kB]  
Get:5 http://us.archive.ubuntu.com/ubuntu bionic-updates/main amd64 nfs-common amd64 1:1.3.4-2.1ubuntu5.2 [205 kB]
```

Made two directories /home/webbrowser/public in read only mode and /home/webbrowser/private in read write modes:

```
webbrowser@webbrowser:~$ sudo chmod 755 public  
webbrowser@webbrowser:~$ sudo chmod 777 private  
webbrowser@webbrowser:~$ sudo nano /etc/exports  
Use "fg" to return to nano.  
[8]+ Stopped sudo nano /etc/exports  
webbrowser@webbrowser:~$ sudo nano /etc/exports  
webbrowser@webbrowser:~$ sudo cat /etc/exports  
# /etc/exports: the access control list for filesystems which may be exported  
# to NFS clients. See exports(5).  
#  
# Example for NFSv2 and NFSv3:  
# /srv/homes hostname1(rw,sync,no_subtree_check) hostname2(ro,sync,no_subtree_check)  
#  
# Example for NFSv4:  
# /srv/nfs4 gss/krb5l(rw,sync,fsid=0,crossmnt,no_subtree_check)  
# /srv/nfs4/homes gss/krb5l(rw,sync,no_subtree_check)  
#  
/home/webbrowser/public *(ro,sync,no_subtree_check)  
/home/webbrowser/private 192.168.1.138(rw,sync,no_subtree_check)  
webbrowser@webbrowser:~$ sudo exportfs -arvf  
exporting 192.168.1.138:/home/webbrowser/private  
exporting */home/webbrowser/public  
webbrowser@webbrowser:~$ sudo systemctl start nfs-kernel-server  
webbrowser@webbrowser:~$ sudo systemctl enable nfs-kernel-server  
Synchronizing state of nfs-kernel-server.service with SysV service script with /lib/systemd/systemd-sysv-install.  
Executing: /lib/systemd/systemd-sysv-install enable nfs-kernel-server  
webbrowser@webbrowser:~$ sudo systemctl status nfs-kernel-server  
● nfs-server.service - NFS server and services  
   Loaded: loaded (/lib/systemd/system/nfs-server.service; enabled; vendor preset: enabled)  
   Active: active (exited) since Wed 2020-04-15 09:26:15 PDT; 3h 56min ago  
 Main PID: 4486 (code=exited, status=0/SUCCESS)  
    Tasks: 0 (limit: 2295)  
   CGroup: /system.slice/nfs-server.service  
  
Apr 15 09:26:14 webbrowser systemd[1]: Starting NFS server and services...  
Apr 15 09:26:15 webbrowser systemd[1]: Started NFS server and services.  
webbrowser@webbrowser:~$ ls  
apache2backup Desktop Documents Downloads examples.desktop index.html index.html.save Music Pictures public Public Templates Videos  
webbrowser@webbrowser:~$ cd private/  
webbrowser@webbrowser:~/private$ ls  
divya  
webbrowser@webbrowser:~/private$
```

Went on the client side and checked the exported mount and mounted the exported directories:

```
client1@client1:/$ cd mnt  
client1@client1:/mnt$ ls  
private public  
client1@client1:/mnt$ cd -  
client1@client1:/$ sudo mount -t nfs 192.168.1.142:/home/webbrowser/public /mnt/public  
[sudo] password for client1:  
client1@client1:/$ sudo mount -t nfs 192.168.1.142:/home/webbrowser/private /mnt/private  
client1@client1:/$ mount
```

Checked the temporary mount using mount -a:


```

client1@client1: ~
File Edit View Search Terminal Help
mqueue on /dev/mqueue type mqueue (rw,relatime)
systemd-1 on /proc/sys/fs/binfmt_misc type autofs (rw,relatime,fd=35,pgrp=1,timeout=0,minproto=5,maxproto=5,direct,pipe_ino=25661)
debugfs on /sys/kernel/debug type debugfs (rw,relatime)
fusectl on /sys/fs/fuse/connections type fusectl (rw,relatime)
configfs on /sys/kernel/config type configfs (rw,relatime)
/var/lib/napd/snaps/core_7270.snap on /snap/core/7270 type squashfs (ro,nodev,relatime,x-gdu.hide)
vmware-vmblock on /run/vmblock-fuse type fuse.vmware-vmblock (rw,relatime,user_id=0,group_id=0,default_permissions,allow_other)
tmpfs on /run/user/121 type tmpfs (rw,nosuid,nodev,relatime,size=201340k,mode=700,uid=121,gid=125)
/var/lib/napd/snaps/core18_1066.snap on /snap/core18/1066 type squashfs (ro,nodev,relatime,x-gdu.hide)
/var/lib/napd/snaps/gtk-common-themes_1313.snap on /snap/gtk-common-themes/1313 type squashfs (ro,nodev,relatime,x-gdu.hide)
/var/lib/napd/snaps/gnome-3-28-1804_67.snap on /snap/gnome-3-28-1804/67 type squashfs (ro,nodev,relatime,x-gdu.hide)
/var/lib/napd/snaps/gnome-calculator_406.snap on /snap/gnome-calculator/406 type squashfs (ro,nodev,relatime,x-gdu.hide)
/var/lib/napd/snaps/gnome-characters_296.snap on /snap/gnome-characters/296 type squashfs (ro,nodev,relatime,x-gdu.hide)
/var/lib/napd/snaps/gnome-logs_61.snap on /snap/gnome-logs/61 type squashfs (ro,nodev,relatime,x-gdu.hide)
/var/lib/napd/snaps/gnome-system-monitor_100.snap on /snap/gnome-system-monitor/100 type squashfs (ro,nodev,relatime,x-gdu.hide)
tmpfs on /run/user/1000 type tmpfs (rw,nosuid,nodev,relatime,size=201340k,mode=700,uid=1000,gid=1000)
gvfsd-fuse on /run/user/1000/gvfs type fuse.gvfsd-fuse (rw,nosuid,nodev,relatime,user_id=1000,group_id=1000)
192.168.1.142:/home/webbrowser/public on /mnt/public type nfs4 (rw,relatime,vers=4.2,rsize=262144,wsize=262144,namlen=255,hard,proto=tcp,timeo=600,retrans=2,sec=sys,clientaddr=192.168.1.138,local_lock=none,addr=192.168.1.142)
192.168.1.142:/home/webbrowser/private on /mnt/private type nfs4 (rw,relatime,vers=4.2,rsize=262144,wsize=262144,namlen=255,hard,proto=tcp,timeo=600,retrans=2,sec=sys,clientaddr=192.168.1.138,local_lock=none,addr=192.168.1.142)
client1@client1:~$

```

Mounted permanently by writing in /etc/fstab file:

```

client1@client1: ~
File Edit View Search Terminal Help
GNU nano 2.9.3 /etc/fstab Modified
# /etc/fstab: static file system information.
#
# Use 'blkid' to print the universally unique identifier for a
# device; this may be used with UUID= as a more robust way to name devices
# that works even if disks are added and removed. See fstab(5).
#
# <file system> <mount point> <type> <options> <dump> <pass>
# / was on /dev/sda1 during installation
UUID=38bce06a-f5e4-4f59-bdd0-03b464473939 / ext4 errors=remount-ro 0 1
/swapfile none swap sw 0 0
/dev/fd0 /media/floppy0 auto rw,user,noauto,exec,utf8 0 0
192.168.1.142:/home/webbrowser/public /mnt/public nfs defaults,_netdev 0 0
192.168.1.142:/home/webbrowser/private /mnt/private nfs defaults,_netdev 0 0

```

Sent file and checked if the file is being shared from the client to the server:

```

client1@client1:~$ sudo nano /etc/fstab
client1@client1:~$ cd /
client1@client1:/$ cd /mnt
client1@client1:/mnt$ ls
public
client1@client1:/mnt$ cd private/
client1@client1:/mnt/private$ touch divya
client1@client1:/mnt/private$ ls
divya
client1@client1:/mnt/private$

```

Part 7: VPN

- For the implementation of above concept I have implemented VPN server-client concept for which we will be requiring two virtual machines.

- First task would be to install two packages at both the machines which are SSH and strongswan as I am using strongswan tool for this concept.
- After that edit file /etc/sysctl.conf as below at both the machines for IP forwarding.

```
root@client1:~# cat >> /etc/sysctl.conf << EOF
> echo net.ipv4.ip_forward = 1
> net.ipv4.conf.all.accept_redirects = 0
> net.ipv4.conf.all.send_redirects = 0
> EOF
root@client1:~#
```

Then create unique key by below command and copy that key and paste it along with IP addressing of source to destination under file /etc/ipsec.secrets at both the machines

```
root@webbrowser:~# openssl rand -base64 64
7to2SeybQuEuW5vkUICx5a0HuDxuQAz9nnccMZjAx0LIck4hHDgaRoRU5rQK6LA
Cl9VongobfJp1vTGLABcnA==
root@webbrowser:~#
```

```
File Edit View Search Terminal Help
GNU nano 2.9.3 /etc/ipsec.secrets

# This file holds shared secrets or RSA private keys for authentication.

# RSA private key for this host, authenticating it to any other host
# which knows the public part.
192.168.1.142 192.168.1.138 : PSK "7to2SeybQuEuW5vkUICx5a0HuDxuQAz9nnccMZjAx0LIck4hHDgaRoRU5rQK6LA
Cl9VongobfJp1vTGLABcnA=="
```

Then under file /etc/ipsec.conf add below lines for establishing the route/connection where we have actually added information regarding source and destination IPs along with its gateway.

```
File Edit View Search Terminal Help
GNU nano 2.9.3 /etc/ipsec.conf Modified

# right=192.168.0.2
# rightsubnet=10.2.0.0/16
# rightid="C=CH, O=Linux strongSwan CN=peer name"
# auto=start
conn Client1-to-Webbrowser
  authby=secret
  left=%defaultroute
  leftid=192.168.1.138
  leftsubnet=192.168.1.1/24
  right=192.168.1.142
  rightsubnet=192.168.1.1/24
  lke=aes256-sha2_256-modp1024!
  esp=aes256-sha2_256!
  keyingtries=0
  ikelifetime=1h
  lifetime=8h
  dpddelay=30
  dpdtimeout=120
  dpdaction=restart
  auto=start
```

```
root@client1:~# ipsec restart
Stopping strongSwan IPsec...
Starting strongSwan 5.6.2 IPsec [starter]...
root@client1:~# ipsec status
Security Associations (0 up, 1 connecting):
Client1-to-Webbrowser[1]: CONNECTING, 192.168.1.138[%any]...192.168.1.142[%any]
root@client1:~#
```

ARP spoofing between client and web server

ARP spoofing between server and client

[illegible]

ARP Table at client end:

```
client@client: ~
File Edit View Search Terminal Help
client@client:~$ arp -a
web (192.168.1.142) at 00:0c:29:0a:32:47 [ether] on ens33
Server (192.168.1.252) at 00:0c:29:4c:96:b8 [ether] on ens33
█
```

DNS Spoof:

```
root@attacker: ~
File Edit View Search Terminal Tabs Help
root@attacker:~
root@attacker:~# nano hosts.txt
root@attacker:~# dnsspoof -i ens33 -f hosts.txt
dnsspoof: listening on ens33 [udp dst port 53 and not src 192.168.1.101]
192.168.1.137.40256 > 192.168.1.252.53: 41535+ PTR? 101.1.168.192.in-addr.arpa
192.168.1.137.54848 > 192.168.1.252.53: 3253+ PTR? 101.1.168.192.in-addr.arpa
█
```

5. ADDED FEATURES

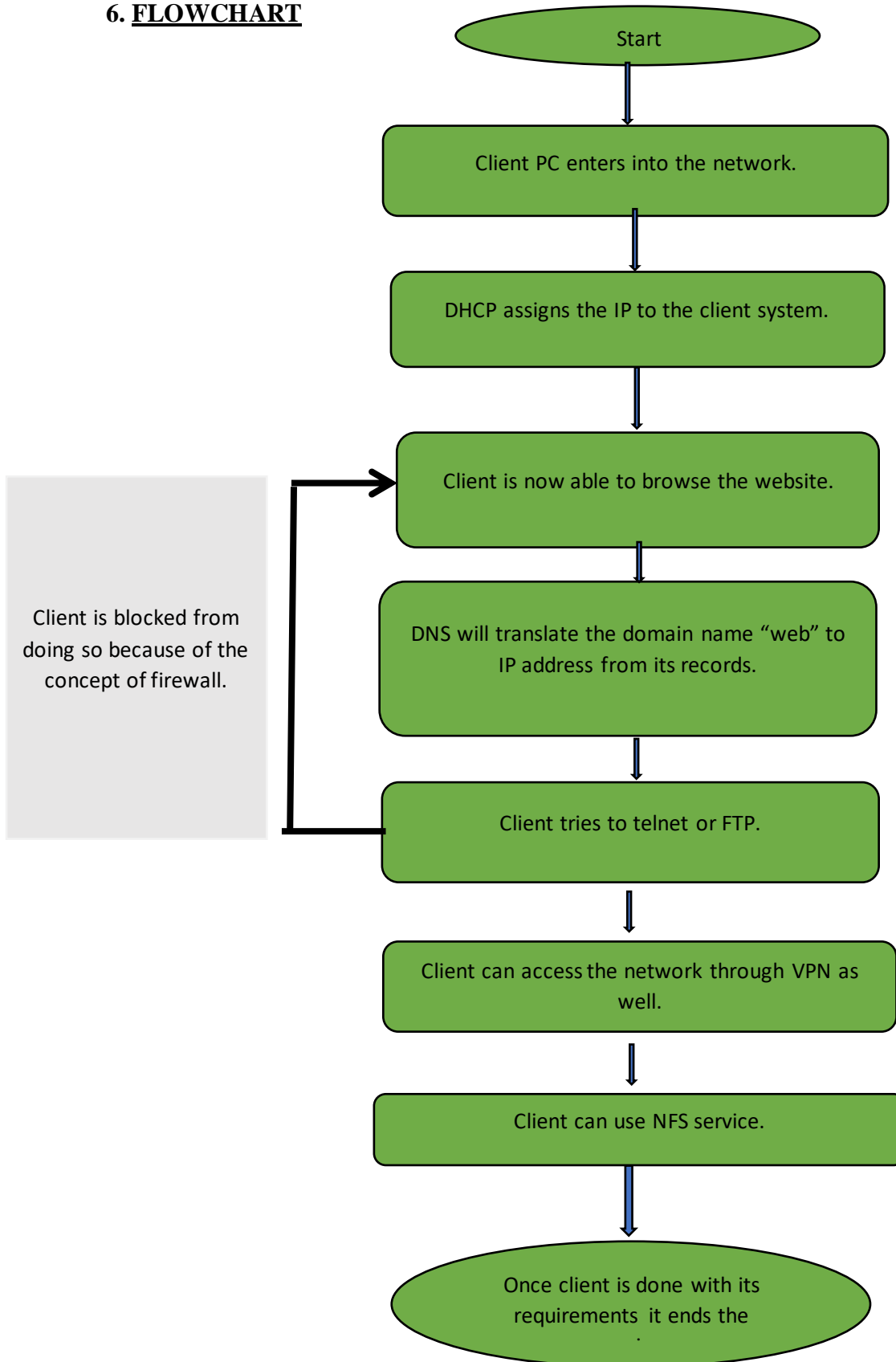
I. We have implemented NFS server, ARP Spoofing, IPSEC VPN Tunneling.

II. Implemented firewall rules for blocking FTP and telnet services.

III. SSH (Secure Shell): SSH takes place between the SSH server and SSH client where in the server shares its public key with the client so that the client can login into the server without the password.

IV. Shell Script: We have done backup and zipped the configuration files using scripting. We have written a script which creates a backup file and backup log and performs a rsync to the remote host.

6. FLOWCHART



7. ALGORITHM

Setup of DHCP:

Installed isc-dhcp-server packages

Configured the dhcpd.conf and dhcpd6.conf

Setup of DNS Master-Slave:

Installed BIND9 package at both machines.

Configured file /etc/bind/named.conf.local for creating zones at both master and slave machines.

Edited file for forward zone /etc/bind/for.DMS.com so that DNS will be able to resolve queries by translating domain name to IP address.

Edited file for reverse zone of both IPv4/IPv6 which are /etc/bind/rev.DMS.com and rev6.DMS.com so that it can process the reverse task of the forward zone file.

Setup of WebServer:

Installed the Apache2 Webserver. Edited the index.html file in /var/www/html

Setup of Firewall:

Used IP tables for setting access and deny rules over below ports:

ACCEPT: 80-HTTP, 22-SSH and ICMP

DENY: 20, 21, -FTP, 23-TELNET

Setup of Backup Server:

Installing openssh-server and client packages.

Writing a Bash script for scheduling the backup of file and sending to the DHCP server using ssh and rsync protocols.

8. TESTING

DHCP:

To check that the dhcp server is assigning IP address to the client, there are two methods.

At the client end:

```
client@client:~$ sudo dhclient -v ens33
Internet Systems Consortium DHCP Client 4.3.5
Copyright 2004-2016 Internet Systems Consortium.
All rights reserved.
For info, please visit https://www.isc.org/software/dhcp/

Listening on LPF/ens33/00:0c:29:2d:e0:3e
Sending on   LPF/ens33/00:0c:29:2d:e0:3e
Sending on   Socket/fallback
DHCPREQUEST of 192.168.1.137 on ens33 to 255.255.255.255 port 67 (xid=0x28d824e4)
DHCPACK of 192.168.1.137 from 192.168.1.130
RTNETLINK answers: File exists
bound to 192.168.1.137 -- renewal in 287 seconds.
client@client:~$
```

At the DHCP server end: `sudo less /var/lib/dhcp/dhcpd.leases`

```
# The format of this file is documented in the dhcpd.leases(5) manual page.
# This lease file was written by isc-dhcp-4.3.5

# authoring-byte-order entry is generated, DO NOT DELETE
authoring-byte-order little-endian;

lease 192.168.1.100 {
    starts 3 2020/04/15 18:52:10;
    ends 3 2020/04/15 19:02:10;
    tstp 3 2020/04/15 19:02:10;
    cltt 3 2020/04/15 18:52:10;
    binding state free;
    hardware ethernet 00:0c:29:ce:a2:9c;
}
lease 192.168.1.101 {
    starts 3 2020/04/15 23:43:43;
    ends 3 2020/04/15 23:53:43;
    cltt 3 2020/04/15 23:43:43;
    binding state active;
    next binding state free;
    rewind binding state free;
    hardware ethernet 00:0c:29:0a:32:47;
    client-hostname "attacker";
}
lease 192.168.1.101 {
    starts 3 2020/04/15 23:48:23;
    ends 3 2020/04/15 23:58:23;
    cltt 3 2020/04/15 23:48:23;
    binding state active;
    next binding state free;
    rewind binding state free;
    hardware ethernet 00:0c:29:0a:32:47;
    client-hostname "attacker";
}
lease 192.168.1.101 {
    starts 3 2020/04/15 23:52:51;
    ends 4 2020/04/16 00:02:51;
    cltt 3 2020/04/15 23:52:51;
    binding state active;
    next binding state free;
    rewind binding state free;
    hardware ethernet 00:0c:29:0a:32:47;
    client-hostname "attacker";
}
/var/lib/dhcp/dhcpd.leases
```


DNS:

- We have implemented concept of Master-Slave DNS by using two virtual machines. Our DHCP is assigning IPs to our Master-Slave DNS VMs.
- For the testing of the same we have used host command to see whether it is able to translate host name into IP address. Please find below screenshot for the same.

```
mkapadia@Server:/etc/bind$ host Client.DMS.com
Client.DMS.com has address 192.168.1.251
Client.DMS.com has IPv6 address 2001:1200:1100:1000::251
mkapadia@Server:/etc/bind$ host Server.DMS.com
Server.DMS.com has address 192.168.1.252
Server.DMS.com has IPv6 address 2001:1200:1100:1000::252
mkapadia@Server:/etc/bind$ host 192.168.1.251 192.168.1.252
Using domain server:
Name: 192.168.1.252
Address: 192.168.1.252#53
Aliases:

251.1.168.192.in-addr.arpa domain name pointer Client.DMS.com.
mkapadia@Server:/etc/bind$ host 2001:1200:1100:1000::251 2001:1200:1100:1000::252
Using domain server:
Name: 2001:1200:1100:1000::252
Address: 2001:1200:1100:1000::252#53
Aliases:

1.5.2.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.1.0.0.1.1.0.0.2.1.1.0.0.2.ip6.arpa domain name pointer Client.DMS.com.
mkapadia@Server:/etc/bind$ host Client.DMS.com 192.168.1.252
Using domain server:
Name: 192.168.1.252
Address: 192.168.1.252#53
Aliases:

Client.DMS.com has address 192.168.1.251
Client.DMS.com has IPv6 address 2001:1200:1100:1000::251
```

Also used dig and nslookup command for checking the correctness of the configuration of DNS and we are able to get detailed information of translation as below.

```
mkapadia@Server:/etc/bind$ dig -6 Client.DMS.com

; <<>> DiG 9.11.3-ubuntu1.11-Ubuntu <<>> -6 Client.DMS.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 59712
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 2, ADDITIONAL: 4

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: d067880b1a73b71846ffa4e45e97513912448a8d5d4eae2a (good)
;; QUESTION SECTION:
;Client.DMS.com.                IN      A

;; ANSWER SECTION:
Client.DMS.com.                86400   IN      A      192.168.1.251

;; AUTHORITY SECTION:
DMS.com.                       86400   IN      NS      Server.DMS.com.
DMS.com.                       86400   IN      NS      Client.DMS.com.

;; ADDITIONAL SECTION:
Client.DMS.com.                86400   IN      AAAA    2001:1200:1100:1000::251
Server.DMS.com.               86400   IN      AAAA    2001:1200:1100:1000::252
Server.DMS.com.               86400   IN      A      192.168.1.252

;; Query time: 0 msec
;; SERVER: ::1#53(::1)
;; WHEN: Wed Apr 15 11:23:53 PDT 2020
;; MSG SIZE rcvd: 194
```

```

mkapadia@Server:/etc/bind$ dig -x 192.168.1.251

; <<>> DiG 9.11.3-ubuntu1.11-Ubuntu <<>> -x 192.168.1.251
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 54878
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 65494
;; QUESTION SECTION:
;251.1.168.192.in-addr.arpa.      IN      PTR

;; ANSWER SECTION:
251.1.168.192.in-addr.arpa. 86400 IN      PTR      Client.DMS.com.

;; Query time: 0 msec
;; SERVER: 127.0.0.53#53(127.0.0.53)
;; WHEN: Wed Apr 15 11:25:13 PDT 2020
;; MSG SIZE rcvd: 83

mkapadia@Server:/etc/bind$

```

```

mkapadia@Server:/etc/bind$ nslookup Client.DMS.com 192.168.1.251
Server:      192.168.1.251
Address:     192.168.1.251#53

Name:   Client.DMS.com
Address: 192.168.1.251
Name:   Client.DMS.com
Address: 2001:1200:1100:1000::251

```

```

mkapadia@Server:/etc/bind$ ping Client.DMS.com
PING Client.DMS.com (Client.DMS.com (2001:1200:1100:1000::251)) 56 data bytes
64 bytes from Client.DMS.com (2001:1200:1100:1000::251): icmp_seq=1 ttl=64 time=0.357 ms
64 bytes from Client.DMS.com (2001:1200:1100:1000::251): icmp_seq=2 ttl=64 time=0.764 ms
64 bytes from Client.DMS.com (2001:1200:1100:1000::251): icmp_seq=3 ttl=64 time=0.668 ms
64 bytes from Client.DMS.com (2001:1200:1100:1000::251): icmp_seq=4 ttl=64 time=0.326 ms
64 bytes from Client.DMS.com (2001:1200:1100:1000::251): icmp_seq=5 ttl=64 time=0.718 ms
^Z
[6]+  Stopped                  ping Client.DMS.com
mkapadia@Server:/etc/bind$ ping4 Client.DMS.com
PING Client.DMS.com (192.168.1.251) 56(84) bytes of data.
64 bytes from Client.DMS.com (192.168.1.251): icmp_seq=1 ttl=64 time=0.636 ms
64 bytes from Client.DMS.com (192.168.1.251): icmp_seq=2 ttl=64 time=2.47 ms
64 bytes from Client.DMS.com (192.168.1.251): icmp_seq=3 ttl=64 time=0.717 ms
64 bytes from Client.DMS.com (192.168.1.251): icmp_seq=4 ttl=64 time=0.971 ms
^Z
[7]+  Stopped                  ping4 Client.DMS.com
mkapadia@Server:/etc/bind$

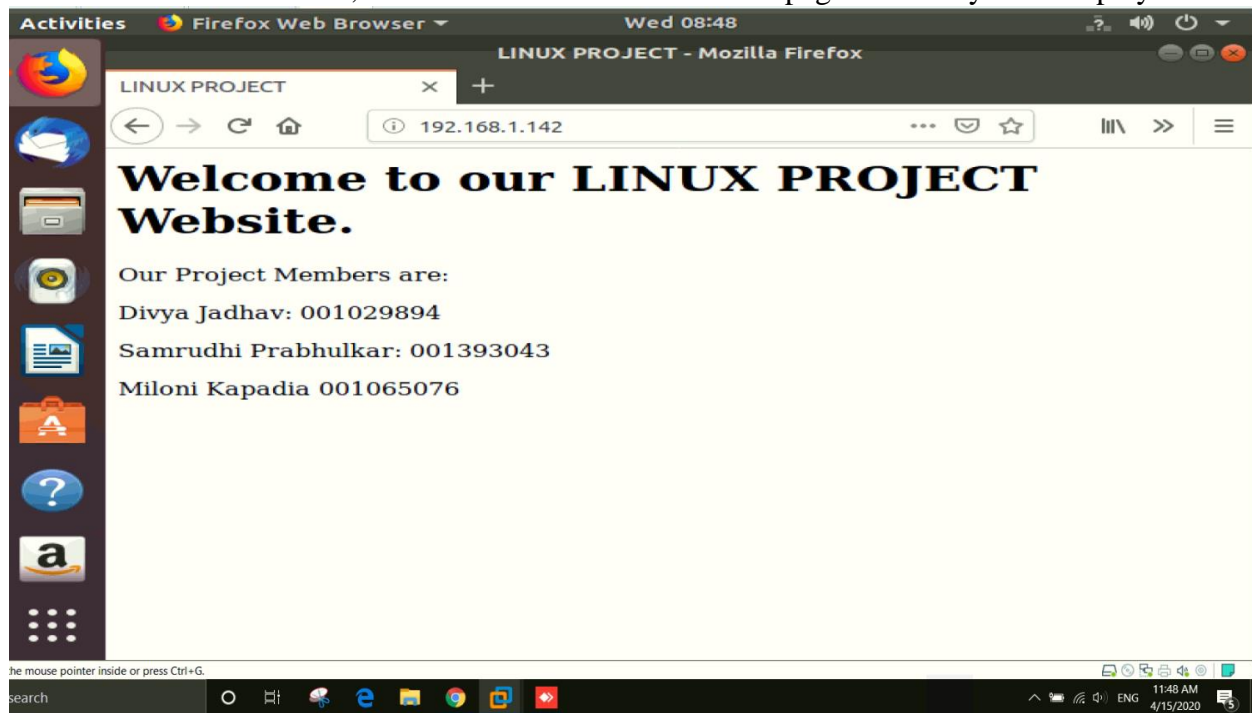
```

Also tried to fetch our website through our domain name, which we are able to get as our DNS is translating the domain name to its IP and our page is able to load.



Web Server:

Since the web server is allocated an IP address of 192.168.1.142, when the client enters that IP address in its web browser, the DNS resolves it and the web page created by us is displayed.



FIREWALL:

Implemented rules for accessing ssh,http and icmp (WITH echo replies)

Denied rules for FTP and Telnet:

```
webbrowser@webbrowser: ~  
File Edit View Search Terminal Help  
target prot opt source destination  
webbrowser@webbrowser:~$ sudo iptables -A INPUT -m conntrack --ctstate ESTABLISHED,RELATED -j ACCEPT  
webbrowser@webbrowser:~$ sudo iptables -A INPUT -p tcp --dport ssh -j ACCEPT  
webbrowser@webbrowser:~$ sudo iptables -A INPUT -p tcp --dport 80 -j ACCEPT  
webbrowser@webbrowser:~$ sudo iptables -A INPUT -p icmp --icmp-type 8 -j ACCEPT  
webbrowser@webbrowser:~$ sudo iptables -A INPUT -p icmp --icmp-type 0 -j ACCEPT  
webbrowser@webbrowser:~$ sudo iptables -A INPUT -p tcp --dport 20 -j DROP  
webbrowser@webbrowser:~$ sudo iptables -A INPUT -p tcp --dport 21 -j DROP  
webbrowser@webbrowser:~$ sudo iptables -A INPUT -d -p tcp -dport 23 -j DROP  
Bad argument 'tcp'  
Try 'iptables -h' or 'iptables --help' for more information.  
webbrowser@webbrowser:~$ sudo iptables -A INPUT -p tcp --dport 23 -j DROP  
webbrowser@webbrowser:~$ sudo iptables -L  
Chain INPUT (policy DROP)  
target prot opt source destination  
ufw-before-logging-input all -- anywhere anywhere  
ufw-before-input all -- anywhere anywhere  
ufw-after-input all -- anywhere anywhere  
ufw-after-logging-input all -- anywhere anywhere  
ufw-reject-input all -- anywhere anywhere  
ufw-track-input all -- anywhere anywhere  
ACCEPT all -- anywhere anywhere ctstate RELATED,ESTABLISHED  
ACCEPT tcp -- anywhere anywhere tcp dpt:ssh  
ACCEPT tcp -- anywhere anywhere tcp dpt:http  
ACCEPT icmp -- anywhere anywhere icmp echo-request  
ACCEPT icmp -- anywhere anywhere icmp echo-reply  
DROP tcp -- anywhere anywhere tcp dpt:ftp-data  
DROP tcp -- anywhere anywhere tcp dpt:ftp  
DROP tcp -- anywhere anywhere tcp dpt:telnet  
Chain FORWARD (policy DROP)  
target prot opt source destination  
ufw-before-logging-forward all -- anywhere anywhere  
ufw-before-forward all -- anywhere anywhere  
ufw-after-forward all -- anywhere anywhere  
ufw-after-logging-forward all -- anywhere anywhere
```

BACKUP SERVER:

By using rsync and ssh protocol in the bash script the backup file is zipped and send to the DHCP server every 9am in the morning with the current date Time stamp using Cronjobs.

```
dhcpserver@ubuntu:~$ cd backupdetails/  
dhcpserver@ubuntu:~/backupdetails$ ls  
apache2backup backup-2020-04-12.tar.gz backup-2020-04-14.tar.gz  
backup backup-2020-04-13.tar.gz backup-2020-04-15.tar.gz  
dhcpserver@ubuntu:~/backupdetails$
```

9. FUTURE IMPROVEMENTS

- We can add several components to make this network more robust, secure and scalable.
- We can add VLANs which would support an increasing number of PC's or clients entering in the network.
- We can add different firewall rules for allowing or denying the services on different ports.
- Use more networking technologies to ensure security within the network.

References:

<https://tools.ietf.org/html/rfc2131>