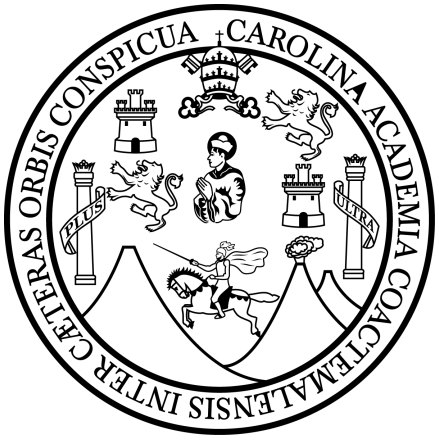
UNIVERSIDAD SAN CARLOS DE GUATEMALA

CENTRO UNIVERSITARIO DE OCCIDENTE

DIVISIÓN CIENCIAS DE LA INGENIERÍA

CARRERA DE INGENIERÍA CIENCIAS Y SISTEMAS



LABORATORIO DE REDES DE COMPUTADORAS 1

"**SÉPTIMO SEMESTRE"**

ING.: FRANCISCO ROJAS

ESTUDIANTES: LUIS ESTUARDO BOLAÑOS GONZÁLEZ - 201731766
MARIO MOISES RAMIREZ TOBAR - 201830007
YEFER RODRIGO MIGUEL ALVARADO TZUL - 201731163
FATIMA ODRA DANIELA TEZO SUM - 201831039

TRABAJO: GESTION DE PROYECTOS (PROYECTO FINAL)

FECHA: 11 de mayo de 2,021

DESCRIPCIÓN DEL PROYECTO

Se solicita una arquitectura de red con los siguientes componentes:

- 1 Servidor VPN.
- 1 Enrutador con Gateway hacia internet.
- 1 cliente VPN con Gateway hacia internet.

El enrutador tiene a su merced a la red A, red B, red C, red D, cada una de estas con un número determinado de máquinas, la red D vendría a ser la red Administrativa del sistema.

El cliente VPN tiene a su merced a la red U la cual tiene una capacidad de 5 maquinas.

La comunicación entre estas redes se resume en las siguientes posibilidades:

- Red A hacia B si.
- Red B hacia A si.
- Red A hacia C si.
- Red C hacia A si.
- Red U hacia B si.
- Red B hacia U no.
- Red D (Administración) hacia A, B, C, U si, en viceversa no.
- Cualquier otra modalidad de comunicación está negada.

El fin del proyecto es emular la comunicación desde una red central (enrutador y sus redes) y una red sucursal (Cliente y su propia red denominada U) esto mediante el servidor VPN, así se logrará una comunicación a larga distancia entre redes emulando una red local.

El servidor VPN debe ir conectado hacia el cliente y hacia el enrutador, con esto se logra una comunicación tanto desde las máquinas del Cliente (la red U) hacia las máquinas de las redes a merced del enrutador.

REQUERIMIENTOS TECNICOS

- Uso del sistema operativo Centos en modo texto virtualizado dentro de KVM.
- Creación de servidor VPN mediante OpenVPN.
- Conexión exitosa entre el servidor VPN, el cliente y el enrutador.
- Lograr una comunicación de esquina a esquina desde alguna de las redes dentro del router hacia un cliente y viceversa.
- Llevar un control en los permisos de comunicación entre redes.
- Creación de puentes transparentes para las comunicaciones entre redes.

HERRAMIENTAS

- Sistema operativo Centos 7 en modo texto.
- OpenVPN version 2.4.10.
- Easy-rsa version 3.0.8.
- Paquete net - tools.
- Interfaz nmtui.
- Editor de texto nano.
- Lenguaje Bash.

OBJETIVOS

Objetivos Generales:

- Aplicar los conocimientos sobre redes, conexiones VPN, enrutadores, así como el manejo de las direcciones IP.
- Implementar una red capaz de lograr la comunicación entre sus diferentes nodos.

Objetivos Específicos:

- Lograr un buen manejo en los permisos de comunicación entre los diferentes nodos de la red.
- Crear un servidor VPN capaz de aceptar el traspaso de datos entre dos localizaciones distintas.

MARCO TEÓRICO

Router

Un **router**, **enrutador**, (del inglés ***router***) o **encaminador**, es un dispositivo que permite interconectar computadoras que funcionan en el marco de una red. Su función es la de establecer la ruta que destinará a cada paquete de datos dentro de una red informática.

El funcionamiento básico de un enrutador o encaminador, como se deduce de su nombre, consiste en enviar los paquetes de red por el camino o ruta más adecuada en cada momento. Para ello almacena los paquetes recibidos y procesa la información de origen y destino que poseen. Con arreglo a esta información reenvía los paquetes a otro encaminador o bien al anfitrión final, en una actividad que se denomina 'encaminamiento'. Cada encaminador se encarga de decidir el siguiente salto en función de su tabla de reenvío o tabla de encaminamiento, la cual se genera mediante protocolos que deciden cuál es el camino más adecuado o corto, como protocolos basado en el algoritmo de Dijkstra.

Por ser los elementos que forman la capa de red, tienen que encargarse de cumplir las dos tareas principales asignadas a la misma:

- ***Reenvío de paquetes*:** cuando un paquete llega al enlace de entrada de un encaminador, este tiene que pasar el paquete al enlace de salida apropiado. Una característica importante de los encaminadores es que no difunden tráfico difusivo.

- ***Encaminamiento de paquetes*** **:** mediante el uso de algoritmos de encaminamiento tiene que ser capaz de determinar la ruta que deben seguir los paquetes a medida que fluyen de un emisor a un receptor.

Por tanto, debemos distinguir entre reenvío y encaminamiento. Reenvío consiste en coger un paquete en la entrada y enviarlo por la salida que indica la tabla, mientras que por encaminamiento se entiende el proceso de hacer esa tabla.

VPN

La VPN, acrónimo de “Virtual Private Network” o “Red Privada Virtual” en español, asegura que su computadora ya no se pueda rastrear, y funciona de la siguiente manera: la VPN, en lugar de conectarse a su proveedor de Internet, conecta su computadora a un servidor VPN haciendo uso de una conexión segura y encriptada. A continuación, el servidor VPN se contacta con el sitio web que usted está buscando. Aquí, los detalles sobre su visita se almacenan como siempre, pero en función de la dirección IP del servidor VPN, en lugar de la suya.

Servidor VPN

Un servidor VPN es un servidor físico o virtual que está configurado para alojar y entregar servicios VPN a usuarios de todo el mundo. El servidor es una combinación de hardware VPN y software VPN que permite a los clientes VPN conectarse a una red privada segura. A diferencia de la mayoría de los servidores, un servidor VPN generalmente tiene más puertos de comunicaciones lógicos y físicos.

Todo el proceso comienza con usted ejecutando un cliente VPN. Lo conecta con el servidor VPN y comienza a enviar su tráfico a través de su ISP. Sin embargo, esta vez todos los datos están encriptados por los protocolos VPN con los que está configurado el servidor, lo que significa que su ISP (o cualquier otra persona) no puede monitorearlos.

Una vez que el servidor VPN reciba toda la información encriptada, procederá a descifrarla y la enviará al servidor web designado. Posteriormente, el servidor VPN encriptará los datos que recibe de dicho servidor web y se los enviará a través de su ISP. Una vez que reciba esos datos en su dispositivo, el cliente VPN los descifrara por usted.

Para tener una mejor idea de cómo funciona todo esto, imagine que hay un "túnel" establecido entre el cliente VPN y el servidor VPN. Cualquier información que pase por ese túnel está encriptada y, por lo tanto, es ilegible para cualquiera que esté fuera del túnel.

Cliente VPN

El VPN Cliente se utiliza para buscar el acceso a través del VPN Gateway y establecer la conexión, construyendo un túnel seguro para traer los datos de los usuarios y corporaciones.

OpenVPN

OpenVPN es una herramienta de conectividad basada en software libre: SSL (Secure Sockets Layer), VPN Virtual Private Network (red virtual privada). OpenVPN ofrece conectividad punto-a-punto con validación jerárquica de usuarios y host conectados remotamente. Resulta una

muy buena opción en tecnologías Wi-Fi (redes inalámbricas IEEE 802.11) y soporta una amplia configuración, entre ellas balanceo de cargas. Está publicado bajo la licencia GPL, de software libre.

CREACIÓN DEL SERVIDOR VPN Y CONEXIÓN CON EL CLIENTE

Para la configuración del servidor VPN se deben configurar los siguientes comandos dentro del servidor CentOS 7.

Instalar el repositorio epel-release.

```
[root@server ~]# dnf install -y epel-release
```

Se instala la aplicación:

```
[root@server ~]# dnf install -y openvpn easy-rsa
```

Se accede al directorio de easy-rsa, donde se ubican los archivos necesarios para la configuración:

```
[root@server ~]# cd /usr/share/easy-rsa/3.0.8
```

Si se desea, se puede copiar el contenido al directorio de openvpn: (en mi caso lo haré, por comodidad de configuración)

```
[root@server ~]# cp -r * /etc/openvpn/server/.
```

Cambiamos al directorio dónde se copiaron los archivos:

```
[root@server ~]# cd /etc/openvpn/server/
```

Se inicia el PKI (Public Key Infrastructure) y se construye la autoridad de certificación (CA):

```
[root@server server]# ./easyrsa init-pki
[root@server server]# ./easyrsa build-ca nopass
```

Se genera y firma el certificado del servidor:

```
[root@server server]# ./easyrsa gen-req serverA nopass
[root@server server]# ./easyrsa sign-req server serverA nopass
```

Se genera y firma el certificado para el cliente:

```
[root@server server]# ./easyrsa gen-req clientA nopass
[root@server server]# ./easyrsa sign-req client clientA nopass
```

se genera el dh.pem (parámetros de Diffie-Hellman) que establece la fortaleza en el intercambio de las claves:

```
[root@server server]# ./easyrsa gen-dh
```

Copiar el archivo de configuración ejemplo al directorio de configuración:

```
[root@server openvpn]# cp
/usr/share/doc/openvpn/sample/sample-config-files/server.conf /etc/openvpn/server/.
```

Editamos el archivo '/etc/openvpn/server.conf':

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

Se edita el archivo /etc/openvpn/server.conf y se configuran las rutas correctas de los certificados y llaves, para este ejemplo, quedaría como se muestra a continuación.

```
# Any X509 key management system can be used.
# OpenVPN can also use a PKCS #12 formatted key file
# (see "pkcs12" directive in man page).

ca /etc/openvpn/server/pki/ca.crt
cert /etc/openvpn/server/pki/issued/serverA.crt
key /etc/openvpn/server/pki/private/serverA.key  # This file should be kept secret

# Diffie hellman parameters.
# Generate your own with:
#   openssl dhparam -out dh2048.pem 2048

dh /etc/openvpn/server/pki/dh.pem
.
topology subnet
.
.
push "redirect-gateway def1 bypass-dhcp"
.
.
push "dhcp-option DNS 8.8.8.8"
push "dhcp-option DNS 8.8.4.4"
.
.
# The second parameter should be '0'
# on the server and '1' on the clients.
;tls-auth ta.key 0 # This file is secret
.
.
# You can uncomment this out on
# non-Windows systems.
user nobody
group nobody
```

Si alguno de los parámetros resaltados no se configura, el servicio no podrá ser iniciado.

```
```

Se habilita el ip forwarding.

```
[root@server ~]# sysctl -w net.ipv4.ip_forward=1
```

Para hacerlo de forma persistente:
//verifican como se llama el .conf

```
[root@server ~]# echo net.ipv4.ip_forward=1 >> /etc/sysctl.d/sysctl-additionals.conf
```

Establecer las reglas del firewall:

```
[root@server ~]# firewall-cmd --zone=trusted --add-masquerade --permanent
[root@server ~]# firewall-cmd --set-default-zone=trusted
[root@server ~]# firewall-cmd --add-service=openvpn --permanent
[root@server ~]# firewall-cmd --reload
[root@server ~]# firewall-cmd --list-all
```

Deshabilitar SElinux:

```
[root@server ~]# setenforce 0
```

Se comprueba su funcionamiento con:

```
[root@server ~]# cd /etc/openvpn/server/
[root@server server]# openvpn --config server.conf
```

La siguiente línea al final de la salida, indica que levantó correctamente:

```
.... Initialization Sequence Complete
```

Se inicia y habilita el servidor:

```
[root@server ~]# systemctl enable openvpn-server@server  --now
[root@server ~]# systemctl status openvpn-server@server
```

Se puede comprobar que la interfaz virtual se haya habilitado con 'ip addr':

```
[root@server ~]# ip addr
.
.
5: tun0: <POINTOPOINT,MULTICAST,NOARP,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UNKNOWN group
default qlen 100
    link/none
    inet 10.8.0.1/24 brd 10.8.0.255 scope global tun0
       valid_lft forever preferred_lft forever
```

```
    inet6 fe80::dab7:46e8:cc6e:4b79/64 scope link flags 800
       valid_lft forever preferred_lft forever
```

Configuración del cliente

Instalar el repositorio epel-release.

```
[root@client ~]# dnf -y install epel-release
```

Se instala la aplicación:

```
[root@client ~]# dnf -y install openvpn
```

Se comprueba la conexión entre los equipos (servidor y cliente), y se copian los certificados del generados en el servidor hacia el cliente: (no olvides sustituir ***ip_client*** por dirección ip de tu cliente):

```
[root@server ~]# cd /etc/openvpn/server/pki/
[root@server pki]# scp ca.crt root@ip_client:/etc/openvpn/client/.
[root@server pki]# scp issued/clientA.crt root@ip_client:/etc/openvpn/client/.
[root@server pki]# scp private/clientA.key root@ip_client:/etc/openvpn/client/.
```

Se edita el archivo de configuración para el **cliente**:

```
[root@client ~]# vi /etc/openvpn/client/clientA.conf
```

Y se agregan las siguientes líneas (no olvides sustituir ***ip_server*** por la dirección ip de tu servidor):

```
client
dev tun
proto udp
remote ip_server 1194
ca ca.crt          #en caso de usar otra ruta, modificarla
cert clientA.crt   #en caso de usar otra ruta, modificarla
key clientA.key    #en caso de usar otra ruta, modificarla
verb 5

remote-cert-tls server
auth-nocache
cipher AES-256-CBC
```

Se guarda el archivo: y ejecuta la configuración para comprobar el funcionamiento::

```
[root@client ~]# cd /etc/openvpn/client/
```

```
[root@client openvpn]# openvpn --config clientA.conf
```

La siguiente línea al final de la salida indica que la VPN levantó correctamente, si no se despliega esa línea es necesario revisar la configuración:

```
.... Initialization Sequence Complete
```

Si la VPN levantó correctamente, se puede habilitar el servicio de la siguiente manera:

```
[root@client ~]# systemctl start openvpn-client@clientA
```

O bien, para iniciarla y habilitarla para que inicie con el arranque del sistema:

```
[root@client ~]# systemctl enable openvpn-client@clientA  --now
```

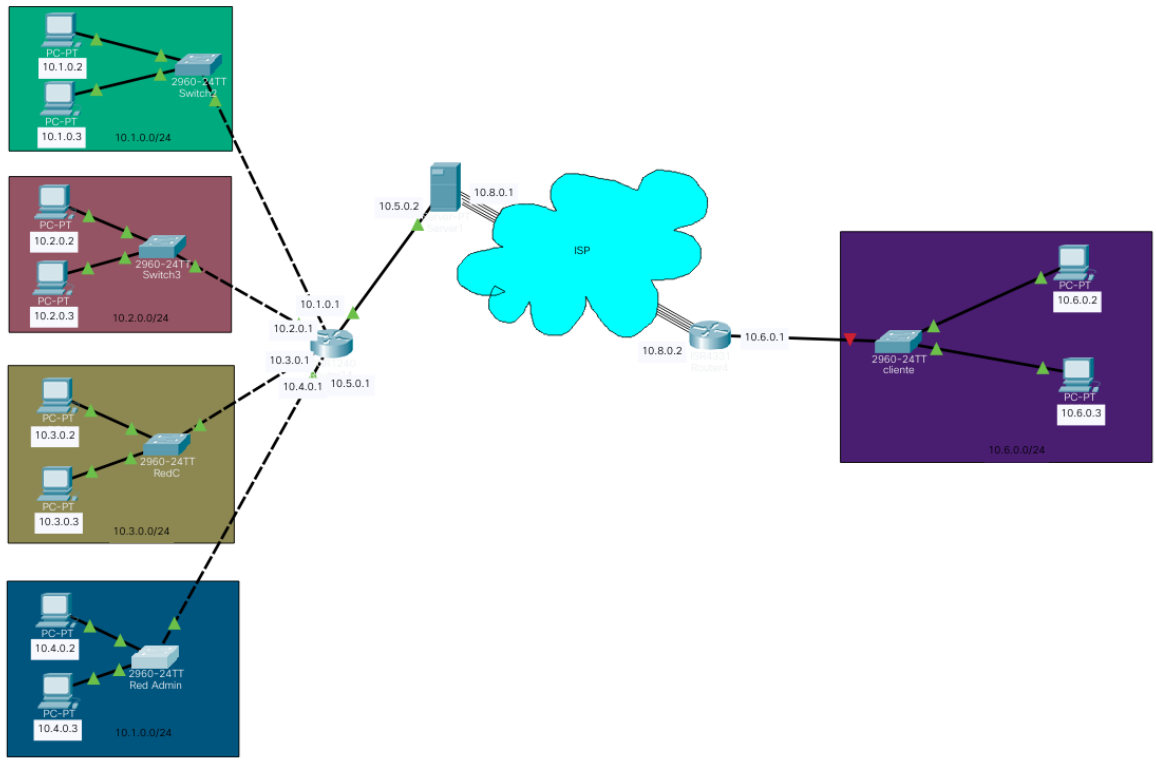
Al igual que con el servidor, se debió crear una interfaz virtual:

```
[root@server ~]# ip addr
.
.
5: tun0: <POINTOPOINT,MULTICAST,NOARP,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UNKNOWN group
default qlen 100
    link/none
    inet 10.8.0.2/24 brd 10.8.0.255 scope global tun0
       valid_lft forever preferred_lft forever
    inet6 fe80::a2fc:5174:af3f:5c3e/64 scope link flags 800
       valid_lft forever preferred_lft forever
```

Con lo anterior, la VPN está establecida y funcional.

MANUAL PARA ORGANIZAR LAS DIRECCIONES IP

Arquitectura del sistema



Pequeña descripción de IP's y Mascaras de red

- Host se refiere a un equipo de cómputo 'final' es decir, una máquina a la que se está conectado dentro de una red.
- Las máscaras de subred sólo pueden tener los siguientes valores debido a que sólo hacen referencia a la cantidad de bits que se usan de red y cuantos para host.
    - 255.0.0.0 -> para una red como 192.0.0.0 en la que pueden haber host tales como 192.12.33.2 o 192.44.22.4 donde sólo la primera 'sección' hace referencia a la red y las otras tres al 'número' de host.

- 255.255.0.0 -> para una red como 192.159.0.0 en la que pueden haber host tales como 192.159.23.1 o 192.159.13.5, donde las primeras dos secciones son de red y las otras dos de host.
- 255.255.255.0 -> para una red como 192.168.120.0 en la que pueden haber host tales como 192.168.120.4 o 192.168.120.24, donde las primeras tres secciones son de red y las otra de host.
- Se puede usar la notación /#bits para hacer referencia a la máscara de red
  - <ip>/8 para una máscara del tipo 255.0.0.0
  - <ip>/16 para una máscara del tipo 255.255.0.0
  - <ip>/24 para una máscara del tipo 255.255.255.0

Nota: Es importante tener en cuenta que para que dos host puedan tener comunicación entres sí, más allá de la conexión física, deben coexistir en la misma red y por lo tanto, tener los mismos bits de red y la misma máscara de subred.

Agregar una interfaz de red a nuestra máquina virtual

1. Teniendo nuestra máquina virtual procederemos a 'abrir' sus configuraciones, al ejecutar observamos en la parte superior izquierda un símbolo de 'información', entraremos ahí.
2. Hasta abajo a la izquierda veremos una opción 'agregar hardware'
3. Buscaremos en las opciones 'Red' y aceptaremos la configuración que viene por defecto.
4. Repetiremos los pasos del 1 al 3 para cada interfaz de red extra que necesitemos cada máquina tiene por defecto 1 interfaz de red.

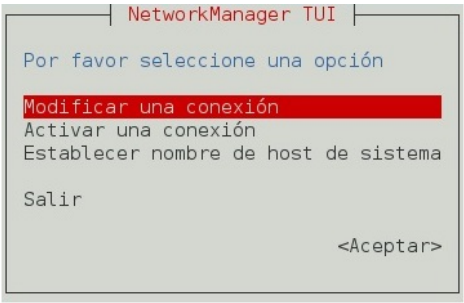
Cambiar una dirección Ip

Utilizando comando nmtui

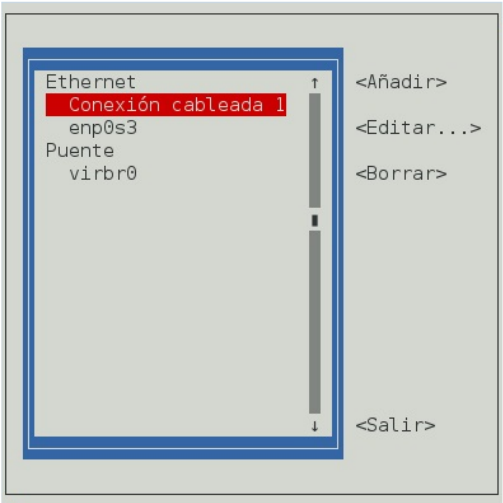
Utilizamos el comando nmtui para acceder a las configuraciones de nuestra red

```
[root@enrutador ~] nmtui
```

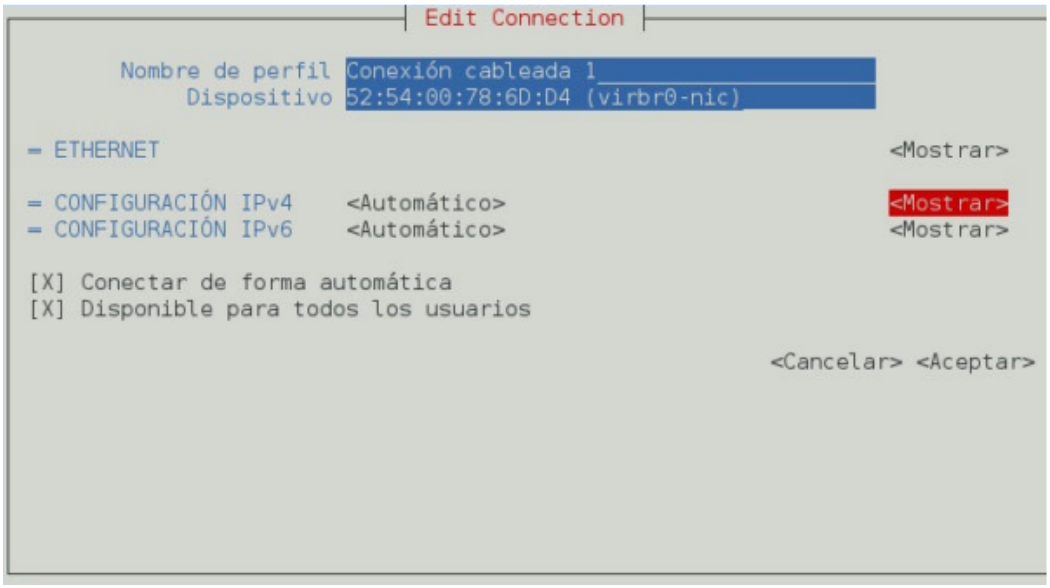
1. Se nos abrirá una ‘ventana’ en la que podremos acceder a las configuraciones para cada red, en este caso particular procederemos en <**Modificar una conexión>,** en el cual podremos ver un listado de todas nuestras interfaces de red.



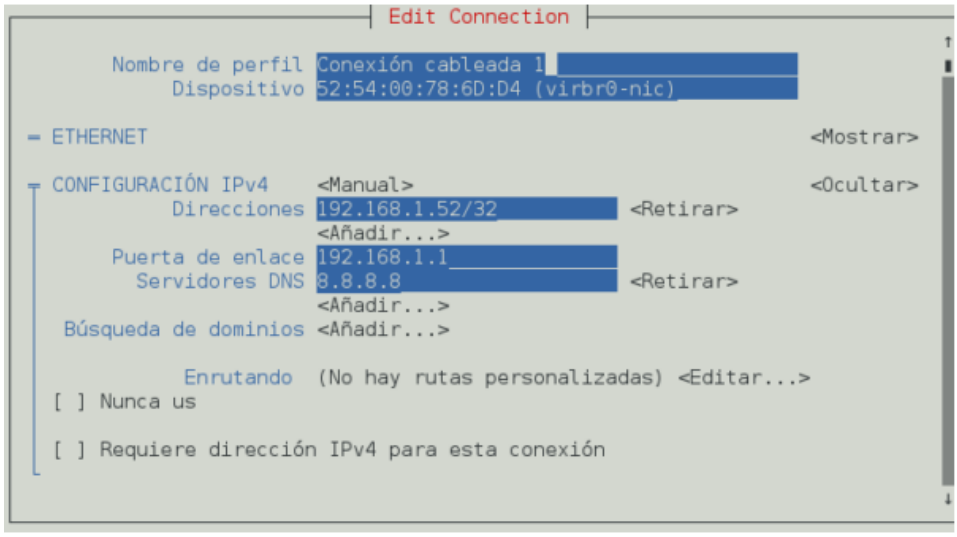
2. Aquí buscaremos la interfaz de red que deseamos modificar, la seleccionaremos y daremos enter.



3. Aquí podremos cambiar el nombre de la ‘conexión’ en este caso lo vemos como conexión cableada 1, es conveniente configurar este valor para poder diferenciar fácilmente a qué nos referimos.



4. Iremos hacia la sección de configuración IPv4, daremos enter en ‘automático’ y lo cambiaremos a manual, después de eso, daremos enter en mostrar.
    a. Configuración IPv6 podemos cambiarlo a ignorar si se desea.



    b. Colocaremos la dirección ip que corresponda + ‘/’ + <Numero de bits de la máscara de subred (24 generalmente)>
    c. Agregamos la puerta de enlace (si hace falta)
    d. Y el servidor DNS 8.8.8.8

e.

5. Aceptamos
6. En el submenú de activar una conexión, entramos, activamos y desactivamos cada conexión que hayamos modificado con anterioridad y aceptamos.
7. Aceptamos y salimos de nmtui
8. Ejecutamos el comando '**systemctl restart network.service**'
9. Verificamos que los cambios se realizaron correctamente con '**ip addr**'

Mediante archivos de configuración

Accedemos a la ubicación /etc/sysconfig/network-scripts/

Configuración de Router

Agregar hardware y configuración de Ip's para las interfaces

Será necesario agregar 5 interfaces de red para administrar las diferentes conexiones tal y como se puede observar en la arquitectura del sistema, el router estará organizado con direcciones ip estáticas, de la siguiente forma:

- Interfaz 1: Irá conectada al servidor, por lo que se asignará la red 10.0.5.0/24
    - IP = 10.5.0.1
- Interfaz 2: Irá conectada a la red A, por lo que se asignará la red 10.0.1.0/24
    - IP = 10.1.0.1
- Interfaz 3: Irá conectada a la red B, por lo que se asignará la red 10.0.2.0/24
    - IP = 10.2.0.1
- Interfaz 4: Irá conectada a la red C, por lo que se asignará la red 10.0.3.0/24
    - IP = 10.3.0.1
- Interfaz 5: Irá conectada a la red D, por lo que se asignará la red 10.0.4.0/24

- IP = 10.4.0.1

Agregando los servicios necesarios para un correcto funcionamiento

```
[root@enrutador ]: firewall-cmd --add-service=dhcp --permanent
[root@enrutador ]: firewall-cmd --add-masquerade --permanent
[root@enrutador ]: firewall-cmd --reload
```

Al agregar estos servicios garantizamos que las interfaces dentro nuestro router envíen los paquetes correctamente entre ellas y garantizando la comunicación con cada equipo de la subred a la que pertenezcan.

Configuración de sub-redes

Red A

Accederemos a cada equipo y cambiaremos su ip por una que forme parte de la red 10.1.0.0/24, sin usar la ip 10.1.0.1/24 que ya le fue asignada a la interfaz encargada de recibir los paquetes de esta red en el router.

Red B

Accederemos a cada equipo y cambiaremos su ip por una que forme parte de la red 10.2.0.0/24, sin usar la ip 10.2.0.1/24 que ya le fue asignada a la interfaz encargada de recibir los paquetes de esta red en el router.

Red C

Accederemos a cada equipo y cambiaremos su ip por una que forme parte de la red 10.3.0.0/24, sin usar la ip 10.3.0.1/24 que ya le fue asignada a la interfaz encargada de recibir los paquetes de esta red en el router.

Red D

Accederemos a cada equipo y cambiaremos su ip por una que forme parte de la red 10.4.0.1/24, sin usar la ip 10.4.0.1/24 que ya le fue asignada a la interfaz encargada de recibir los paquetes de esta red en el router.

Red U

Accederemos a cada equipo y cambiaremos su ip por una que forme parte de la red 10.6.0.0/24, sin usar la ip 10.6.0.1/24 que ya le fue asignada a la interfaz encargada de recibir los paquetes de esta red en el cliente-vpn.

Configuración servidorVpn - router

Agregaremos una interfaz de red a la máquina virtual que ejecutará el servidor vpn, esta se conectará al router mediante la red 10.5.0.0/24, sabiendo que, la interfaz del router tiene asignada la dirección 10.5.0.1/24, asignaremos al servidor la dirección ip 10.5.0.2/24

Configuración clienteVpn - Red u

Agregaremos una interfaz de red a la máquina virtual que ejecutará el cliente vpn, esta se conectará a la red u mediante la red 10.6.0.0/24, para ello asignaremos al cliente la dirección ip 10.6.0.1/24

CONCLUSIONES

Las redes de datos tienen como función primaria y de mucha importancia la facilitación de la comunicación ya que las mismas permiten conectarnos de forma global con nuestra familia, amigos etc. Esto por medio de procedimientos que son distintos entre sí, utilizando las redes, haciendo que la comunicación llegue al destino y a tiempo.

Con la llegada de la tecnología se comienza el proceso de implementarla en las organizaciones laborales, profesionales y personales ya que la tecnología nos da las herramientas necesarias para la realización de los diferentes procesos y actividades que requerimos en nuestro diario vivir.

Gracias a la correcta implementación de las redes y cada herramienta que se utiliza nos aporta una comunicación constante donde el costo nos es favorable, también debemos de realizar distintos procedimientos los cuales nos permitan la seguridad de los datos que se manejan entre las redes y además de la configuración adecuada de una conexión remota que permita la comunicación a grandes distancias.

BIBLIOGRAFÍA

A., D. (2020, 27 octubre). *Configurar un servidor VPN de Linux con OpenVPN - Guía paso a paso*. Tutoriales Hostinger. https://www.hostinger.es/tutoriales/como-configurar-vpn-linux-con-openvpn

Almaliki, Z. A. (2020, 1 octubre). *Take care of your privacy and create your own virtual private network*. Medium. https://towardsdatascience.com/new-story-a6420a1f097

Asghar, J. J. (2020, 14 febrero). *CentOS 8 as my new router*. jjasghar rants and ramblings. https://jjasghar.github.io/blog/2020/02/14/centos-8-as-my-new-router/

Carles, J. (2019, 30 marzo). *Crear y configurar servidor openvpn*. geekland. https://geekland.eu/crear-y-configurar-servidor-openvpn/

Carles, J. (2021, 1 enero). *Encontrar servidor con DNS dinamico*. geekland. https://geekland.eu/encontrar-servidor-con-dns-dinamico/

colaboradores de Wikipedia. (2021, 24 abril). *OpenVPN*. Wikipedia, la enciclopedia libre. https://es.wikipedia.org/wiki/OpenVPN

Córdoba, D. (2018, 8 noviembre). *OpenVPN + EasyRSA-3: Montando la VPN*. Junco TIC. https://juncotic.com/openvpn-easyrsa-3-montando-la-vpn/

Digicert. (s. f.). *SSL Certificate Country Codes - Create CSR | DigiCert.com*. https://www.digicert.com/kb/ssl-certificate-country-codes.htm

Escarcha, A. (2012, 1 diciembre). *[Solucionado] linux | Cómo eliminar los servicios del*. EnMiMaquinaFunciona.com. https://www.enmimaquinafunciona.com/pregunta/120820/-como-eliminar-los-servicios-del-sistema-

Fernández, L. (2020, 22 febrero). *¿Nuevo en Linux? Revisa esta lista de comandos básicos de redes*. RedesZone. https://www.redeszone.net/tutoriales/redes-cable/comandos-basicos-redes-linux/

G.B., S. (2017, 19 diciembre). *Como instalar GIT en CentOS 7*. Linux para todos. https://www.sololinux.es/instalar-git-centos-7/

Gite, V. (2021, 18 abril). *CentOS 8 Set Up OpenVPN Server In 5 Minutes*. NixCraft. https://www.cyberciti.biz/faq/centos-8-set-up-openvpn-server-in-5-minutes/#comments

Indiana University. (2018, 14 mayo). *About fully qualified domain names (FQDNs)*. The Trustees of Indiana University. https://kb.iu.edu/d/aiuv

Jiménez, J. (2020, 17 julio). *Cómo comprobar qué puertos tenemos abiertos en Linux*. RedesZone. https://www.redeszone.net/2018/07/10/comprobar-puertos-abiertos-linux/

Kiarie, J. (2021, 18 febrero). *How to Install and Configure OpenVPN Server in CentOS 8/7*. TecMint. https://www.tecmint.com/install-openvpn-in-centos/

Linksys. (s. f.). *Linksys Official Support - Learning about Ping test*. https://www.linksys.com/cz/support-article?articleNum=135197

Linux Mint Forums. (2016, 14 febrero). *No me funciona la red cableada (SOLUCIONADO) - Linux Mint Forums*. https://forums.linuxmint.com/viewtopic.php?t=216693

Linux Mint Forums. (2019, 9 octubre). *Conexión cableada en LM 19.2 [Solucionado] - Linux Mint Forums*. https://forums.linuxmint.com/viewtopic.php?t=303281

O. (2020, 17 julio). *OpenVPN/openvpn*. GitHub. https://github.com/OpenVPN/openvpn/blob/master/sample/sample-config-files/client.conf

OpenVpn. (2021, 29 abril). *How To Guide: Set Up & Configure client/server VPN*. https://openvpn.net/community-resources/how-to/

Proxmox. (s. f.). *Download Proxmox software, documentation, agreements*. https://www.proxmox.com/en/downloads

Proyectoa.com. (2018, 6 junio). *Comando nslookup en Linux CentOS 7 - Proyecto A*. Proyecto A - Tutoriales, código fuente, open source sobre bases de datos, sistemas operativos, lenguajes de programación, virtualización, Linux, Windows, Android, Visual Studio .NET, C#, Delphi, Oracle, SQL Server, MySQL, Java. https://proyectoa.com/foros/tema/comando-nslookup-en-linux-centos-7/

S. (2018, 23 septiembre). *Linux: Como saber el gateway (puerta de enlace)*. SYSADMIT. https://www.sysadmit.com/2018/09/linux-como-saber-gateway-puerta-de-enlace.html

Simic, S. (2021a, abril 20). *How to Install OpenVPN on CentOS 7 or 8*. Knowledge Base by PhoenixNAP. https://phoenixnap.com/kb/openvpn-centos

StackOverflow. (2014, 14 octubre). *Copying files with scp: connection timed out*. https://stackoverflow.com/questions/26364318/copying-files-with-scp-connection-timed-out

Tagliaferri, L. (2020, 29 abril). *Cómo instalar Git en CentOS 8*. DigitalOcean. https://www.digitalocean.com/community/tutorials/how-to-install-git-on-centos-8-es

TrescomaTres. (2017, 5 diciembre). *CentOS – Abrir y cerrar puertos*. https://blog.trescomatres.com/2017/12/centos-abrir-y-cerrar-puertos/

VPN-server (Centos 8). (2020, 15 octubre). Google Docs. https://docs.google.com/document/d/18CuuuK3V7r1JSpY3zyuYuhjhEsdGpr6YqswwsM6ULz8/edit#

Yohan, J. (2021, 29 abril). *Reference manual for 2.4*. OpenVPN. https://openvpn.net/community-resources/reference-manual-for-openvpn-2-4/