Proyecto #2 - Creaci﷽n de servicio VPN, Servidor web(VHost) y servidor SQL

Debido a la pandemia de Covid-19, la empresa Noire S.A. los ha contratado a usted ysu compañero, para implementar una solución de VPN para habilitar a sus colaboradores la opción de teletrabajo. Para demostrar su solución deberá considerar al menos los seisdispositivos que se muestran en el siguiente diagrama.La empresa Noire S.A. requiere desplegar toda su infraestructura en la nube de Azure, para lo cual usted deberá aprovisionar y configurar una servidor de OpenVPN, que será el único que tendrá un puerto expuesto a Internet (el puerto 1194, en TCP/UDP), en otras dossub-redes de Azure, usted deberá desplegar una servidor web con Apache2 y un servidor deMySQL.

Los trabajadores podrán acceder a dichos recursos desde clientes en Windows, GNU/Linux y Android, el nivel de acceso dependerá del rol del trabajador.

El personal de tecnologías podrá acceder tanto al servidor web, como al servidor de base de datos, mientras que el resto elpersonal solo podrá acceder al servidor de Apache2, para consumir los sitios que est﷽npublicados en dicho servido

Contenido:

Configuración de Autoridad certificadora, del servidor y primer cliente

Parte #1: Instalación de paquetes y servicios.

Todas las siguientes librerias son necesarias para la utilizar satisfactoriamente el servicio de VPN

```
apt-get install openvpn openssl ca-certificates iptables
mkdir -p /etc/openvpn/server/easy-rsa/
wget  https://github.com/OpenVPN/easy-rsa/releases/download/v3.0.8/EasyRSA-
3.0.8.tgz
chown -R root:root /etc/openvpn/server/easy-rsa/
tar xz EasyRSA-3.0.8.tgz /etc/openvpn/server/easy-rsa/
```

el permiso root:root será para que el usuario root o el grupo root pueda tener acceso a esa carpeta.

Parte #2: Iniciar entorno de EasyRSA PKI y configuración de CA (Autoridad certificadora)

Primero nos drigimos a la carpeta del servidor e ingresamos los siguientes comandos

```
cd /etc/openvpn/server/easy-rsa/
./easyrsa init-pki
```

Easy-rsa es una utilidad para crear y administrar Autoridades certificadoras, en terminos sencillos esto significa que puede crear y administrar CA Raiz, CA intermedias, clientes finales solicitar y firmar certificados e incluso las listas de revocación de certificados CRL

Parte #3: Creación de Deffie Hellman

En este punto crearemos la clave o módulos Diffie-Hellman utilizados por OpenVPN al establecer el primer contacto entre los nodos de la VPN.

```
./easyrsa gen-dh
```

Parte 4: Generación de la llave y el certificado de la Autoridad Certificadora

el .crt Es la clave pública de la autoridad certificante encapsulada en un formato de certificado digital x509 que tanto cliente como servidor OpenVPN utilizarán para identificarse entre si con confianza mutua

la .key es la clave privada RSA de la autoridad certificante, y es con la que se firman las claves y certificados del servidor.

```
./easyrsa --batch build-ca nopass
```

Parte 5: Copiar la llave y el certificado de la CA

Esto es necesario para facilitar la configuración del servidor OpenVPN

```
cp pki/ca.crt /etc/openvpn/server
cp pki/private/ca.key /etc/openvpn/server
```

Parte #6 Creación de certificado y llave del servidor:

Luego de crear la CA crearemos el certificado y la llave para el servidor

```
EASYRSA_CERT_EXPIRE=3650 ./easyrsa build-server-full server nopass
cp pki/issued/server.crt /etc/openvpn/server
cp pki/private/server.key /etc/openvpn/server
```

Parte 7: Creación de certificado y llave para el cliente

Con el siguiente comando crearemos la key y el crt del primer cliente cliente.

```
EASYRSA_CERT_EXPIRE=3650  ./easyrsa build-client-full JuanDanielWindows nopass
```

Parte 8: Generar CRL

El siguiente comando será para crear una lista de certificados revocados

```
./easyrsa gen-crl
```

Parte 9: Administrar los permisos de la CRL

El siguiente permiso es para que el documento no tenga dueño ni grupo y cualquiera la pueda acedder

```
chown nobody:nogroup /etc/openvpn/server/crl.pem
```

Parte 10: Generar una llave para TLS-Crypt

Esto nos servirá para agregar soporte para usar la autentificación TLS y de este modo fortificar la seguridad del servidor VPN.

```
openvpn --genkey --secret /etc/openvpn/server/tc.key
```

Configuración de OpenVPN

Parte 1: Configuración del servidor

Se configura el archivo server.conf de openvpn, lo mejor utilizar los archivos de configuración de muestra de OpenVPN como punto de partida para su propia configuración.

```
local 10.0.4.7
port 1194
proto udp
dev tun
ca ca.crt
cert server.crt
key server.key
dh dh.pem
auth SHA512
tls-crypt tc.key
topology subnet
server 10.8.0.0 255.255.255.0
push "redirect-gateway def1 bypass-dhcp"
ifconfig-pool-persist ipp.txt
push "dhcp-option DNS 168.63.129.16"
push "dhcp-option DNS 10.0.4.5"
keepalive 10 120
cipher AES-256-GCM
user nobody
group nogroup
persist-key
persist-tun
verb 3
crl-verify crl.pem
explicit-exit-notify
```

Acá especificamos:

1. ip privada del servidor
2. Puerto que utilizará el servidor, en este caso es el 1194 (puerto para OpenVPN)
3. Protocolo de la capa de transporte
4. Crea un tunel de enrutamiento
5. Certificado de la Autoridad Certificadora
6. Certificado del servidor
7. Llave del servidor
8. Archivo DH para intercambio de claves
9. autenticación de tipo Sha512 (función hash criptografica)
10. Llave del TLS
11. La red que creará será de tipo subred
12. El pool de ips para la subred con su debida mascara
13. Esta directiva se habilita para redirigir a todos los clientes por el gateway predeterminado del servidor
14. Los servidores DNS que utilizará el clente, deberán ser servidores de tipo publico
15. Cifrado criptografico que utilizara en la capa de transporte
16. No va tener ningun grupo o usuario especifico
17. evita acceder a ciertos recursos al reiniciar que puede que ya no sea accesible porque de la degradación de privilegios.
18. Envia el nivel apropiado al log
19. Lista de certificados revocados.
20. Notifica a los clientes cuando el servidor se reinicia y tambien reconecta de forma automatica

Parte 2: Configuración del cliente

Al igual que el archivo anterior buscamos el archivo client.conf y configuramos de la siguiente manera

```
client
dev tun
proto udp
remote 52.188.21.10 1194
resolv-retry infinite
nobind
persist-key
persist-tun
remote-cert-tls server
auth SHA512
cipher AES-256-GCM
ignore-unknown-option block-outside-dns
block-outside-dns
auth-nocache
verb 3
```

Se especifica

1. El archivo será la configuración de los clientes.
2. utilizara el mismo que el servidor, en este caso un tunel de enrutamiento
3. La ip publica del servidor
4. Resuelve de forma indefinida o infinita el nombre del servidor OV
5. Los clientes no necesitan ligar un puerto especifico de comunicación
6. Preserva los datos de key y tun despues de un reinicio
7. Verifica los certificados del servidor para checar que la clave es correcta.
8. autenticación de tipo Sha512
9. Cifrado de la capa de transporte, debe ser igual al servidor
10. Bloquea dns o servidores adaptados a otros puertos de red.
11. No almacena llaves en la cache
12. Envia el nivel apropiado de log

Programación de script para creación de usuarios nuevos

El siguiente script fue realizado con el fin de ejecutar comandos por medio de una bash scripting y lograr crear nuevos usuarios para el sistema, todo esto con el fin de facilitar las labores administrativas

Script:

```
    new_client () {
    {
    cat /etc/openvpn/server/client-common.txt
    echo "<ca>"
    cat /etc/openvpn/server/easy-rsa/pki/ca.crt
    echo "</ca>"
    echo "<cert>"
    sed -ne '/BEGIN CERTIFICATE/,$ p' /etc/openvpn/server/easy-
rsa/pki/issued/"$client".crt
```

```
    echo "</cert>"
    echo "<key>"
    cat /etc/openvpn/server/easy-rsa/pki/private/"$client".key
    echo "</key>"
    echo "<tls-crypt>"
    sed -ne '/BEGIN OpenVPN Static key/,$ p' /etc/openvpn/server/tc.key
    echo "</tls-crypt>"
    } > ~/"$client".ovpn
}
    echo
    echo "Nombre del nuevo usuario y tipo de sistema operativo (ejemplo:
(PedroWindows)):"
    read -p "Nombre: " unsanitized_client
    client=$(sed
's/[^0123456789abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ_-]/_/g' <<<
"$unsanitized_client")
    while [[ -z "$client" || -e /etc/openvpn/server/easy-
rsa/pki/issued/"$client".crt ]]; do
    echo "$client: Nombre no valido, intente de nuevo."
    read -p "Nombre: " unsanitized_client
    client=$(sed
's/[^0123456789abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ_-]/_/g' <<<
"$unsanitized_client")
    done
    cd /etc/openvpn/server/easy-rsa/
    EASYRSA_CERT_EXPIRE=3650 ./easyrsa build-client-full "$client" nopass
    new_client
    echo
    echo "$client added. Configuration available in:" ~/"$client.ovpn"
    exit
```

Configuración de servicios de Firewall

Configurar el servidor para que acepte paquetes por el puerto 1194 (OpenVPN)

```
iptables -P INPUT ACCEPT
iptables -P FORWARD ACCEPT
iptables -P OUTPUT ACCEPT
iptables -A INPUT -p udp -m udp --dport 1194 -j ACCEPT
iptables -A FORWARD -m state --state RELATED,ESTABLISHED -j ACCEPT
```

En este proyecto el acceso a los servidores es sumamente restringido, de forma que solo los administradores de la red pueden ingresar a los mismos, en este caso por medio de el firewall iptables habilitaremos la conexión a solo 1 dispositivo para ingresar y el resto de la red de cliente será bloqueada.

```
root@NoireVPN:~# iptables -A FORWARD -s 10.10.0.2 -d 10.0.4.5 -j ACCEPT
root@NoireVPN:~# iptables -A FORWARD -s 10.10.0.2 -d 10.0.4.6 -j ACCEPT
root@NoireVPN:~# iptables -A FORWARD -s 10.10.0.0/24 -d 10.0.4.5 -j DROP
root@NoireVPN:~# iptables -A FORWARD -s 10.10.0.0/24 -d 10.0.4.6 -j DROP
root@NoireVPN:~# iptables -A FORWARD -s 10.10.0.0/24 -j ACCEPT
```

1. Acepta todos los paquetes de entrada
2. acepta todos los paquetes de salida al exterior de la red
3. Acepta todos los paquetes generados dentro de la red
4. Acepta todos los paquetes de entrada del protocolo de transporte UDP del puerto 1194
5. Acepta todo el trafico de entrada a las conexiones establecidas
6. Acepta todo el trafico de la ip 10.10.0.2 (Administrador de la red) hacia la ip 10.0.4.5 (Servidor web)
7. Acepta todo el trafico de la ip 10.10.0.2 (Administrador de la red) hacia la ip 10.0.4.5 (Servidor DB)
8. Deniega el acceso a cualquier dispositivo proveniente de la red 10.10.0.0/24 hacia el servidor web
9. Deniega el acceso a cualquier dispositivo proveniente de la red 10.10.0.0/24 hacia el servidor DB
10. Acepta todo el trafico proveniente de la red 10.10.0.0/24

Configuración de Vhost Apache 2 y servicio DNS

Parte #1: Instalar los paquetes de apache2 y el Servicio de DNS

Para elaborar está sección necesitamos instalar los siguientes servicios por medio de los comando:

```
sudo apt-get update
sudo apt-get install apache2 bind9
```

Parte #2: Crear los directorios de las 2 páginas web

Cada una de las páginas deberá tener su propio directorio que contendra todo su codigo web

```
sudo mkdir -p /var/www/html/noire1.isw612.xyz
sudo mkdir -p /var/www/html/noire2.isw612.xyz
```

Parte #3: Crear la configuración de los Vhost

En este paso copiamos el archivo por defecto y creamos una copia para cada una de nuestras vhost

```
sudo cp /etc/apache2/sites-available/000-default.conf /etc/apache2/sites-
available/noire1.isw612.xyz
sudo cp /etc/apache2/sites-available/000-default.conf /etc/apache2/sites-
available/noire2.isw612.xyz
```

Parte 4: Configuración de los sitios

Se configura los archivos anteriormente generados para que queden de la siguiente forma

```
<VirtualHost *:80>
        ServerAdmin jtrejosp@est.utn.ac.cr
        ServerName noire1.isw612.xyz
        ServerAlias www.noire1.isw612.xyz

        DirectoryIndex index.html
        DocumentRoot /var/www/noire1.isw612.xyz/public_html

        <Directory /var/www/noire1.isw612.xyz/public_html>
                DirectoryIndex index.html
                AllowOverride All
                Require all granted
        </Directory>

        ErrorLog ${APACHE_LOG_DIR}/noire1.isw612.xyz.error.log
        LogLevel warn
        CustomLog ${APACHE_LOG_DIR}/noire1.isw612.xyz.access.log combined
</VirtualHost>
```

```
<VirtualHost *:80>
        ServerAdmin jtrejosp@est.utn.ac.cr
        ServerName noire2.isw612.xyz
        ServerAlias www.noire2.isw612.xyz

        DirectoryIndex index.html
        DocumentRoot /var/www/noire2.isw612.xyz/public_html

        <Directory /var/www/noire2.isw612.xyz/public_html>
                DirectoryIndex index.html
                AllowOverride All
                Require all granted
        </Directory>

        ErrorLog ${APACHE_LOG_DIR}/noire2.isw612.xyz.error.log
        LogLevel warn
        CustomLog ${APACHE_LOG_DIR}/noire2.isw612.xyz.access.log combined
</VirtualHost>
```

Parte 5: Creación de los archivos html que presentará la pagina

Con los siguientes comando crearemos los directorios de las 2 páginas web e ingresamos el siguiente codigo en los archivos .html

```
mkdir /var/www/noire1.isw612.xyz/public_html/
cd /var/www/noire1.isw612.xyz/public_html/
nano index.html
![noire1](imgs\Noire1html.PNG)
```

```
mkdir /var/www/noire2.isw612.xyz/public_html/
cd /var/www/noire1.isw612.xyz/public_html/
```

```
nano index.html
![noire1](imgs\Noire1html.PNG)
```

Parte #6: Habilitar las paginas web

Con el siguiente comando se habilitaran las paginas web para los que deseen acceder a ellos

```
a2ensite noire1.isw612.xyz
a2ensite noire2.isw612.xyz
```

Parte #7 Reiniciar los servicios de apache2

```
systemctl restart apache2
```

Configuración del servicio Domain Name Server

Parte #1: Instalación de paquetería

apt-get install bind9

Parte #2: Cofiguración del archivo named.conf.local

Acá definiremos las zonas directas e inversas para acceder a nuestras páginas web

```
nano /etc/bind/named.conf.local
![local ns](imgs\named.conf.local.PNG)
```

Parte #3: Configuración de zonas directas

Acá configuramos para que el servicio traduzca una cadena de nombre a una dirección ip especifica donde se encuentra el equipo o servicio

```
nano /etc/bind/db.noire1.isw612.xyz
![noire1 ns](imgs\ns.noire1.PNG)
```

```
nano /etc/bind/db.noire2.isw612.xyz
![noire1 ns](imgs\ns.noire2.PNG)
```

Parte #4: Configuración de zonas inversas

lo que harán estas zonas es tomar una dirección ip y traducirla a una cadena de nombres.

```
nano /etc/bind/db.4.0.10noire1.in-addr.arpa
![noire1 ns](imgs\ns.noire2i.PNG)
```

```
nano /etc/bind/db.4.0.10noire2.in-addr.arpa
![noire1 ns](imgs\ns.noire1i.PNG)
```

Parte #5 Reiniciar los servicios de bind9 y comprobar el estado

```
systemctl restart bind9
systemctl status bind9
```

Configuración de la Base de Datos en MySQL

Parte #1: Instalacion de la paqueteria

Este comando instalará todos los archivos necesarios para la administración de base de datos en MySQL

```
apt-get install mysql-server
```

Parte #2: ingresar a mysql como root

Ingresamos a la base de datos como el usuario root para crear el nuevo usuario

```
mysql -u root -p
```

Parte #3: Creacion de nuevo usuario y darle permisos para tener privilegios

```
CREATE USER 'Juan' IDENTIFIED BY 'secret123';
GRANT ALL PRIVILEGES ON *.* TO 'Juan';
```

Parte #4: Crear bases de datos Sakila, Northwind y World

```

```

```
CREATE DATABASES Sakila
CREATE DATABASES Northwind
CREATE DATABASES World
```